

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

JAZYKY A KÓDOVÁNÍ WEBOVÝCH APLIKACÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

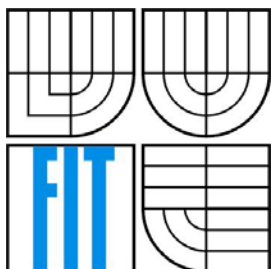
AUTOR PRÁCE
AUTHOR

JAROSLAV CHARVÁT

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

JAZYKY A KÓDOVÁNÍ WEBOVÝCH APLIKACÍ

LANGUAGES AND WEB APPLICATIONS ENCODING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAROSLAV CHARVÁT

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MICHAEL KUNC

BRNO 2008

Zadání bakalářské práce

Řešitel: **Charvát Jaroslav**
Obor: Informační technologie
Téma: **Jazyky a kódování webových aplikací**
Kategorie: Web

Pokyny:

1. Seznamte se s technologiemi pro vývoj webových aplikací a informačních systémů, zejména s PHP a MySQL.
2. Seznamte se se znakovými sadami vhodnými pro kódování webových aplikací.
3. Prostudujte možnosti databáze MySQL z hlediska uložení dat v různém kódování.
4. Navrhněte aplikaci, na které demonstujete použití různých jazyků a kódování. Minimum je deset odlišných jazyků, včetně asijských.
5. Vytvořte ukázkovou vícejazyčnou webovou aplikaci. Implementujte možnost přepínání mezi znakovými sadami. Dále implementujte funkci umožňující libovolné konverze znakových sad.
6. Zhodnoťte dosažené výsledky a diskutujte další možné pokračování tohoto projektu.

Literatura:

- Stejskal, J.: Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu. Computer Press, 2004.
- Gilmore, W. J.: Velká kniha PHP5 & MySQL. Zoner Press, 2005.
- Ullman, L.: PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek. Computer Press, 2004.

Při obhajobě semestrální části projektu je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Kunc Michael, Ing.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2007
Datum odevzdání: 23. ledna 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Štětčova 2

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Jaroslav Charvát**
Id studenta: 84437
Bytem: Lnářská 114, 396 01 Humpolec
Narozen: 01. 05. 1985, Pelhřimov
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Jazyky a kódování webových aplikací
Vedoucí/školitel VŠKP: Kunc Michael, Ing.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění


1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel


.....
Autor

Abstrakt

Cílem této bakalářské práce je vytvoření webové aplikace, která demonstruje možné použití různých kódování znaků pro různé abecedy a jazyky. V návrhu popisuji asi nejsložitější část práce.

S postupem v řešení jsem do značné míry musel i měnit samotný návrh. Dále se tu zabývám rozбором technologií, které jsem k realizaci využil. V části tohoto dokumentu nazvané Implementace se pokusím vysvětlit realizaci jednotlivých částí této práce, jakož i sloučení do celku. Aplikace umožňuje ovládání v 10 jazycích.

Klíčová slova

Diskusní skupina, kódování, PHP, CSS, HTML, JavaScript, MySQL, SQL.

Abstract

The point of this bachelor's thesis is creation of web application, which shows using different char sets for various alphabets and languages. In proposal I describe the hardest part of this work. During implementation I had to largely modify proposal. Next I am concerned with analyzing technologies, which I used for realization. I am trying to explain realization parts of the work in part of this document I called Implementation. Application offer controlling in 10 languages.

Keywords

Discussible group, encoding, PHP, CSS, HTML, JavaScript, MySQL, SQL.

Citace

Jaroslav Charvát: Jazyky a kódování webových aplikací, bakalářská práce, Brno, FIT VUT v Brně, 2008

Jazyky a kódování webových aplikací

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michaela Kuncce
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jaroslav Charvát

8. 1. 2008

Poděkování

Chtěl bych poděkovat mému vedoucím v této práci panu Ing. Michaelu Kuncovi za jeho vedení a návrhy k tomuto dílu.

© Jaroslav Charvát, 2008.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

| | |
|---|----|
| Obsah | 1 |
| Úvod | 3 |
| 1 Kódování na internetu | 4 |
| 1.1 Historie | 4 |
| 1.2 Současnost | 4 |
| 1.3 Možné pokračování | 4 |
| 2 Použité programovací nástroje | 5 |
| 2.1 Tvorba webové aplikace | 5 |
| 2.2 Programovací jazyk PHP | 5 |
| 2.3 Databáze MySQL | 6 |
| 2.4 Programovací jazyk JavaScript | 7 |
| 2.5 Nástroj pro úpravu stylů CSS | 7 |
| 3 Návrh řešení | 8 |
| 3.1 Rozbor problému | 8 |
| 3.1.1 Asijské jazyky a možné znakové sady | 8 |
| 3.2 ER Diagram, use case diagram | 9 |
| 3.2.1 Původní návrh | 10 |
| 3.2.2 Výsledný návrh | 10 |
| 3.3 Rozvržení vzhledu a funkce ovládacích prvků | 13 |
| 3.3.1 Vzhled | 13 |
| 3.3.2 Funkce ovládacích prvků | 14 |
| 4 Implementace | 16 |
| 4.1 Databáze | 16 |
| 4.1.1 Téma | 16 |
| 4.1.2 Příspěvek | 16 |
| 4.1.3 Uživatel | 16 |
| 4.1.4 Překlad | 17 |
| 4.2 Skripty PHP | 17 |
| 4.2.1 Struktura stránky | 17 |
| 4.2.2 Přepínání jazyků | 17 |
| 4.2.3 Přihlašování | 19 |
| 4.2.4 Odhlašování | 19 |
| 4.2.5 Registrace | 20 |
| 4.2.6 Nástěnka | 20 |

| | | |
|-----|------------------------------|----|
| 4.3 | Ostatní..... | 20 |
| 5 | Závěr..... | 22 |
| 5.1 | Další možné pokračování..... | 22 |
| | Literatura..... | 23 |
| | Seznam příloh..... | 24 |
| | Seznam Obrázků..... | 25 |

Úvod

Všichni žijeme ve velmi uspěchané době. Většina věcí kolem nás se velmi rychle mění a téměř každý se snaží přizpůsobit nebo dokonce se na tomto faktu aktivně podílet. Svět počítačů, jak jej někteří nazývají, není výjimkou. Spíše naopak. V dřívějších dobách to byly hlavně armádní účely, jež popoháněli informační technologie kupředu. Do dnešní doby se však hodně změnilo. Internet už dávno nefunguje jako armádní počítačová síť několika mála vybraných amerických základů ani jako síť pro vědecké účely, kde každý článek měl svůj studijní účel.

Dnes jsou výpočetní technika a internet existenčně důležité pro většinu vědních, komerčních, ale i soukromých odvětví ve všech průmyslově vyspělých zemích. Toto neustálé rozšiřování pole působnosti má za následek stále stoupající nároky na kvalitu a rozsah možných funkcí. Rychlost internetu se stále zvyšuje a to nejen od poskytovatelů k poslednímu článku, uživateli, ale v celém jeho rozsahu. Kvalita webových aplikací se pořád zlepšuje, ustavičně se zvyšují nároky na vzhled a tzv. user-friendly prostředí. Stejně tak se mění i kapacita paměťových prvků. Kde se dříve lidé snažili uspořit každý kB, dnes se málokdo stará o rozdíly v řádech MB. Pro to co se člověku líbí nebo je jednoduché se to naučit, nejlépe oboje, je ochoten obětovat hodně.

S rozšířením internetu do takřka celého světa se objevilo několik do té doby neřešených problémů. Já zde zmíním jen jeden, jelikož je předmětem řešení této práce. Jedná se o problém s kódováním. Zprvu se nepředpokládalo, že se vše takto vyvine a že internet stejně jako celá výpočetní technika se bude používat ve většině zemí světa. A zde se dostáváme k jádru problému. Angličtina je sice asi nejpoužívanějším jazykem na světě, přesto bychom se mylili, myslet si, že každého člověka, který bude chtít používat internet, donutíme naučit se anglicky. Takový internet by se nikdy nedostal do pozice, kterou má dnes. Je proto nutné zohlednit všechny jazyky a to není možné pomocí normalizovaných znakových sad, které byly v počítačích původně používány.

Bohužel v té době nebyl nikdo, kdo by vytvořil universální kódování jako je dnes UTF-8 a výrazně tak zjednodušil zápis znaků v jakémkoliv jazyce. Jelikož se nikdo takový nenašel, vzniklo nepřehledné množství porovnávání. Převody mezi znakovými sadami byly velmi obtížné. Postupem času se kódování vytříbila a efektivněji se zaplnily kódové tabulky sloučením podobných jazyků.

Dnešní situace je nepoměrně lepší. Nejen že pro zapsání podobných jazyků můžeme použít jedno, nejčastěji jednobajtové kódování, ale můžeme použít zcela univerzální několikabajtové kódování UTF-8, s jehož pomocí se dá zapsat téměř jakýkoliv jazyk. Tento luxus se zaplatí větším místem v paměti, které takový text zabere. Na znacích a jazyku záleží, o kolik paměťové nároky vzrostou. Každý si může zvolit, co je mu příjemnější.

1 Kódování na internetu

Písmena, číslice, slova, ale avšak i celé věty a články nejsou v počítači uloženy jinak, než jako číselné informace. Záleží už poté jen na tom, jak tyto čísla budou interpretována. K tomu slouží kódování nebo též znaková sada. Jinak tomu není ani na internetu.

1.1 Historie

V minulosti zde bylo velké množství různých kódování, což způsobovalo velké problémy. Každé zařízení si díky jinému kódování mohlo vyložit data jinak. Začaly vznikat snahy o sjednocení a tím i odstranění takto vzniklých nedostatků.

Kolem let 1963 a 1964 došlo k zavedení prvních standardů. Tyto standardy jsou ASCII (1963) a EBCDIC (1964) zabírající v paměti 1Byte na znak, ale využito je jen dolních 7 bitů. První zatím nebyl potřeba. Jelikož je na světě mnoho různých jazyků a mnoho odlišných abeced, nemohly tyto standardy vůbec stačit. Toto vedlo ke vzniku velikého množství ad-hoc rozšíření a k vytvoření nových kódových sad.

1.2 Současnost

V současnosti máme spoustu znakových sad. Dokonce i několik pro jeden jazyk. Velikost různých kódování může být různá. Velikostí myslíme počet bytů na jeden znak a je závislá na obsáhlosti abecedy, jež je ukládána.

Například Windows-1250 někdy též uváděno jako cp-1250 je jednobajtovým kódováním, ve kterém je možné vyjádřit nejen jazyky jako je český a anglický, ale dále albánštinu, chorvatštinu, rumunštinu, polštinu, slovenštinu a maďarštinu. Tato znaková sada zabírá v paměti 1byte za každý znak. To je umožněno podobností těchto abeced, neboť mají vždy několik znaků společných.

Další možností je použití některé z variant tabulek Unicode. Tyto obsahovaly nejprve 16 později 32 bitů. Vznikl systém kódování znaků založen na Unicode – UTF-8 viz později.

1.3 Možné pokračování

Jak to bude vypadat v budoucnosti, je těžké odhadnout. Osobně si myslím, že využívání znakových sad jako je UTF-8 bude stále narůstat. Je sice pravda, že každé písmeno může být zakódováno ve více než jednom bytu, ale tato cena je při dnešních rychlostech síťových prostředků a kapacitách paměťových médií celkem zanedbatelná.

2 Použité programovací nástroje

Pro tvorbu webových aplikací je v dnešní době nepřehledné množství nástrojů. V této práci je využito především programovacího jazyka PHP a databázového systému MySQL. Vzhledem k zaměření mé bakalářské práce na kódování webových aplikací je využití MySQL od verze 4.1 nespornou výhodou. Vysvětlení viz 2.3 Databáze MySQL. Jako jazyk pro vytvoření samotné aplikace byl použit jazyk PHP hlavně pro jeho dobré komunikační vlastnosti se serverem MySQL. Dále jsem využil softwaru phpMyAdmin, což je nástroj na vytvoření a úpravu databází na serveru. Bylo zapotřebí vybrat vhodnou verzi, která dovolí vytvoření tabulek, sloupců a databází s různým kódováním, což je předpokladem použití.

2.1 Tvorba webové aplikace

K vytvoření webové prezentace, tzv. stránky se hlavně zpočátku využíval jazyk HTML. HTML nebo též Hypertext Markup Language. Jedná se o zapisování textu, který má být na stránce vyobrazen, mezi párové značky, tzv. tagy. Tagy určují, jak má být text formátován. Na následujícím řádku je vidět zápis mezi párové znaky `<a>` a ``. „popisek obrázku“ je odkazem a na jaké místo ukazuje je vyjádřeno parametrem href, tedy na soubor login.php.

```
<a href="login.php">  
  popisek odkazu  
</a>
```

Př. Zápis odkazu v jazyce HTML

Tento jazyk byl později zpřísněn o striktní dodržování ukončovacích značek a rozšířen o několik dalších pravidel. Takto změněný jazyk se jmenuje XHTML. Avšak využití HTML nebo XHTML v dnešní době bez dalších prostředků není moc časté, neboť se jedná o takzvané statické stránky. Neumožňují interakci s uživatelem, a tudíž mohou zobrazovat jen stále stejné informace. To se mění s příchodem skriptovacích jazyků jako je PHP, Perl, apod.

2.2 Programovací jazyk PHP

Jak již bylo výše řečeno, HTML bez dalších prostředků je nedostatečné. Zde vzniká prostor pro jazyk PHP. Díky němu jsme schopni dynamicky měnit obsah, do té doby statických stránek. Na základě informací zadaných uživatelem nebo automaticky samotným skriptem je do stránky v HTML kódu vkládán text. Je též možné text nevkładat do HTML, ale pomocí PHP vygenerovat celou stránku. To

však není příliš časté. V následující ukázce je znázorněno vložení textu do HTML stránky. Pro dobré porovnání se samotným HTML jsem záměrně použil vložení odkazu na stránku.

```
<a href="login.php">
```

```
<?php
```

```
if ($prihlasen) echo 'Přihlásit';
```

```
else echo 'Odhlásit';
```

```
?>
```

```
</a>
```

Př. Použití PHP pro vkládání textu do HTML

Veškerý PHP kód se musí vyskytovat mezi značkami „<?php” a “?>”. Tyto značky řeknou serveru, na kterém jsou stránky nahrány, že před odesláním webové prezentace klientovi je nutné provedení příkazů vloženého skriptu. Význam zbylých dvou řádků ukázky je následující. Operátor „if“ slouží pro rozdělení kódu na základě podmínky v závorce. Konkrétně tady otestuje proměnnou \$prihlasen, je-li nenulová, provede se příkaz echo 'Přihlásit'; v opačném případě echo 'Odhlásit';. Funkce echo vloží řetězec, převzatý jako parametr v uvozovkách nebo apostrofech, do HTML kódu.

Bližší informace o PHP využité v této práci budou případně uvedeny v sekci věnované implementaci.

2.3 Databáze MySQL

Databázový systém MySQL patří v dnešní době bezesporu k nejlepším. To je patrně zapříčiněno pravděpodobně těmito důvody.

Základní verze pro nekomerční využití je distribuována zdarma. To vedlo a vede k jejímu snadnému a rychlému rozšíření nejen po internetu. Dále skvěle spolupracuje s jazykem PHP, který je dnes taktéž velmi často používán. Databáze byla také převedena na více platforem, než bylo v době jejího vývoje běžné. Uveďme si například UNIX, WIN32 a OS/2.

Toto byly důvody všeobecného rozšíření jako účinného prostředku, při tvorbě webových aplikací s ukládáním dat do databáze. Pro vytvoření bakalářské práce se zaměřením na kódování zde však vzniká ještě jeden důvod proč si vybrat právě tento databázový systém. Od verze 4.1 totiž došlo k výraznému zlepšení způsobu, jak tato databáze pracuje s kódováním. Dříve bylo možno nastavit jednu znakovou sadu pro celou databázi. S příchodem verze 4.1 už tomu tak není. Je možné nastavit jiné porovnávání pro celou databázi, tabulku a dokonce i sloupce. Pokud např. není nastaveno kódování pro sloupce, použije se to, jež bylo určeno pro tabulku. Není-li uvedeno ani to, MySQL použije znakovou sadu určenou pro celou databázi.

2.4 Programovací jazyk JavaScript

Tento jazyk nebyl využit v této práci ve velkém rozsahu, proto jen stručně. Stejně jako PHP slouží i JavaScript pro doplnění značkovacího jazyka HTML.

Na rozdíl od PHP se však JavaScript kód provádí na straně klienta a ne serveru jak tomu bylo u PHP. Výhodou je méně časté odesílání zpráv na server, což při pomalejším připojení k serveru může ušetřit čas.

2.5 Nástroj pro úpravu stylů CSS

Dnešní webové prezentace na internetu nemají snahu pouze obsáhnout co možná nejpřesněji a nejpodrobněji danou tematiku, jak tomu bylo v dobách využití internetu jen nebo převážně pro vědecké účely. Dnes je nedílnou potřebou zaujmout návštěvníka stránek i jejich úpravou, vzhledem, tzv. design. Vytvořit dobrý design na stránkách jen pomocí HTML nebo XHTML není jednoduché.

Z tohoto důvodu vznikly tzv. kaskádové styly – CSS. CSS nebo též Cascading style sheet je jazykem vytvořeným pro popsání vzhledu HTML, XHTML nebo XML. CSS je většinou externí soubor, ve kterém jsou definovány veškeré potřebné informace o elementech obsažených v HTML dokumentu. Druhou možností zápisu těchto definic je možné přímo do hlavičky (mezi tagy <head> a </head>). Tato druhá možnost však není tak přehledná, proto se používá jen ojediněle.

Jak vypadá syntaxe, můžeme vidět na následující ukázce.

a.hlavni

```
{  
  text-decoration: none;  
  color: red;  
}
```

Př. Definování odkazů ve třídě hlavní v jazyce CSS

Nyní vysvětlím význam jednotlivých symbolů. V prvním řádku řetězec .hlavni určuje, že určujeme element, který patří do třídy hlavní. Písmeno „a“ zde představuje tag <a> v jazyce HTML, což je značka pro odkaz. Složené závorky nám pak ohrazují rozsah pravidel pro tento selektor (celý řetězec určující element nebo elementy u kterých se vlastnosti projeví) a.hlavni. V našem příkladě se mezi závorkami vyskytuje nastavení dvou vlastností. První je text-decoration: none; což zruší jinak tak běžné podtržení odkazů. Druhé pak nastaví barvu písma tohoto odkazu na červenou.

Byla zde uvedena jen krátká ukázka, na popsání všech vlastností zde není místo.

3 Návrh řešení

V této kapitole bude podrobně popsán návrh řešení. Během zhotovování webové aplikace, jež je součástí bakalářské práce, jsem se dostával do situací, které jsem při samotném návrhu neuvažoval. Z tohoto důvodu jsem byl nucen návrh řešení oproti původnímu do značné míry měnit.

Při vytváření běžné webové prezentace se setkáváme jen s minimem problémů, které zde vyvstaly na povrch. Veškeré změny, jež jsem prováděl během tvorby aplikace, jsou připsány podle významu k jednotlivým kapitolám

V první kapitole nazvané Rozbor problému se pokusím zamyslet nad zadáním. Co vše je požadováno, jak toho dosáhnout, případně co vše k tomu využít.

Druhá je pak věnována návrhu databáze – ER Diagramu a vysvětlení proč jsem jej vytvořil zrovna tak, jaký byl původní návrh a proč ke změně došlo.

Do poslední kapitoly patří návrh vzhledu, rozmístění a význam ovládacích prvků. Dále je zde uvedeno, co vše je možné ovládat, vytvářet, atd.

3.1 Rozbor problému

Podle zadání je požadováno vytvořit webovou aplikaci, která bude schopna pracovat hned s několika jazyky. Počet jazyků je stanoven na minimálně 10. Já jsem si zvolil následující: Čeština, Angličtina (spíše USA verze), Bulharština, Francouzština, Němčina, Polština, Ruština, Švédština, Japonština a Čínština. Pro správné fungování všech těchto jazyků by bylo možné jednoduše použít znakovou sadu UTF-8, to však není u této práce možné. Důležitou součástí zde je nejen přepínání jazyků, ale s jejich pomocí demonstrovat také různé ukládání v různých znakových sadách.

Nejprve je nutné určit, jak budou jednotlivé jazyky interpretovány. Pro Češtinu, Angličtinu a Polštinu byla využita znaková sada Windows-1250. Pro jazyky používající azbuku jako svou abecedu bylo vybráno porovnávání Windows-1251. Ty jazyky, které v základním dělení patří do Indoevropských jazyků, v našem případě Francouzština, Švédština a Němčina, jsou kódovány pomocí Latin 1 (iso-8858-1). Poslední skupinou je Japonština a Čínština. Zde ovšem narážíme na několik problémů, proto jsem tomu věnoval samostatnou kapitolu.

3.1.1 Asijské jazyky a možné znakové sady

Podívejme se nejprve na Čínštinu. Čínský jazyk je jedním z nejvíce používaných jazyků na světě. Mluví s ním 870 milionů – 1,2 miliardy lidí [3]. Čína je velká země a je územně dělená na několik provincií, autonomních území, speciálních administrativních regionů a dalších částí. Není ani pravidlem, že po celé zemi je možné se domluvit jedním jazykem. Je zde totiž používáno hned několik dialektů.

V Čínské republice se například používá úřední Čínština, Na většině území Čínské lidové republiky je to zase Mandarínština. Další dialekty již nejsou tak běžné proto zde nebudou zmíněny. Důležité však je, že obyvatelé mluvící různými nářečím si nerozumí. Možným způsobem dorozumění je písmo. Nezáleží totiž téměř na tom, kterými z dialektů se snažíte psát, znaky pro to budou převážně stejné. Výslovnost je nutno chápat jinak než je tomu například u latinky. Každé slovo je zapsáno pomocí jednoho nebo několika grafémů (znaků). Grafémy pro vyjádření nějaké skutečnosti (slovo, fráze, ...), jsou většinou ve všech dialektech stejné. Rozdílné je však čtení nebo též výslovnost těchto grafémů, což označujeme jako morfém. Gramotnost obyvatelstva je poměrně špatná, neboť aktivně se používá kolem 6000 znaků, ale existuje jich mnohem více. Největší slovníky obsahují kolem 80 000 znaků.

Japonštinou mluví kolem 130 milionů lidí. Oproti Čínštině je hlavním rysem Japonštiny rozdělení do úrovní zdvořilosti. Úrovní zdvořilosti může být kolem 12 a nejen díky tomu je považována za jeden z nejsložitějších jazyků na světě. Pro zápis je zde hned několik možností. Mohou být použité čínské znaky Kandži, dvě fonetické slabičné abecedy Hiragana, Katakana a ve výjimečných případech i latinka rómadží.

| Rómadží | Wataši | wa | miruku | (w)o | nomimasu |
|---------|--------|-------------|--------|-------------|----------|
| Kana | わたし | は | ミルク | を | のみます。 |
| Kandži | 私 | は | ミルク | を | 飲みます |
| Význam | Já | (partikule) | mléko | (partikule) | piji |

Př. Ukázka způsobů zápisu japonského jazyka. Převzato z [9].

Po zvážení všech těchto informací jsem se rozhodl využít pro oba jazyky čínských znaků Kandži a zakódovat je pomocí utf-8.

3.2 ER Diagram, use case diagram

Rozvržení databáze bylo jedním z prvních problémů, kterými jsem se v této práci zabýval. Jedná se o tu část návrhu, která v průběhu implementace prodělala nejvíce změn. Největšími změnami prošly zřejmě struktury tabulek.

Vlastnosti tabulek a tudíž i celé databáze byly měněny podle požadavků a průběžně získaných informací o možnosti ukládání dat v různých znakových sadách, jejich opětovné získání a použití v naší webové aplikaci. Postup návrhu databáze jsem rozdělil do dvou následujících kapitol. V původním návrhu budou předvedeny první myšlenky na rozvržení databáze, a co jsem tím sledoval. Ve druhé jsou popsány veškeré změny, jež byly oproti první verzi provedeny a proč tomu tak bylo. Oba návrhy budou doplněny obrázkem.

3.2.1 Původní návrh

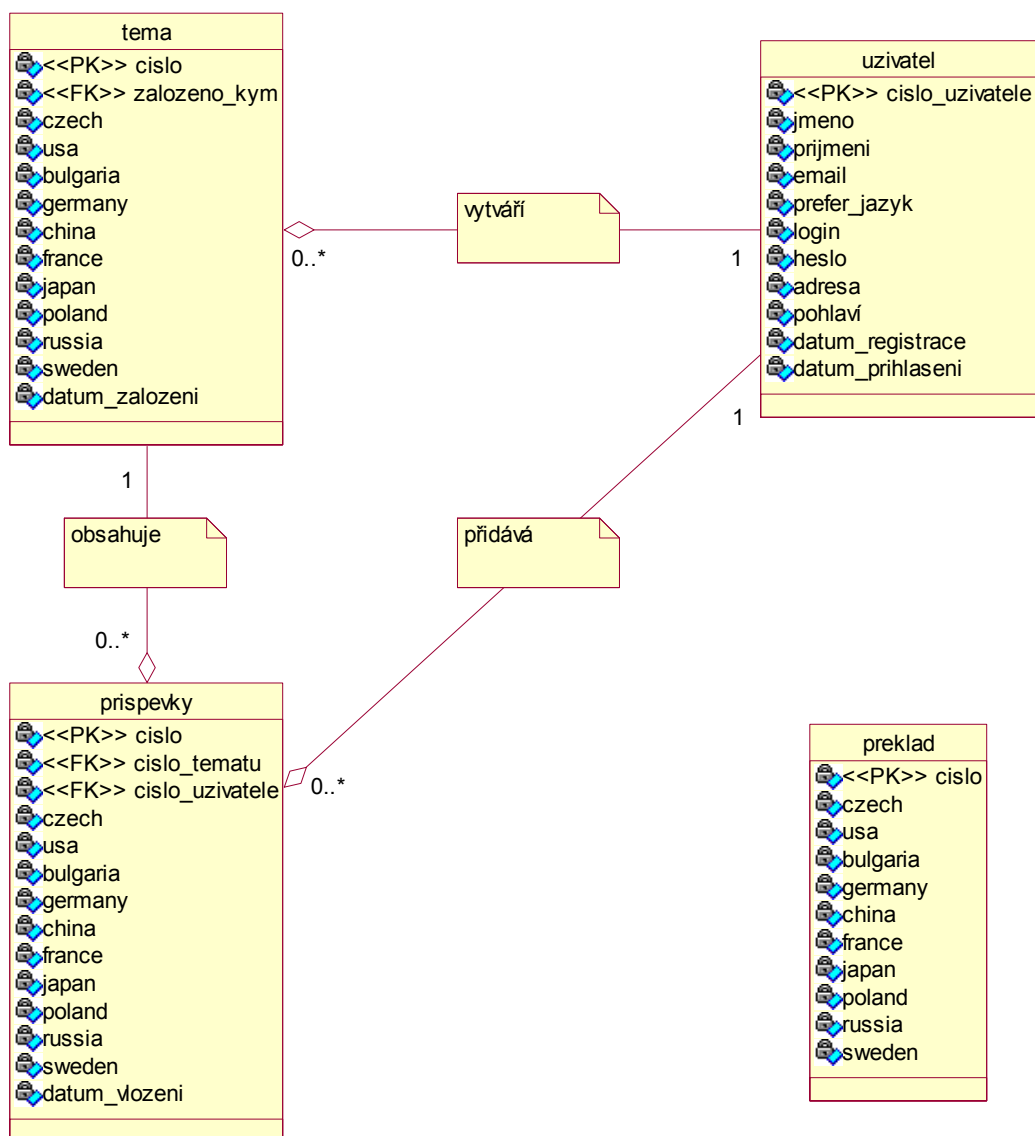
První návrh diagramu vztahů nebyl zprvu vytvořen. Jednalo se jen o představu, jak by mohla výsledná databáze vypadat. K jeho vzniku došlo až ve chvíli tvorby tohoto dokumentu a to hlavně z demonstrativních důvodů. Ze snahy interpretovat původní myšlenky o možné struktuře databáze. Jak by mohl vypadat první ER Diagram databáze, je možné vidět v příloženém seznamu obrázků pod číslem 1.

Za povšimnutí zde stojí asi hlavně tabulka téma, která zde obsahuje pouze 4 položky. První atribut „číslo“ zde představuje primární klíč, „založeno_kým“ pak číslo uživatele, který téma vytvořil. Nejdůležitější však jsou atributy „jméno“ a „jazyk“. Ve jméně měl být uložen název tématu uložený v UTF-8 a jazyk obsahoval informaci, v jakém jazyce bylo jméno napsáno. Jelikož výsledné stránky nebudou ve většině případů používat znakovou sadu UTF-8 je informace ze sloupce jazyk nutná pro zpětné zapsání textu do vhodné znakové sady jazyka.

Takto navržený systém byl postaven na předpokladu, že témat nebude mnoho a tudíž neměl být velký rozdíl mezi velikostí dat v UTF-8 a jiným vhodnějším kódování (např. Windows 1250 pro češtinu). Zároveň tento systém měl do značné míry zjednodušit operování mezi znakovými sadami. Ani jeden z těchto předpokladů však není správný. Vytvořeno může být i nespočet témat prázdných nebo jen s několika málo příspěvky, což by znamenalo narůstající nároky na paměťový prostor. Je sice pravda, že by se ve skriptech PHP poměrně dobře manipulovalo s daty z tabulky téma, neboť by byla vždy dostupná informace o porovnávání zprostředkovaně z atributu jazyk, avšak v rámci MySQL by to mohlo způsobovat nemalé potíže například při sestavování víceúrovňových dotazů.

3.2.2 Výsledný návrh

Oba výše zmíněné důvody a několik dalších vedly k přepracování návrhu ve finální variantu. Tu je možné vidět níže.



Př. ER Diagram, finální verze

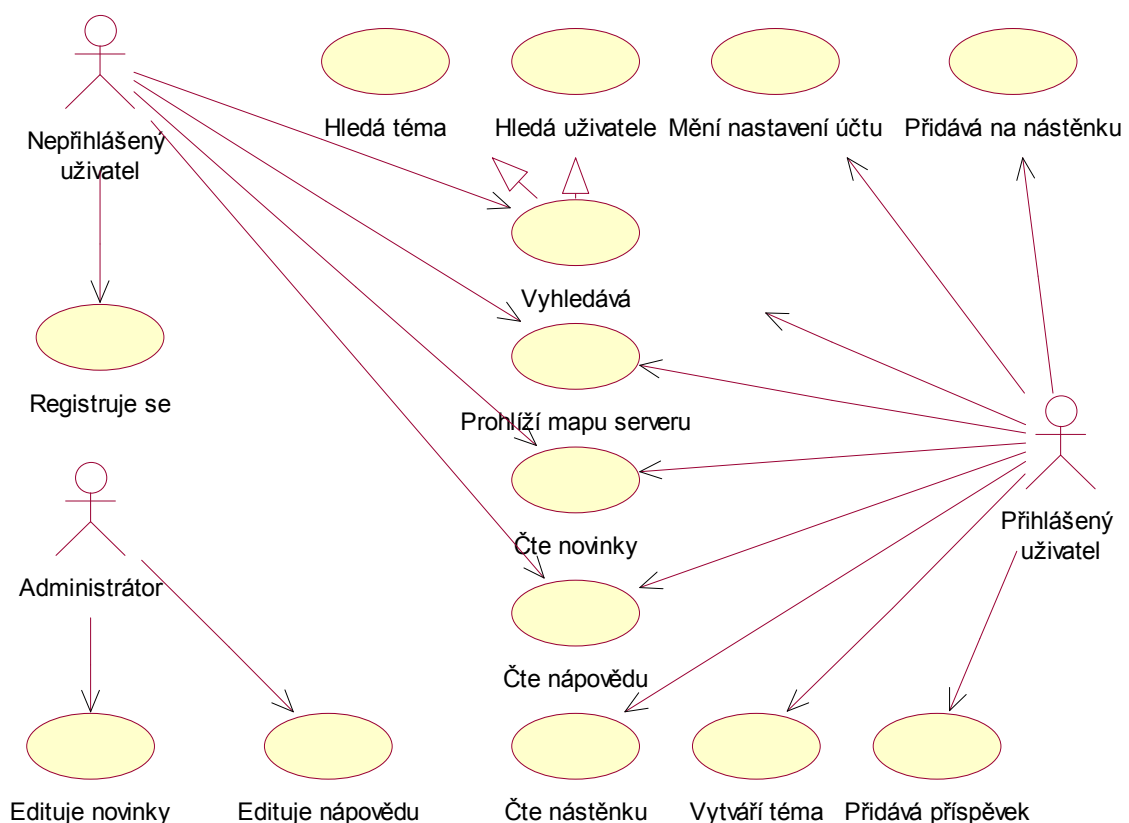
Porovnáme-li nyní tabulky téma v obou verzích, je zřejmé, že i nová verze obsahuje způsob opětovného získání informace o použitém kódování. Nyní je to však přímo název sloupce. Data již navíc nejsou ukládána universálně v UTF-8, nýbrž ve vhodně zvoleném kódování, které je ke každému sloupci vybráno zvlášť. Podobně je tomu tak i s tabulkou příspěvky.

U tabulky uživatel to již není tak jednoduché. Aby bylo možné něco podobného aplikovat i na tuto tabulku, musela by obsahovat k téměř každému současnému sloupci dalších devět pro zbylé jazyky nebo vytvořit devět nových tabulek. Toto je nemyslitelné, proto jsou informace uloženy v UTF-8. Bude-li nutné převádět tyto informace do jiných kódování, bude využito funkce CONVERT, která je součástí jazyka MySQL.

Druhou velikou změnou oproti původnímu návrhu je bezesporu tabulka překlad. Na první pohled není jasný její význam. Není žádným vztahem spojena s žádnou z ostatních tabulek, je však pro výslednou aplikaci nepostradatelná. Pro její správné vysvětlení jsem nucen částečně zasáhnout do kapitoly implementace. HTML soubor (popř. skript PHP) výsledné stránky obsahuje text, který bude vyobrazen a v hlavičce dokumentu nastavení kódové stránky, které má internetový prohlížeč použít pro jeho zobrazení. Aby bylo umožněno přepínání mezi různými jazyky, nabízely se dvě možné varianty. Pro první je nutné každému souboru vytvořit devět dalších variant s patřičně přeloženým textem a nastaveným kódováním v hlavičce dokumentu nebo to vyřešit dynamicky. Já jsem zvolil druhou možnost. Podrobnější vysvětlení přepínání bude vysvětleno v kapitole 4 věnované implementaci.

Další změny návrhu nemají tak zásadní význam. Do tabulek byly přidány atributy jako datum_vložení, datum_založení, datum_registrace a datum_přihlášení, které doplňují vlastnosti objektů.

Pro dobré vysvětlení funkcí celé aplikace byl vytvořen use case diagram. Vidět jej lze pod tímto textem.



Př. Use case diagram

Z obrázku je patrné rozdělení na 3 skupiny přístupu k aplikaci. Prvním je Administrátor. V databázi není veden nikdo jako uživatel s nadstandardním oprávněním. Za administrátora se považuje osoba, jež má přístup ke zdrojovým souborům. Novinky, mapa serveru i nápověda nejsou uloženy v databázi jako samostatné tabulky, nýbrž jsou v tabulce překlad jako jakýkoliv jiný text aplikace. Vysvětlení viz kapitola implementace. Chce-li tedy administrátor vložit např. novinku, zedituje vhodným způsobem soubor novinky.php a přeloží zprávu do všech jazyků. Takto přeložené informace pak vloží do tabulky překlad v databázi.

Druhou skupinou je nepřihlášený uživatel. Ten si může prohlížet téměř veškeré příspěvky vyjma nástěnky. Další možností je se zaregistrovat a po přihlášení využívat výhod přihlášených uživatelů. Je mu též umožněno vyhledávat uživatele nebo téma.

Poslední skupinou je přihlášený uživatel. Na rozdíl od nepřihlášených uživatelů je mu povoleno přidávat témata a vkládat do nich články. Číst příspěvky na nástěnce, vytvářet na ni další. Poslední funkcí je změna nastavení jakožto většiny údajů, jež zadal při registraci.

3.3 Rozvržení vzhledu a funkce ovládacích prvků

Cílem kapitoly je seznámit čtenáře s návrhem vzhledu, umístěním a funkcí ovládacích prvků. Kapitola byla rozdělena na dvě podkapitoly. První je věnována vzhledu a rozmístění prvků, druhá se zabývá jejich funkcí.

3.3.1 Vzhled

Rozměry aplikace jsou pevně specifikovány na 800x516pixelů. Pod úvodním a zároveň hlavním obrázkem je vloženo přepínání jazyků v podobě deseti vlajek a slovních popisků v angličtině sloužící jako odkazy.

Pod nastavením jazyků je vloženo hlavní menu. Nezohledňuje, zda je uživatel přihlášen či nikoliv, pokaždé nabízí tři funkce - mapu serveru, nápovědu a novinky. Vlevo od tohoto horního menu je vidět aktuální datum a napravo login jméno právě přihlášeného uživatele. Není-li nikdo přihlášen, ukáže se nepřihlášen.

Dalším prvkem je boční menu. Je závislé na faktu, zda je uživatel přihlášen. Není – li, je mu zobrazena pouze registrace, funkce pro hledání a přihlášení. Pokud je, pak jsou jeho možnosti rozsáhlejší. Kromě vyhledávání uživatelů a témat, což mohl i jako nepřihlášený, je mu nabídnuto i Přidávání témat a příspěvků, prohlížení všech témat, nahlížení na nástěnku a případně i vkládání nových informací na ni.

3.3.2 Funkce ovládacích prvků

Zde bude vysvětlen význam všech odkazů a tlačítek, co se stane po jejich použití a jaké jsou podmínky jejich využití.

3.3.2.1 Přepínání jazyků

Funkce přepínání jazyku je poskytována prostřednictvím 10 vlajek států, na jejichž jazyky se můžete přepnout. Ty jsou navíc doplněny slovními popisky pro upřesnění. Kliknutím na jeden z těchto odkazů vede k znovunačtení stránky ve zvoleném jazyce a jemu odpovídající znakové sadě.

3.3.2.2 Náповěda

Náповěda je ta část aplikace, kde je uživatel seznámen s ovládaním. Každý možný odkaz je zde krátce popsán, aby návštěvník získal představu, jak tyto stránky ovládat.

3.3.2.3 Novinky

Druhou možností v nabídce horního menu jsou novinky. Zde si každý může přečíst nejnovější informace o serveru, co bylo přidáno nebo změněno.

3.3.2.4 Mapa serveru

Na základě informace, zda je návštěvník přihlášen či nikoliv, jsou mu představeny veškeré odkazy, které může v tomto stavu spatřit. Tyto odkazy jsou seřazeny do struktury odpovídající faktu, jak je možné je vidět.

3.3.2.5 Registrace

Předpokládá se fakt, že každý uživatel bude registrován jen jednou. Toto je hlídáno kontrolou duplicity emailu a přihlašovacího jména. Za tohoto předpokladu je registrace přístupná jen pro návštěvníky, jež nejsou přihlášení. Po zaregistrování je každému uživateli zaslán email s přihlašovacím jménem a heslem, které vyplnil do registračního formuláře, napsaný v anglickém jazyce.

3.3.2.6 Hledání

Hledání je funkce pro všechny. Každý si zde najde, co potřebuje. U každého výpisu je vzhledem k nastavenému stránkování zobrazen maximálně počet záznamů odpovídající tomuto nastavení. Vyhledávání je rozděleno do dvou sekcí.

První se týká uživatelů. Po zadání krátkého řetězce jsou v databázi vyhledáni všichni, jejichž jméno, příjmení nebo login se shoduje s řetězcem, nebo jej obsahuje. Seznam těchto záznamů se poté ukáže jako tabulka na stránce a na konci každého řádku je vidět obálka. Ta představuje možnost poslat vybranému uživateli krátký email. Nebyl-li zadán žádný text, funkce vypíše všechny záznamy.

Druhá je pak zaměřena na témata. Stejně jako u uživatelů je zde vhodné zpřesnit vyhledávání krátkým textem, který napomůže při hledání. Výsledek hledání je opět ukázán v tabulce a jména témat fungují jako odkaz. Pro jeho využití již však musí být návštěvník přihlášen. Odkaz zobrazuje veškeré příspěvky obsažené v tématu.

3.3.2.7 Přihlášení

Přihlášení probíhá zadáním loginu a hesla a odesláním těchto dat na server. Ten zkontroluje všechny záznamy v tabulce uživatel pro login a zakódované heslo vnitřní funkcí `OLD_PASSWORD()`. Pokud je nalezen právě jeden záznam (více není možné vzhledem ke kontrolované duplicitě loginu), dojde k přihlášení.

3.3.2.8 Odhlášení

Tato funkce je mnohem jednodušší. Po kliknutí na odkaz jsou odstraněny prostředky představující přihlášení, např. proměnné. Odkaz je samozřejmě stejně jako všechny zbývající v této kapitole zpřístupněn jen přihlášeným uživatelům.

3.3.2.9 Nástěnka

Na nástěnku si uživatelé přidávají informace, které se mohou hodit každému. Jsou vypisovány od nejnovějšího po nejstarší

3.3.2.10 Nastavení

Zde si může uživatel poupravit většinu informací, jež zadal při registraci. Mezi tyto údaje patří například jméno, příjmení, adresa a podobně. Jsou to data, která se mohla skutečně změnit např. sňatkem, přejmenováním. Není povoleno měnit email, login nebo úplně vymazat editovatelná data.

3.3.2.11 Přidat téma

Funkce vytvoří nové téma. Podle nastaveného jazyka uloží jeho jméno do vhodného sloupce, kde se uloží patřičně zakódované.

3.3.2.12 Zobraz téma

Zobrazí všechna témata, která byla v aktuálně nastaveném jazyce vytvořena. Obdobně jako u funkce hledej, zohledňuje proměnnou stránkování, na jejímž základě pak vypisuje záznamy.

4 Implementace

Implementace obsahuje nejrůznější informace o tom, jak byla celá webová prezentace tvořena. Byla rozdělena do několika kapitol a v každé z nich se nachází několik ukázek.

4.1 Databáze

Celou databázi jsem vytvářel v programu phpMyAdmin. Tento nástroj slouží pro vytváření a správu databází na serveru MySQL. Byla použita verze 2.9.1.1, jelikož umí dostatečně využít nových vlastností MySQL. Nejdůležitější je právě možnost nastavení odlišných kódování pro různé sloupce. Přístup aplikace k databázi je prováděn pomocí skriptů PHP, proto tato oblast bude rozebrána až v kapitole skripty PHP.

V databázi jsou vytvořeny 4 tabulky. Tři jsou vzájemně provázány pomocí primárních a cizích klíčů, čtvrtá je samostatně.

4.1.1 Téma

Tabulka téma obsahuje 1 primární klíč (číslo) a 1 cizí klíč. Primární pro zachování entitní integrity, cizí, kterým je „založeno_kým“, je pak pro vyhledávání v tabulce uživatelů. Dále je zde deset sloupců, každý pro jiný jazyk. Použité znakové sady pro jméno uložené do jednoho z těchto sloupců jsou Windows 1250 pro češtinu, angličtinu a polštinu, Windows 1251 pro bulharštinu a ruštinu, Latin 1 pro němčinu, francouzštinu a švédštinu a UTF-8 pro zbylé což je japonština a čínština. Posledním sloupcem je datum_vytvoření, což je informace typu DATETIME kdy došlo z vytvoření tématu.

4.1.2 Příspěvek

Tato tabulka je v jistých ohledech velmi podobná tabulce téma. Obsahuje jeden primární klíč (číslo) a dva cizí (číslo_tématu, číslo_uživatele). Číslo_tématu ukazuje do tabulky téma a nese informaci, do kterého tématu článek patří. Číslo_uživatele, jak již jeho jméno naznačuje, odpovídá primárnímu klíči číslo v tabulce uživatel a ukazuje tak na uživatele, který příspěvek napsal. Datum_vložení je informace o tom, kdy byl článek vytvořen.

4.1.3 Uživatel

Tabulka uživatel obsahuje informace o uživateli, které o sobě vyplnil při registraci. Na rozdíl od dvou výše zmíněných tato tabulka je celá kódována v UTF-8. V návrhu databáze je zdůvodněno proč tomu tak je. Prvním sloupcem je opět primární klíč, jenž byl pojmenován jako číslo_uživatele. Dále je to jméno, příjmení, email.

Za více pozornosti stojí preferovaný jazyk, což je číslo od 1 do 10 a označuje jazyk, který se uživateli aktivuje pokaždé, když se přihlásí do systému.

Login, heslo a adresa jsou stejně jako jméno, příjmení a email uloženy v textové podobě a to pomocí UTF-8.

U pohlaví jsem obdobně jako u preferovaného jazyka volil k uložení informace číslo, neboť je to mnohem úspornější než text. V našem případě „0“ představuje muže a „1“ ženu.

Poslední položky jsou datum_registrace a datum_přihlášení. Datum_registrace obsahuje, jak již jméno naznačuje, datum a čas kdy se uživatel zaregistroval. Datum_přihlášení informuje kdy a v kolik hodin byl uživatel naposledy přihlášen.

4.1.4 Překlad

Tuto tabulku jsem si záměrně nechal jako poslední, neboť je v této databázi ojedinelá. Tabulka, nazvaná překlad, totiž obsahuje veškerý text, který jen je možné v celé aplikaci najít. Je to prostředek, jež umožňuje dynamické změny jazyků a kódových stránek. Každý řádek odpovídá jednomu nebo více místům v aplikaci, kam je za pomoci jazyka PHP a jeho proměnných vkládán jako text ze sloupců, jež odpovídají právě nastavenému jazyku. Podrobněji tento proces bude vysvětleno v sekci skriptů PHP. Jako každá má i tato primární klíč, který se opět jmenuje „číslo“.

4.2 Skripty PHP

Do této kapitoly budou zahrnuty i soubory záhlaví a zápatí, které využívají převážně jen jazyka HTML a úseky PHP kódu jsou zde spíše výjimkou.

4.2.1 Struktura stránky

Každá stránka je tvořena nejméně 3 soubory. Prvním je vždy soubor záhlaví, následuje jeden až několik souborů, které se starají o vlastní obsah stránky a ukončeno je to souborem zápatí. Aplikace je logicky rozdělena podle toho, zda je uživatel přihlášen či nikoliv, proto sem se rozhodl rozdělit i soubory záhlaví a zápatí na záhlaví_login, záhlaví_logout, zápatí_login a zápatí_logout. Pomocí těchto 4 souborů je tvořen veškerý vzhled aplikace.

4.2.2 Přepínání jazyků

Na začátku každého skriptu se zjišťuje informace, zda došlo ke klinutí na odkaz pro změnu jazyka (vlajka a popisek). Pokud ano, je nastavená proměnná jazyk v superglobálním poli proměnných „GET“ a s touto proměnnou zavolán původní skript. Ten se o přepnutí postará.

Na následujícím obrázku je dobře vidět, jakým způsobem ke kontrole dochází a co se poté děje. Vše bude vysvětleno o několik řádků níže.

```

if (isset($_GET['jazyk'])){
    $_SESSION['jazyk'] = $_GET['jazyk'];
    include("./includes/urci_znak.inc");
    if (isset ($_SESSION['preklad']))
        unset($_SESSION['preklad']);
}
include("./prijem.php");

```

Př. Kontrola přepnutí jazyka

Dojde-li tedy k požadavku na změnu jazyka, bude nastavená proměnná `$_GET['jazyk']` a tudíž i splněna podmínka, kterou můžeme vidět na obrázku. Zapamatujeme si nový jazyk do `$_SESSION['jazyk']` a vložíme soubor pro nastavení znakové sady a způsobu pozdější komunikace s databází MySQL. Toto je řešeno pomocí dvou switchů, které na základě zvoleného jazyka vyberou vhodné řetězce a uloží je do proměnných `$_SESSION['zn_sada']` a `$_SESSION['names']`. První ze zmíněných bude použit pro nastavení znakové sady v hlavičce stránky, druhý pak pro nastavení komunikace mezi skriptem PHP a MySQL serverem.

Do proměnné `$_SESSION['preklad']` bude vzápětí uložen veškerý text v, námi nově zvoleném, jazyce. Aby toto bylo umožněno, musíme nejprve odstranit starý obsah a dát tak najevo, že potřebujeme nový. O samotné naplnění této proměnné se už postará jiný skript – `prijem.php`. Zde ukážu jen jeho část.

```

if (! isset ($_SESSION['preklad'])){
    $dotaz = "SET NAMES ' ". $_SESSION['names']. " ' ";
    mysql_query($dotaz);
    $dotaz = "SELECT ' ". $_SESSION['jazyk']. " FROM preklad ";
    $vysledek = mysql_query ($dotaz);
    $i = 0;
    while ($radek = mysql_fetch_array($vysledek, MYSQL_NUM)){
        $_SESSION['preklad'][$i] = $radek[0];
        $i++;
    }
}

```

Př. Získání překladu pro aktuální stránku

Na obrázku je dobře patrně hlavní předpoklad pro naplnění proměnné `$_SESSION['preklad']`, což je, že tato proměnná není nastavena.

Nejprve dojde k nastavení komunikaci se serverem MySQL příkazem SET NAMES, poté načteme obsah všech řádků ve sloupci, jehož jméno je stejné, jako je právě nastavený jazyk.

4.2.3 Přihlašování

Přihlašování je velmi jednoduché. Po vyplnění uživatelského jména a hesla jsou tyto údaje zkontrolovány na zakázané řetězce, jako je například DROP TABLE. Vyhovují-li, odešle PHP skript informace na MySQL server a zjistí, zda opravdu existuje uživatel s těmito údaji. Pokud někoho najde, nastaví dvě proměnné \$_SESSION['user_id'] a \$_SESSION['user_login']. Ty pak fungují jako kontrola na stránkách, které vyžadují, aby byl uživatel přihlášen.

Nakonec se provede zápis do tabulky uživatel a je zde změněn datum a čas posledního přihlášení na aktuální.

```
$dotaz = "SELECT cislo_uzivatele, login, datum_prihlaseni FROM uzivatel WHERE
        (login = '$login' AND heslo = OLD_PASSWORD ('$heslo'))";
$vysledek = mysql_query ($dotaz);
if (mysql_num_rows($vysledek) != 1) {
...
$radek = mysql_fetch_array($vysledek);
$_SESSION['user_id'] = $radek['cislo_uzivatele'];
$_SESSION['user_login'] = $radek['login'];
...
$dotaz = "UPDATE uzivatel SET datum_prihlaseni = NOW()
WHERE cislo_uzivatele = '". $_SESSION['user_id']. "'";
$vysledek = mysql_query ($dotaz);
if (mysql_num_rows($vysledek) != 1) {
    echo $_SESSION['preklad'][24]; //Použití varovného textu z tabulky překlad
}
```

Př. Přihlášení uživatele a aktualizování záznamu posledního přihlášení

4.2.4 Odhlašování

Odhlašování je podstatně jednodušší, proto zde nebudu uvádět žádné ukázky. Stará se o něj skript nazvaný logout.php. Je-li spuštěn, pak dojde ke zrušení většiny údajů v superglobálním poli SESSION. Mezi tyto proměnné patří především user_id a user_login, jež jsou kontrolovány na všech stránkách s neveřejným přístupem. Každý skript aplikace navíc kontroluje proměnnou

\$_SESSION['time'], která v sobě nese informaci o datu a čase posledního načtení stránky. Je-li její rozdíl od aktuálního času větší než povolená hranice, dojde k automatickému přesměrování na logout.php a tím i k odhlášení.

4.2.5 Registrace

Pokud se návštěvník rozhodne k registraci, bude vyzván k vyplnění krátkého formuláře. Je nucen vyplnit především povinná data, která jsou červeně zvýrazněna. Po vyplnění a kliknutí na tlačítko uložit se jeho data zkontrolují. První předpoklad je, že žádný z textových polí nesmí obsahovat nepovolené řetězce jako je například DROP TABLE. Druhá podmínka je neduplicitní login a email. Třetí kontrolou je, zda byly vyplněny veškeré povinné údaje. Poslední má na starost, zda bylo heslo a jeho kontrola zadána stejně. Jsou-li splněny všechny čtyři podmínky, dojde k zaregistrování a uživateli je poslán email, kde je uvedeno jeho přihlašovací jméno a heslo. Tento mail je napsán v angličtině.

4.2.6 Nástěnka

Pro nástěnku nebylo třeba vytvářet novou tabulku v databázi. Využil jsem tabulek téma a příspěvek. V tabulce téma vznikl záznam, pojmenovaný jako nástěnka s primárním klíčem číslo = 3. Když pak je potřeba vypsat všechny články na nástěnce, použije se jednoduchá podmínka WHERE číslo = 3 při prohledávání tabulky příspěvky.

```
$dotaz = 'SELECT cislo_uzivatele, ' . $_SESSION['jazyk'] . ', dat_vlozeni FROM prispevky
WHERE (cislo_tematu=3 AND ' . $_SESSION['jazyk'] . ' <>''') ORDER BY
dat_vlozeni DESC';
```

Př. Vyhledávání článků nástěnky

Takto nalezené záznamy jsou seřazeny od nejnovějšího k nejstaršímu a jsou zobrazeny v tabulce, kde každý řádek obsahuje tři záznamy. Ke každému záznamu je navíc ukázán i datum vložení článku a na konci jméno uživatele, který jej tam vložil.

4.3 Ostatní

Do této kapitoly jsem vložil jedinou část aplikace, které nešla zařadit nikam jinak. Jedná se o krátký skript napsaný v jazyce JavaScript.

Ten slouží jako kontrola při registraci. Když uživatel zapomene nebo záměrně nevyplní některý nebo všechny povinné údaje nedojde k odeslání dat na server, nýbrž tento skript automaticky upozorní na tento nedostatek zprávou napsanou v anglickém jazyce. Skript je vytvořen hlavně

z demonstrativního důvodu, předvést, že je rychlejší zkontrolovat data přímo na straně klienta. Toto platí především u uživatelů s pomalejším internetem. Pro případ, že uživatel má vypnuty JavaScript funkce dojde k implicitní kontrole dat přímo na serveru. Uvedu zde krátkou ukázkou kontroly jednoho z údajů.

```
if (document.registruj.email.value.length <= 0){  
    alert ("email is missing.");  
    document.registruj.email.value = "*** @ ***";  
    document.registruj.email.focus();  
    vse_ok = 'Ne';  
}
```

Př. Kontrola zadaných údajů pomocí JavaScript.

5 Závěr

Výsledkem mé bakalářské práce je webová aplikace, která je schopná efektivně pracovat s deseti jazyky. Jedná se o fórum, kde se může kdokoliv zaregistrovat a poté komunikovat s kýmkoliv v jednom z těchto 10 implementovaných jazyků. Mezi zvolené jazyky patří: Čeština, Angličtina, Bulharština, Francouzština, Němčina, Polština, Ruština, Švédština, Japonština a Čínština. Vzhledem ke složitosti překladu do Čínštiny a Japonštiny je překlad u těchto dvou jazyků zaměněn za jiný náhodně vybraný Čínský nebo Japonský text.

Ke zhotovení tohoto díla jsem byl donucen hluboce se ponořit do tajů kódování a zde nalézt cestu, jak co nejjednodušeji vyjít vstříc všem požadavkům. Hlavním cílem zde bylo použít pro různé jazyky také různé, avšak vhodně zvolené, znakové sady.

Za vlastní přínos bych pak považoval fakt, že jsem vytvořil a používal přístup ke kódování tak jako asi nikdo přede mnou. Nejedná se o zvrát v celé problematice používání porovnávání, spíše o využití aktuálních možností a jejich demonstraci na půdě velmi často používané triády (PHP, MySQL, phpMyAdmin). Před tím, než se pokusím vystihnout další možné pokračování na tomto projektu, bych chtěl ještě naznačit, proč vlastně byl tento přístup lepší. Proč nepoužít všude UTF-8 a nestarat se tak o nějaké kódování. Hlavním důvodem je bezesporu úspora místa. Přesné výpočty nejsou moc možné, spíše bych se pokusil udělat odhad. Budeme uvažovat stejně časté používání jazyků. Čeština bude zabírat 1B/znak popř. 2B pro diakritiku. U polštiny to bude obdobně, angličtina se pravděpodobně nezmění vůbec. Na rozdíl Ruština a Bulharština bude kódována 3-4B/znak. U Japonštiny a Čínštiny je sice každý znak kolem 5-6B/znak, ale zde bychom příliš nešetřili použitím jiného porovnávání, neboť tyto dva jazyky zabírají sami o sobě více místa díky početnosti znaků.

Pokud tyto čísla zprůměrujeme, zjistíme, že bychom touto aplikací ušetřili přibližně polovinu až dvě třetiny místa. Pokud by pak tato aplikace byla opravdu používána v celosvětovém měřítku, byla by tato úspora velmi cenná.

5.1 Další možné pokračování

Jako další pokračování na tomto projektu bych si nejspíše představoval přepracování této aplikace na veřejně přístupnou šablonu. Ta by pak vývojářům, co se pokouší vytvořit vícejazyčnou webovou prezentaci, napomohla vypořádat se s problémy s tím spojených. Vývojář by si pak jen vybral jazyky a šablona by za něj implementovala přepínání jazyků, popř. vytvořila tabulky s patřičným kódováním. Komfort, který nám již dnes databázové systémy ohledně kódování nabízejí, je stále vyšší, přesto se domnívám, že by mnoho vývojářů tuto šablonu rádo použilo.

Literatura

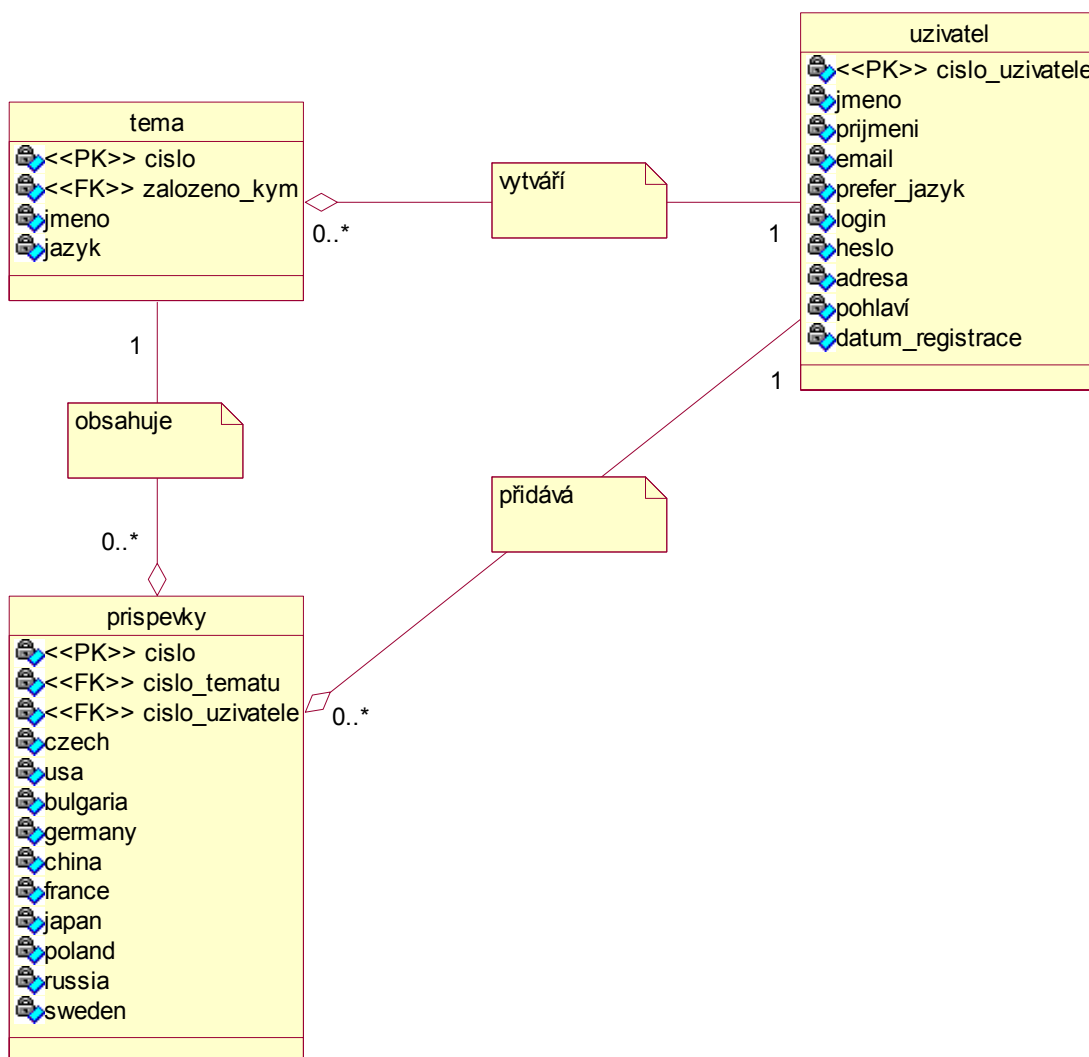
- [1] Ulman, Larry. PHP a MySQL, Brno, Computer Press 2004.
- [2] Kofler, Michael. Mistrovství v Mysql. Brno, Computer press, a.s. 2007
- [3] *Wikipedie: Otevřená encyklopedie*: HTML [online]. c2008 [20. 01. 2008].
Dostupné z WWW: < <http://cs.wikipedia.org/wiki/HTML> >
- [4] *Wikipedie: Otevřená encyklopedie*: Znaková sada [online]. c2008 [20. 01. 2008].
Dostupné z WWW: < http://cs.wikipedia.org/wiki/Znakov%C3%A1_sada >
- [5] *Wikipedie: Otevřená encyklopedie*: Unicode [online]. c2008 [20. 01. 2008].
Dostupné z WWW: < <http://cs.wikipedia.org/wiki/Unicode> >
- [6] *Wikipedie: Otevřená encyklopedie*: Čínština [online]. c2008 [20. 01. 2008].
Dostupné z WWW: < <http://cs.wikipedia.org/wiki/%C4%8C%C3%ADn%C5%A1tina> >
- [7] *MySQL: Oficiální stránky*: CONVERT [online]. c2008 [20. 01. 2008].
Dostupné z WWW: < <http://dev.mysql.com/doc/refman/5.0/en/charset-convert.html> >
- [8] *PHP: Český manuál*: [online]. c2008 [20. 01. 2008].
Dostupné z WWW: < <http://www.php.net/manual/cs/> >
- [9] *Wikipedie: Otevřená encyklopedie*: Japonština [online]. c2008 [20. 01. 2008].
Dostupné z WWW: < <http://cs.wikipedia.org/wiki/Japon%C5%A1tina> >

Seznam příloh

Příloha 1. Seznam obrázků

Příloha 2. CD/DVD ...

Seznam Obrázků



Obrázek 1, Předběžný ER Diagram

Mezinárodní diskuzní skupina

Czech Republic
 USA
 Bulgaria
 France
 Germany
 Poland
 Russia
 Sweden
 Japan
 China

Dnes je 20. 01. 2008 [Mapa serveru](#) [Novinky](#) [Nápověda](#) [Nepřihlášený uživatel](#)

Registrace

Hledej

Login

Heslo

Zapomněli jste heslo?

Jméno :

Příjmení :

Email : @

Preferovaný jazyk : English

Login :

Heslo :

Heslo kontrola :

Adresa :

Pohlaví : Muž Žena

Červeně jsou vyznačené povinné údaje.

Copyright ©Jaroslav Charvat

Obrázek 2 Př. Registrace uživatele v českém jazyce

Mezinárodní diskuzní skupina

Czech Republic
 USA
 Bulgaria
 France
 Germany
 Poland
 Russia
 Sweden
 Japan
 China

Today is 01/20/2008 [server map](#) [news](#) [help](#) [wawa](#)

| | created | topic name | created by | item (article) |
|----------------|----------|--|-----------------|----------------|
| To log out | 01/11/08 | English topic, dont show to other languages | Jarda Brzychudy | 0 |
| Bulletin board | 01/11/08 | second english topic, he he, how u doing lazy man..? | jenda Charvat | 0 |
| settings | | | | |
| Add topic | | | | |
| show topics | | | | |
| search | | | | |

Obrázek 3 Př. Zobrazení témat, anglický jazyk