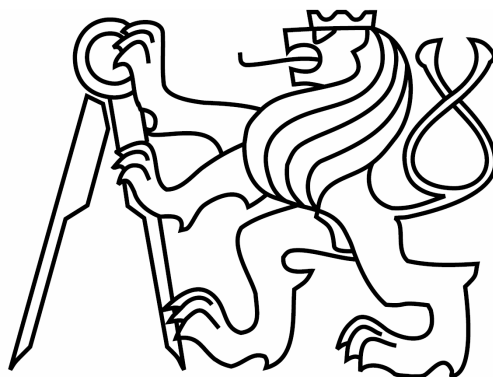


České vysoké učení technické v Praze  
Fakulta elektrotechnická



Bakalářská práce

**Rozšíření databáze konferencí**

Petr Galík

Vedoucí práce: Ing. Fišer Petr Ph.D.

Studijní program: Elektrotechnika a informatika, strukturovaný, bakalářský

Obor: Informatika a výpočetní technika

květen 2008



## **Poděkování**

Děkuji vedoucímu bakalářské práce Ing. Petru Fišerovi Ph.D. za jeho odborné rady, všestrannou pomoc a ochotu se kterou prací vedl.



### **Prohlášení**

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb.. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 14.5.2008

.....



## **Abstract**

The aim of this thesis is to extend web-based information system Database of Conferences on the management of users, adding the possibility of sending a personal e-mail notification, the improvement of discussion forum, preparation of citation data about conference for information system Database of Publications and the elimination of defects in original version.

Extension builds on technologies that were used by the original version of the system.

## **Abstrakt**

Cílem této práce je rozšířit webový informační systém Databáze konferencí o správu uživatelů, přidání možnosti osobního nastavení, zasílání emailových notifikací, zdokonalení diskusního fóra, příprava citačních údajů o konferenci pro informační systém Databáze publikací a odstranění nedostatků původní verze.

Rozšíření staví na technologiích, které používala původní verze systému.





## OBSAH

SEZNAM OBRÁZKŮ .....	XIII
SEZNAM TABULEK .....	XV
<b>1. ÚVOD .....</b>	<b>1</b>
<b>2. POUŽITÉ TECHNOLOGIE .....</b>	<b>2</b>
<b>2.1 Programovací jazyky .....</b>	<b>2</b>
2.1.1 HTML.....	2
2.1.2 CSS.....	2
2.1.3 PHP .....	2
2.1.4 SQL .....	3
2.1.5 JavaScript .....	3
<b>2.2 Serverové technologie .....</b>	<b>4</b>
2.2.1 Apache.....	4
2.2.2 MySQL.....	4
2.2.3 Cron.....	4
<b>3. SPECIFIKACE CÍLE .....</b>	<b>5</b>
<b>4. REALIZACE .....</b>	<b>7</b>
<b>4.1 Datové úložiště.....</b>	<b>7</b>
4.1.1 ER diagramy.....	7
4.1.2 Slovní popis datového úložiště.....	9
4.1.3 Normální formy.....	14
<b>4.2 Rozložení stránky .....</b>	<b>15</b>
<b>4.3 PHP skripty .....</b>	<b>16</b>
<b>4.4 JavaScripty.....</b>	<b>16</b>
<b>4.5 Popis programového řešení rozšíření .....</b>	<b>17</b>
4.5.1 Přihlášení a odhlášení uživatele.....	17
4.5.2 Náhrada výzkumných skupin za témata konferencí .....	17
4.5.3 Uživatelská práva .....	18
4.5.4 Posílání informativních emailů.....	18
4.5.5 Zdokonalení diskusního fóra .....	19
4.5.6 Propojení s Databází publikací.....	20
<b>4.6 Testování .....</b>	<b>20</b>
<b>5. BEZPEČNOST .....</b>	<b>21</b>
<b>6. INSTALAČNÍ MANUÁL.....</b>	<b>22</b>
<b>7. ZÁVĚR.....</b>	<b>24</b>
7.1 Výhled do budoucna .....	24
7.2 Shrnutí.....	24
<b>8. SEZNAM POUŽITÉ LITERATURY A OSTATNÍCH ZDROJŮ .....</b>	<b>25</b>
<b>9. PŘÍLOHY .....</b>	<b>26</b>
9.1 DDL skript pro vytvoření datového úložiště .....	26
9.2 Obsah příloženého DVD .....	29



## SEZNAM OBRÁZKŮ

Obrázek 1: ER diagram původního systému .....	7
Obrázek 2: ER diagram rozšířeného systému.....	8
Obrázek 3: Rozložení stránky.....	15



## SEZNAM TABULEK

Tabulka 1: Popis databázové tabulky <i>konference</i> .....	9
Tabulka 2: Popis databázové tabulky <i>tema</i> .....	10
Tabulka 3: Popis databázové tabulky <i>forum</i> .....	10
Tabulka 4: Popis databázové tabulky <i>uziv</i> .....	11
Tabulka 5: Popis databázové tabulky <i>konftema</i> .....	11
Tabulka 6: Popis databázové tabulky <i>osobkonf</i> .....	12
Tabulka 7: Popis databázové tabulky <i>temauziv</i> .....	13
Tabulka 8: Normální formy .....	14



## 1. ÚVOD

Databáze konferenci je dynamický webový informační systém, vytvořený v roce 2006 Bc. Janem Kemrem. Dynamičnost systému tkvěla v tom, že obsah byl vytvářen uživateli systému a byl ukládán do databáze.

Systém byl spuštěn na serveru Katedry počítačů a byl veřejně dostupný, což zajišťovalo velice snadný přístup odkudkoliv, kde byl počítač s připojením k internetu vybavený webovým prohlížečem.

Účelem systému je tedy udržovat přehlednou evidenci (seznam) konferencí vztahených ke *Katedře Počítačů FEL, ČVUT*. Pomoci něj mohli akademici navzájem sdílet a spravovat informace o konaných konferencích a diskutovat u každé konference v rámci jednoduchého diskusního fóra. U konference se při vytvoření nebo editaci označily výzkumné skupiny v rámci Katedry počítačů, do kterých konference spadala. Bylo možné zobrazovat konference, které spadají pouze do příslušné výzkumné skupiny.

Konference, které měly datum začátku starší než jeden měsíc nebo konference, které uživatel považoval za obsoletní, mohly být přesunuty do kategorie archivních konferencí. Tyto konference se zobrazovaly na samostatném seznamu.

Systém byl schopen rozpoznat identitu uživatele a poté zobrazovat část obsahu v závislosti na identitě přihlášeného. Tento obsah vyjadřoval uživatelovy osobní preference u jednotlivých konferencí a uživatel tyto konference viděl na odděleném seznamu. To urychlovalo a zpřehledňovalo jeho práci se systémem. Mohl si také nastavit počet přijatých a poslaných příspěvků na příslušnou konferenci. Na katedře počítačů, ale docházelo v době vývoje původní verze systému ke změně způsobu provádění autentizace uživatele v rámci školní sítě a proto část, která sloužila k určení identity uživatele nemohlo být plně dokončena.

Konference ve výše uvedených seznamech mohly být řazeny podle uživatelových preferencí. Ve všech konferencích bylo možné vyhledávat pomocí filtru, který hledal zadaný text v názvu a zkratce názvu konference.

Zobrazovaný obsah bylo také možné vytisknout. Systém na tiskárnu posílal jen informace s užitečnou hodnotou. Netiskly se tak například menu a záhlaví stránky.

Realizace systému byla provedena pomocí programovacího jazyka *PHP*, který generoval dynamicky obsah a byl schopen manipulovat s daty uloženými v databázi *MySQL*. Zobrazení generovaného obsahu bylo docíleno pomocí *CSS*, což je jazyk pro popis způsobu zobrazení stránky. V systému se prováděla dvojitá kontrola vstupních údajů zadaných uživatelem. Poprvé se prováděla na počítači uživatele pomocí programovacího jazyka JavaScript. To sice snížilo zátěž severu, ale uživatelův prohlížeč nemusí mít JavaScript povolen nebo ho nemusí podporovat. Podruhé na straně serveru před vlastním uložením uživatelem zadaných dat do databáze.

## 2. POUŽITÉ TECHNOLOGIE

Původní verze systému byla vytvořena technologiemi, jejichž základní popis uvádím v této kapitole. Zadavatelovo přání bylo použíté technologie bez výjimky zachovat, proto neprovádím porovnání s konkurenčními technologiemi.

### 2.1 Programovací jazyky

#### 2.1.1 HTML

*HyperText Markup Language*, označovaný zkratkou *HTML*, je značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému *World Wide Web*, který umožňuje publikaci dokumentů na Internetu.

Jazyk je aplikací dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka *SGML* (*Standard Generalized Markup Language*). Vývoj *HTML* byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.[12]

#### 2.1.2 CSS

*CSS* je zkratka pro anglický název *Cascading Style Sheets*, česky *tabulky kaskádových stylů*. Je to jazyk pro popis způsobu zobrazení stránek, napsaných v jazycích *HTML*, *XHTML* nebo *XML*.

Jazyk byl navržen standardizační organizací *W3C*, autorem prvotního návrhu byl Håkon Wium Lie. Byly vydány zatím dvě úrovně specifikace *CSS1* a *CSS2*, dokončuje se revize *CSS 2.1* a pracuje se na verzi *CSS3*.

Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk *HTML*, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze *HTML* obsahují celou řadu elementů, které popisují způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací nejsou tyto elementy žádoucí.

#### 2.1.3 PHP

*PHP* je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka *HTML*, *XHTML* či *WML*, což je velmi výhodné pro tvorbu webových aplikací. *PHP* lze ovšem také použít i k tvorbě konzolových a desktopových aplikací.

*PHP* skripty jsou prováděny na straně serveru, k uživateli je přenášén až výsledek jejich činnosti. Syntaxe jazyka kombinuje hned několik programovacích jazyků (*Perl*, *C*, *Pascal* a *Java*). *PHP* je nezávislý na platformě, skripty fungují bez úprav na mnoha různých operačních systémech. Obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k většině databázových serverů (mj. *MySQL*, *ODBC*, *Oracle*, *PostgreSQL*, *MSSQL*), podporu celé řady internetových protokolů (*HTTP*, *SMTP*, *SNMP*, *FTP*, *IMAP*, *POP3*, *LDAP*, ...)

*PHP* se stalo velmi oblíbeným především díky jednoduchosti použití a tomu, že kombinuje vlastnosti více programovacích jazyků a nechává tak vývojáři částečnou svobodu v syntaxi. V kombinaci s databázovým serverem (především s *MySQL* nebo *PostgreSQL*) a webovým serverem *Apache* je často využíván k tvorbě webových aplikací. Díky velmi častému nasazení na serverech se vžila zkratka *LAMP* – tedy spojení *Linux*, *Apache*, *MySQL* a *PHP* nebo *Perl*. S verzí *PHP 5* se výrazně zlepšil přístup k objektově orientovanému programování podobnému *Javě*. [1]



#### 2.1.4 SQL

*SQL* je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. *SQL* je zkratka anglických slov *Structured Query Language* (strukturovaný dotazovací jazyk).[2]

#### 2.1.5 JavaScript

*JavaScript* je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti *Netscape*.

Nyní se zpravidla používá jako interpretovaný programovací jazyk pro *WWW* stránky, často vkládaný přímo do *HTML* kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky *GUI* (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků. Je spouštěn na straně klienta.

Jeho syntaxe patří do rodiny jazyků *C/C++/Java*. Slovo *Java* je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem *Java* jej vedle názvu spojuje jen podobná syntaxe. *JavaScript* byl v červenci 1997 standardizován asociací *ECMA* (*European Computer Manufacturers Association*) a v srpnu 1998 *ISO* (*International Organization for Standardisation*). Standardizovaná verze *JavaScriptu* je pojmenována jako *ECMAScript* a z ní byly odvozeny i další implementace, jako je například *ActionScript*. [3]

## 2.2 Serverové technologie

### 2.2.1 Apache

*Apache HTTP Server* je softwarový webový server s otevřeným kódem pro *Linux*, *BSD*, *Microsoft Windows* a další platformy. V současné době dodává prohlížečům na celém světě většinu internetových stránek. Název vznikl z anglického slovního spojení „A patchy server“ (záplatovaný server).[4]

### 2.2.2 MySQL

*MySQL* je databázový systém, vytvořený švédskou firmou *MySQL AB*. Jeho hlavními autory jsou Michael Widenius a David Axmark. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licenci *GPL*, tak pod komerční placenou licenci.

*MySQL* je multiplatformní databáze. Komunikace s ní probíhá pomocí jazyka *SQL*. Podobně jako u ostatních *SQL* databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

Pro svou snadnou implementovatelnost (lze jej instalovat na *Linux*, *MS Windows*, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šířitelný software, má vysoký podíl na používaných databázích.

*MySQL* bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, triggerů a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet. [5]

### 2.2.3 Cron

*Cron* je časově závislá plánovací služba dostupná na *Unixových* systémech. Název této služby byl odvozen od Řeckého *chronos*, což v češtině znamená čas. *Cron* byl v minulosti několikrát přepsán.

Služba *cron* je řízena *crontabem*. *Crontab* je konfigurační soubor, který specifikuje *shellové* příkazy, které se mají spouštět periodicky v čase. Od verze *Unix System V* je služba dostupná všem uživatelům nikoliv pouze super uživateli. Pokud administrátor nespécifikuje jinak, má každý uživatel svůj vlastní seznam úloh. Minimální časový interval mezi spuštěními je jedna minuta.[6]

### **3. SPECIFIKACE CÍLE**

#### **Náhrada výzkumných skupin za témata**

V systému by mělo být odstraněno řazení konferencí do výzkumných skupin a mělo by být nahrazeno řazením do témat konference. Uživatel by měl mít možnost filtrovat pouze konference s danými tématy. Dále by měl mít možnost si přednastavit témata, o která má zájem. Konference, které by spadaly do těchto témat by se zobrazovaly jako výchozí ve všech třech seznamech tj. *seznam*, *osobní* i *archiv*.

#### **Uživatelská práva**

V systému by měla být zakomponována uživatelská práva pro každou činnost, při které se manipuluje s obsahem systému. Nepřihlášený uživatel by měl mít pouze možnost prohlížet obsah vytvořený přihlášenými uživateli. Se zadavatelem jsme se shodli na následujících právech, která může nabývat přihlášený uživatel:

- vytváření nových konferencí
- mazání a úprava existujících konferencí
- správa uživatelů, uživatelských práv a témat
- možnost přispívat do diskusního fóra konferencí

#### **Posílání informativních emailů**

Uživatel by měl mít možnost nechat si zasílat informativní emaily vztažené k časovým údajům jednotlivých konferencí (deadline zaslání příspěvků, finální verze příspěvků, začátek konference). Uživatel by měl také mít možnost si nechat tato upozornění posílat s předstihem, který mu vyhovuje a měl by mít možnost nastavit si předstih výchozí.

Dále by měla být možnost nastavení posílání informativních emailů při editaci konference nebo při provedení změny v diskusním fóru konference s určitým tématem.

#### **Zdokonalení fóra**

V diskusním fóru každé konference by se měl zobrazovat přehled příspěvků zobrazující jejich hierarchickou vazbu s možností zobrazení všech nebo výběru zobrazovaných příspěvků. Diskusní fórum by se mělo ovládat a mělo by vypadat uživatelsky přívětivěji.

#### **Výpis statistik**

Uživatel by měl v osobním nastavení mít možnost ve svých osobních konferencích vidět počet poslaných a přijatých příspěvků pro danou konferenci a v seznamu by se měla zobrazovat tato čísla také, ale jako součet od všech uživatelů.

## **Propojení s informačním systémem Databáze publikací**

Informační systém by měl generovat u každé konference citační údaje o konferenci. Tyto údaje by měli být generovány ve formátu BibTeX a uživatel by je měl ručně přesunout do informačního systému Databáze publikací. Jednalo by se tedy pouze o jednosměrné propojení.

## **Závěrné úpravy**

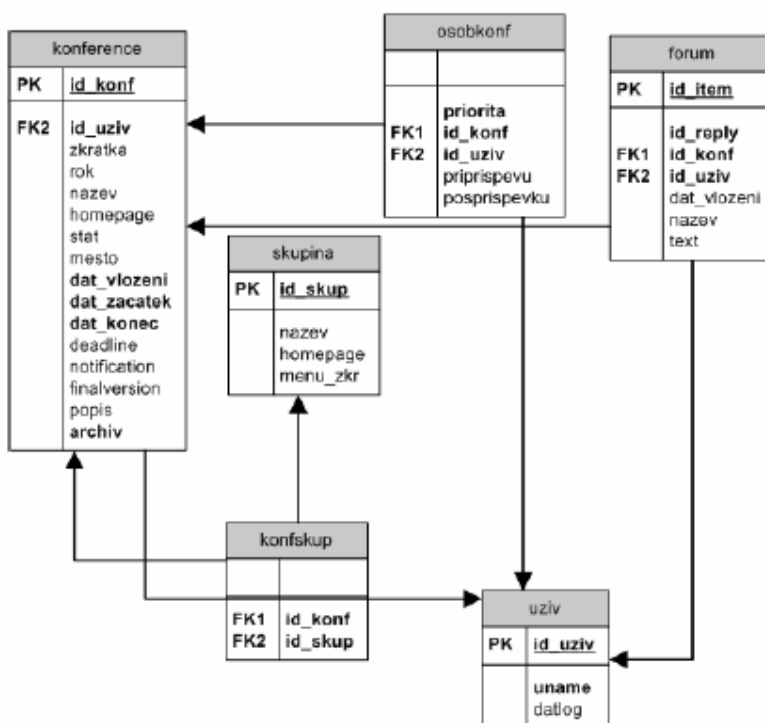
Výsledný produkt by měl odstraňovat nedostatky předchozí verze a měla by být provedena validace do HTML 4.01 Strict.

## 4. REALIZACE

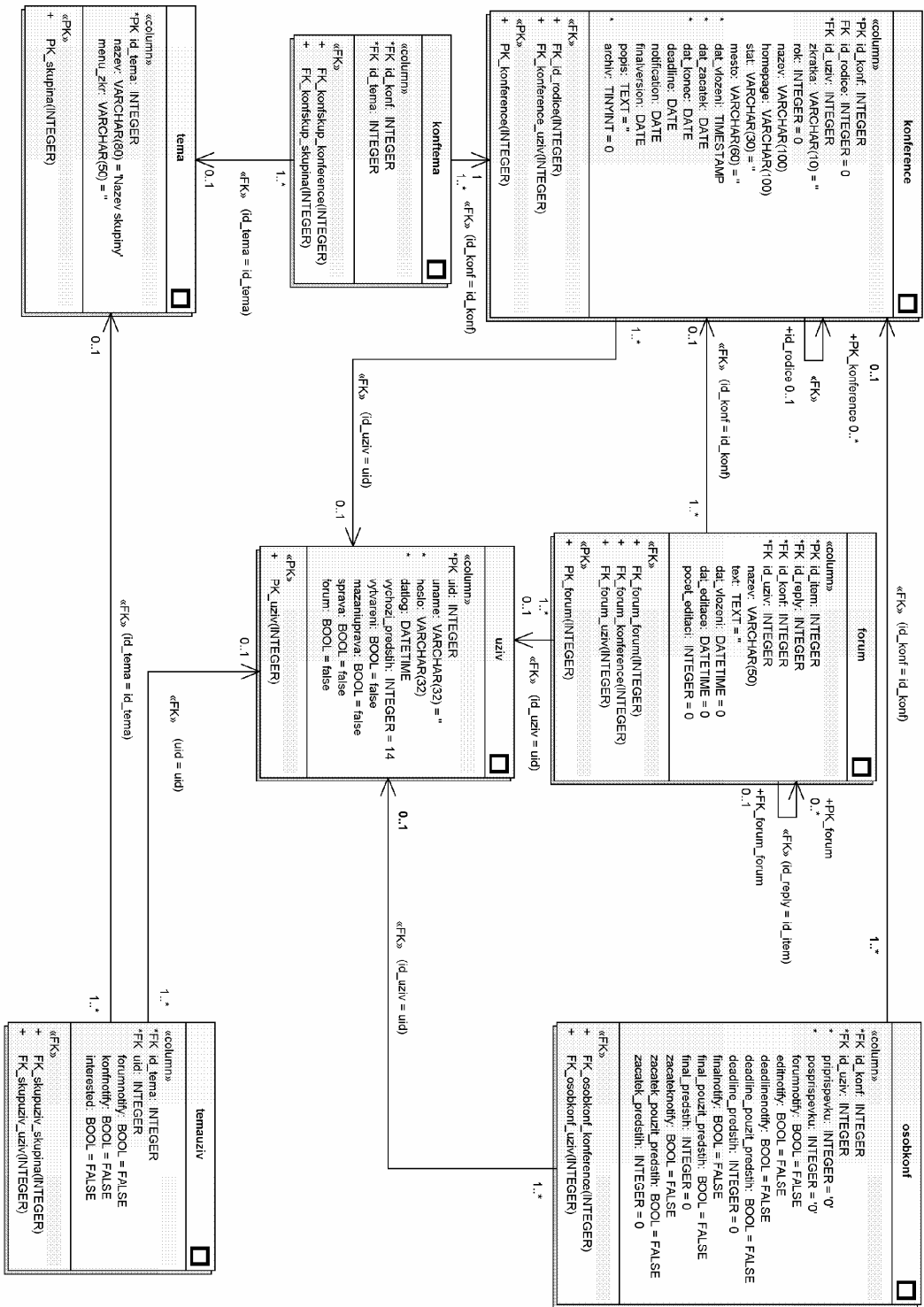
### 4.1 Datové úložiště

V této kapitole popíšeme schéma datového úložiště a jeho vztah k původnímu řešení.

#### 4.1.1 ER diagramy



Obrázek 1: ER diagram původního systému



Obrázek 2: ER diagram rozšířeného systému

#### 4.1.2 Slovní popis datového úložiště

ER diagram rozšířeného systému vychází z ER diagramu původního systému. Z původního řešení byly ponechány tři tabulky a to *konference*, *forum* a *uziv*. K nejvýznamnějším změnám proti původnímu řešení je přejmenování tabulky *skupina* na *tema* a tabulky *konfskup* na *konftema*. To reflektuje odstranění výzkumných skupin a náhradu za témata konferencí. Dále přibyla tabulka *temauziv*, která vyjadřuje uživateli preference k danému tématu. Také do tabulek přibýlo množství nových atributů. V textu níže je uveden slovní popis jednotlivých atributů tabulek. Atributy psané *kurzivou* jsou ponechány z původního řešení se stejným významem.

#### Tabulka *konference*

Tabulka *konference* slouží k evidenci informací o konferenci.

Atribut	Datový typ	Slovní popis
<i>id_konf</i>	INTEGER	Primární klíč tabulky identifikující konferenci.
<i>id_rodice</i>	INTEGER	Cizí klíč identifikující předka konference. Pokud konference vynikla jako obnovení existující konference nastaví se tento atribut. Jinak je nulový.
<i>id_uziv</i>	INTEGER	Cizí klíč identifikující uživatele, který konferenci vytvořil.
<i>zkratka</i>	VARCHAR(10)	Zkratka konference se zobrazuje v seznamu konferencí.
<i>rok</i>	INTEGER	rok konání konference
<i>nazev</i>	VARCHAR(100)	název konference
<i>homepage</i>	VARCHAR(100)	Obsahuje adresu oficiální homepage konference.
<i>stat</i>	VARCHAR(30)	Stát ve kterém se konference koná.
<i>mesto</i>	VARCHAR(60)	Město ve kterém se konference koná.
<i>dat_vlozeni</i>	TIMESTAMP	Časové razítko určující vytvoření konference.
<i>dat_zacatek</i>	DATE	datum začátku konference
<i>dat_konec</i>	DATE	datum konce konference
<i>deadline</i>	DATE	datum deadline zaslání příspěvků
<i>notification</i>	DATE	Datum určující, kdy budou zaslány emaily o (ne)přijetí příspěvků.
<i>finalversion</i>	DATE	Datum do kterého mají být vytvořené finální verze příspěvků.
<i>popis</i>	TEXT	slovní popis konference
<i>archiv</i>	TINYINT(1)	Určuje, zda se jedná o archivní konferenci.

Tabulka 1: Popis databázové tabulky *konference*

### Tabulka *tema*

Tabulka *tema* slouží k evidenci témat do kterých se mohou konference zařazovat.

Atribut	Datový typ	Slovní popis
<i>id_tema</i>	INTEGER	Primární klíč tabulky identifikující téma.
<i>nazev</i>	VARCHAR(80)	název tématu
<i>menu_zkr</i>	VARCHAR(50)	Zkratka názvu tématu se zobrazuje ve filtru témat.

Tabulka 2: Popis databázové tabulky *tema*

### Tabulka *forum*

Tabulka *forum* slouží k evidenci příspěvků v diskusních fórech konferencí.

Atribut	Datový typ	Slovní popis
<i>id_item</i>	INTEGER	Primární klíč tabulky identifikující příspěvek v diskusním fóru.
<i>id_reply</i>	INTEGER	Pokud je příspěvek reakce na jiný příspěvek nastaví se na <i>id_item</i> příspěvku na který odpovídá. Jinak se <i>id_reply</i> nastaví na <i>id_item</i> .
<i>id_konf</i>	INTEGER	Cizí klíč identifikující konferenci, ke které se příspěvek vztahuje.
<i>id_uziv</i>	INTEGER	Cizí klíč identifikující uživatele, který příspěvek vytvořil.
<i>nazev</i>	VARCHAR(50)	název příspěvku
<i>text</i>	TEXT	text příspěvku
<i>dat_vlozeni</i>	DAYTIME	datum a čas vložení příspěvku
<i>dat_editace</i>	DAYTIME	datum a čas poslední editace
<i>pocet_editaci</i>	INTEGER	celkový počet editací daného příspěvku

Tabulka 3: Popis databázové tabulky *forum*



### Tabulka *uziv*

Tabulka *uziv* slouží k evidenci registrovaných uživatelů systému.

Atribut	Datový typ	Slovní popis
<i>uid</i>	INTEGER	Primární klíč tabulky identifikující uživatele. V původním diagramu je označen <i>id_uziv</i> , nicméně ve všech skriptech je pojmenován <i>uid</i> .
<i>name</i>	VARCHAR(32)	jméno uživatele
<i>heslo</i>	VARCHAR(32)	MD5 otisk uživatelského hesla
<i>datlog</i>	DAYTIME	Datum a čas posledního přihlášení do systému
<i>vych_predstih</i>	INTEGER	výchozí předstih pro zaslání informativních emailů
<i>vytvareni</i>	BOOL	Je nastaveno, pokud má uživatel právo na vytváření konferencí.
<i>mazaniuprava</i>	BOOL	Je nastaveno, pokud má uživatel právo na mazání a úpravu konferencí.
<i>sprava</i>	BOOL	Je nastaveno, pokud má uživatel právo na správu uživatelů, uživatelských práv a témat.
<i>forum</i>	BOOL	Je nastaveno, pokud má uživatel právo na přispívání do diskusního fóra konferencí.

Tabulka 4: Popis databázové tabulky *uziv*

### Tabulka *konftema*

Tabulka *konftema* realizuje vazbu M:N mezi tabulkou *konference* a *tema*. Konference může spadat do několika témat. Téma může být stejné u více konferencí.

Atribut	Datový typ	Slovní popis
<i>id_konf</i>	INTEGER	Cizí klíč <i>id_konf</i> vztažený k tabulce <i>konference</i> .
<i>id_tema</i>	INTEGER	Cizí klíč <i>id_tema</i> vztažený k tabulce <i>tema</i> .

Tabulka 5: Popis databázové tabulky *konftema*

### Tabulka *osobkonf*

Tabulka *osobkonf* slouží k evidenci osobního nastavení uživatelů vztahujících se k jednotlivým konferencím.

<b>Atribut</b>	<b>Datový typ</b>	<b>Slovní popis</b>
id_konf	INTEGER	Cizí klíč tabulky identifikující konferenci ke které se osobní nastavení váže.
id_uziv	INTEGER	Cizí klíč identifikující uživatele ke kterému se nastavení konference váže.
priprispevku	INTEGER	Počet příspěvků, které byly uživateli na příslušnou konferenci přijaty.
posprispevku	INTEGER	Počet příspěvků, které uživatel na příslušnou konferenci poslal.
forumnotify	BOOL	Je nastaveno, pokud chce být uživatel upozorněn na změny ve fóru dané konference..
editnotify	BOOL	Je nastaveno, pokud chce být uživatel upozorněn při editaci konference.
deadlinenotify	BOOL	Je nastaveno, pokud chce být uživatel upozorněn na deadline odeslání příspěvků konference.
deadline_pouzit_predstih	BOOL	Je nastaveno, pokud chce uživatel použít jiný než výchozí předstih pro upozornění na deadline poslání příspěvků.
deadline_predstih	INTEGER	Předstih ve dnech, který se použije, když je nastaveno <code>deadline_pouzit_predstih</code> .
finalnotify	BOOL	Je nastaveno, pokud chce být uživatel upozorněn na datum, do kterého se mají poslat finální verze příspěvků.
final_pouzit_predstih	BOOL	Je nastaveno, pokud chce uživatel použít jiný než výchozí předstih pro upozornění na finální verzi příspěvků.
final_predstih	INTEGER	Předstih ve dnech, který se použije, když je nastaveno <code>final_pouzit_predstih</code> .
zacatek_notify	BOOL	Je nastaveno, pokud chce být uživatel upozorněn na začátek konference.
zacatek_pouzit_predstih	BOOL	Je nastaveno, pokud chce uživatel použít jiný než výchozí předstih pro upozornění na začátek konference.
zacatek_predstih	INTEGER	Předstih, který se použije, když je nastaveno <code>zacatek_pouzit_predstih</code> .

Tabulka 6: Popis databázové tabulky *osobkonf*

### Tabulka *temauziv*

Tabulka *temauziv* realizuje vztah M:N mezi tabulkami *uziv* a *tema*. Jeden uživatel si tedy může nastavit osobní preference u několika témat a téma může být nastaveno více uživateli.

<b>Atribut</b>	<b>Datový typ</b>	<b>Slovní popis</b>
id_tema	INTEGER	Cizí klíč id_tema vztažený k tabulce <i>tema</i> .
uid	INTEGER	Cizí klíč uid vztažený k tabulce <i>uziv</i> .
forumnotify	BOOL	Je nastaveno, pokud chce uživatel být upozorňován na změny v diskusních fórech konferencí s daným tématem.
konfnotify	BOOL	Je nastaveno, pokud chce uživatel být upozorňován na vytvoření nebo editaci konference s daným tématem.
interested	BOOL	Je nastaveno, pokud uživatele dané téma zajímá. Tato témata budou automaticky vybrána v zobrazovacím filtru témat.

Tabulka 7: Popis databázové tabulky *temauziv*

### 4.1.3 Normální formy

Pod pojmem normalizace rozumíme proces zjednodušování a optimalizace navržených struktur databázových tabulek. Hlavním cílem je navrhnout databázové tabulky tak, aby obsahovaly minimální počet redundantních dat. Správnost navržení struktur lze ohodnotit některou z následujících normálních forem.

Normální forma	Požadavek pro splnění příslušnosti
Nultá	Tabulka v nulté normální formě obsahuje alespoň jeden sloupec (atribut), který může obsahovat více druhů hodnot.
První	Tabulka je v první normální formě, pokud všechny sloupce (atributy) nelze dále dělit na části nesoucí nějakou informaci. Jeden sloupec neobsahuje složené hodnoty.
Druhá	Tabulka je v druhé normální formě, pokud obsahuje pouze atributy (sloupce), které jsou závislé na celém klíči.
Třetí	Tabulka je ve třetí normální formě, pokud neexistují žádné závislosti mezi neklíčovými atributy (sloupci).
Čtvrtá	Tabulka je ve čtvrté normální formě, pokud sloupce (atributy) v ní obsažené popisují pouze jeden fakt nebo jednu souvislost.
Pátá	Tabulka je v páté normální formě, pokud by se přidáním libovolného nového sloupce (atributu) rozpadla na více tabulek. [7]
Boyce-Coddova	Tabulka je v Boyce-Coddově normální formě, jestliže pro každou netriviální závislost $X \rightarrow Y$ platí, že $X$ obsahuje klíč schématu $R$ .

Tabulka 8: Normální formy

Datové úložiště současného řešení je ve druhé normální formě, protože obsahuje pouze atributy, které jsou závislé na celém klíči. Třetí normální formu už nesplňuje, protože v tabulce *osobkonf* jsou závislosti mezi neklíčovými atributy. Například atribut *zacatek\_predstih* je závislý na *zacatek\_pouzit\_predstih* a ten je závislý na atributu *zacateknotify*. Boyce-Coddovu normální formu splňuje.

## 4.2 Rozložení stránky

Umístění jednotlivých částí stránky vychází z původního řešení a bylo jej dosaženo pomocí Tabulky kaskádových stylů, které jsou definovány v souboru *style.css*, umístěném v adresáři *css*. Při tisku se uplatní styl ze souboru *print.css*, umístěný ve stejném adresáři. Pomocí tohoto stylu jsou nastýlovány části stránky, které nejsou určené pro tisk. Jedná se například o záhlaví, hlavní menu a levé menu.

Záhlaví	
Hlavní menu	
Levé menu	Obsahová část
Zápatí	

Obrázek 3: Rozložení stránky

Největší změny byly provedeny v levém menu, které dříve obsahovalo seznam výzkumných skupin (po kliknutí na skupinu se zobrazily konference, které spadaly do příslušné skupiny) a pod ním se zobrazovalo 5 *nejaktuálnějších* konferencí, jejichž datum začátku bylo nejbližší k aktuálnímu datu. Nyní se na tomto místě zobrazuje menu, které může obsahovat *filtr témat*, menu s relevantními příkazy pro aktuální stránku (pojmenované *další akce*) a 5 *nejaktuálnějších* konferencí.

O zobrazení stránky se stará stejně jako v původním řešení skript *index.php*. Tento skript plní následující funkce:

- Zobrazení záhlaví stránky, hlavního menu, levého menu a zápatí.
- Vložení příslušného PHP skriptu, který generuje obsahovou (pravou) část stránky v závislosti na nastavených proměnných (například při nastavené proměnné *all\_users* se vloží soubor *sprava.php* a ten zde zobrazí seznam uživatelů).
- Pamatování nastavených témat *filtru témat* a případně vyhledávaného řetězce. Toto původní verze systému neprováděla.

Obsahové stránky před prováděním jakékoliv činnosti provádějí kontrolu, zda byly vloženy ze souboru *index.php*, který nastaví hodnotu proměnné *indexphp* na *true*. V případě nesplnění této podmínky se zobrazí seznam konferencí. Tento skript vypadá takto:

```
if(!isset($indexphp) || (isset($indexphp) && !$indexphp) || isset($_GET["indexphp"]))  
    echo("<script type='text/javascript'>document.location='index.php';</script>");
```

### 4.3 PHP skripty

*PHP* skripty vycházejí z původního řešení, které používá funkce pro často prováděné činnosti. Tyto funkce jsou definovány v souboru *fun.php* a funkce pro práci s databází jsou definovány v souboru *db.php*. Obsah obou těchto dvou souborů je vždy viditelný, protože je vložen do souboru *index.php*.

### 4.4 JavaScripty

Java skripty jsou v systému využívány ke kontrole vstupních dat od uživatele. Tyto skripty jsou vkládány do příslušných php skriptů a jsou až na koncovku(.js) pojmenovány stejně. Kupříkladu skript *forum.js* obsahuje kontrolní funkci *ZkontrolujPrispevek*, která je zavolána při odesílání (*onsubmit*) formuláře s novým příspěvkem. Pokud je kontrola v pořádku, funkce vrátí *true* a formulář se odešle na server, jinak se formulář neodešle a zobrazí se upozornění. Kontrolu pak ještě samozřejmě provádí *PHP* skript pro případ, že by uživatelův prohlížeč měl zakázaný *JavaScript* nebo ho nepodporoval. Funkce *ZkontrolujPrispevek* vypadá takto:

```
function ZkontrolujPrispevek()
{
    var Vysledek = true;
    SeznamChyb = "";

    if(document.forum_form.nazev.value.length < 2)
    {
        Vysledek = false;
        SeznamChyb = SeznamChyb + "* Název příspěvku musí mít minimálně dva znaky.\n";
    }

    if(document.forum_form.text.value.length < 2)
    {
        Vysledek = false;
        SeznamChyb = SeznamChyb + "* Text příspěvku musí mít minimálně dva znaky.\n";
    }

    if (!Vysledek)
        alert(SeznamChyb);

    return Vysledek;
}
```

Dále se využívají pro manipulaci s elementy formuláře. Například jde o manipulaci s inputboxy, kterými se zadávají datумы vztažené ke konferenci či provedení inverze checkboxů identifikujících témata, do které se nově vytvořená/editovaná konference zařadí.

## 4.5 Popis programového řešení rozšíření

### 4.5.1 Přihlášení a odhlášení uživatele

Ihned po příchodu je vytvořena pro uživatele session voláním funkce *session\_start*. Dále vložený skript *uziv.php* nastaví `$_SESSION["uid"]` na nulu a `$_SESSION["uname"]` nastaví na *nepřihlášen*. Při úspěšném přihlášení je skript *login.php* upraví podle přihlášeného uživatele. Při odhlášení se vrátí do původních hodnot.

### 4.5.2 Náhrada výzkumných skupin za témata konferencí

Vypisování seznamu konferencí má na starosti skript *seznam.php*. Módy zobrazení jsou zobrazení aktuálních konferencí (položka horního menu Seznam), osobních konferencí (položka horního menu Osobní), archivních konferencí (položka hlavního menu Archiv) nebo se může jednat o výsledky vyhledávání (položka vyhledat v hlavním menu).

Mód zobrazení dostane tento skript již přednastavený v `_SESSION` proměnné pod klíčem *mod*. Témata, do kterých mohou konference spadat, dostane také přednastavené pod klíči *tema\_X*, kde *X* je hodnota atributu tabulky *tema id\_tema*. Důvod, proč tyto proměnné nastavuje do proměnné `_SESSION` skript *index.php*, je ten, že seznam konferencí zobrazuje konference po stránkách a při každém přechodu ze stránky na stránku by se musela posílat všechna zobrazovaná témata. Z těchto informací se sestaví SQL dotaz, který vybírá příslušné konference a ty poté zobrazí.

Výběrový dotaz vypadá takto:

```
SELECT * FROM konference k WHERE
```

```
// Uplatnění filtrace témat – Počet témat, které jsou vyžadovány, může být  
// proměnlivé množství. Zde jsou uvedeny ilustrativně tři a to X, Y a Z.  
EXISTS (SELECT * FROM konftema ks WHERE  
ks.id_tema=X OR ks.id_tema=Y OR ks.id_tema=Z AND ks.id_konf=k.id_konf )
```

```
// Pokud se mají zobrazit archivní konference  
AND k.archiv=true
```

```
// Pokud se mají zobrazit osobní konference  
AND (SELECT COUNT(*) FROM osobkonf ok WHERE k.id_konf=ok.id_konf AND  
ok.id_uziv=$_SESSION["uid"])=1
```

### 4.5.3 Uživatelská práva

Uživatelská práva jsou uložena pro každého uživatele v tabulce *uziv*. Pokud uživatel provádí akci, pro kterou je potřeba nějaké oprávnění, zkontrolují se uživatelská práva pomocí funkcí definovaných v souboru *fun.php*. Tento soubor je vždy vložen do souboru *index.php*, tudíž jsou jeho funkce vždy viditelné. Funkce se jmenují *muze\_vytvaret*, *muze\_mazat\_upravovat*, *muze\_spravovat* a *muze\_prispivat\_do\_fora*. Všechny tyto funkce zkontrolují příslušná práva pro id uživatele, které je uloženo v proměnné *\_SESSION* pod klíčem *uid* a vrátí výsledek jako boolean proměnnou.

### 4.5.4 Posílání informativních emailů

Informativní emaily se zasílají uživatelům systému na adresu *uzivatelske\_jmeno@domena*, kde *uzivatelske\_jmeno* je hodnota atributu *uname* z tabulky *uziv* a *domena* je proměnná prostředí `$_ENV["emailova_domena"]` definovaná v souboru *fun.php*. Toto rozložení je výhodné například v rámci fakulty. Emaily se posílají z adresy *konference@domena*. Emaily jsou posílány pomocí knihovni funkce *PHP mail*. Tato funkce je volána s následujícími parametry a pořadím:

- *prijemci* - seznam příjemců oddělený čárkou
- *predmet* – zakódovaný předmět, kvůli správnému zobrazení češtiny
- *zprava* – informativní zpráva, kterou zakončují upozorněním, aby na tento mail nikdo neodpovídal
- *hlavicky* – informace o kódování textu zprávy a emailovém klientovi, který zprávu odeslal

Před ostrým nasazením systému je potřeba se ujistit, že je vypnutý debug mód zkontrolováním proměnné prostředí `$_ENV["debug_mode"]` definované v souboru *fun.php*. Pokud by byl zapnutý, k posílání emailů by nedocházelo. Pouze by se při dané akci vypisovaly zprávy do stánky.

### Emaily vztažené k události vyvolané uživateli

Emaily vztažené k události vytvoření nebo editace konference jsou zaslány skriptem *konf.php* a při provedení změny v diskusním fóru konference skriptem *forum.php*. Části odesílající informativní emaily fungují velice podobně a proto naznačím programové řešení-poslání emailů při vytvoření konference:

1. Vytvoření textu zprávy obsahujícího informace o nové konferenci
2. Vytvoření seznamu příjemců, kteří mají nastaveno upozorňování na nové konference nebo editaci existujících konferencí u tématu do kterého nová konference spadá
3. Nastavení předmětu a jeho zakódování
4. Nastavení hlaviček
5. Zavolání funkce *mail* s připravenými parametry



## Emaily vztahené k časovým údajům

Emaily vztahené k časovým údajům jednotlivých konferencí posílá skript `script_pro_planovac.php`. Tento skript by měl být před veřejností ukrytý a měl by být zaregistrovaný například pomocí programu Cron (jeho popis je v sekci Serverové technologie) pro spouštění jednou denně.

Tento skript obsahuje funkci *PosliUpozorneni*, která přebírá číselný parametr identifikující událost (0=deadline odesílání příspěvků, 1=finální verze příspěvků, 2=datum začátku konference). Funkce odešle všechny emaily vázané na tuto událost. Je spouštěna třikrát s parametry 0 až 2.

Jako zajímavou část považuji výběrový *SQL* dotaz, jehož jeden výsledkový řádek obsahuje všechny potřebné dodatečné informace pro zaslání jednoho emailu. Zaslání všech emailů se provádí průchodem celého výsledkového pole po řádcích.

```
SELECT u.uname, k.zkratka, k.rok FROM osobkonf ok, konference k, uziv u WHERE
ok.$atribut_osobkonf_1=true AND ok.id_konf=k.id_konf AND ok.id_uziv=u.uid AND
(TO_DAYS(k.$atribut_konference) +IF(ok.$atribut_osobkonf_2, ok.$atribut_osobkonf_3,
u.vychazi_predstih) ) = TO_DAYS(CURRENT_DATE())
```

Dotaz tedy vybere uživatelské jméno, zkratku konference, rok konání konference a pokud má uživatel nastaveno v osobním nastavení konference (tabulka *osobkonf*) upozornění na danou událost (název události obsahuje proměnná *atribut\_osobkonf\_1*). Dále musí platit, že datum události konference (název události konference je uložen v proměnné *atribut\_konference*) plus výchozí předstih nebo jiný uživatelem specifikovaný předstih pro danou událost (název předstihu pro danou událost je uložen v proměnné *atribut\_osobkonf\_3*) se rovná aktuálnímu datu. To zda se použije výchozí předstih nebo jiný uživatelem specifikovaný předstih je zjištěno podmínkou IF.

### 4.5.5 Zdokonalení diskusního fóra

Diskusní fórum je generováno skriptem *forum.php*. Tento skript obsahuje dvě rekurzivní funkce *ZobrazPrispevky* a *ZobrazPrehledPrispevku*. Zbytek skriptu se stará o zobrazení a zpracování přidání nového příspěvku, editaci příspěvku a přidání odpovědi na příspěvek.

Obě uvedené funkce mají parametr *Klic*, který identifikuje příspěvek, na který jsou reakcí. Při prvním voláním je tento parametr nula, protože se jedná o kořenové příspěvky. Obě funkce zobrazí příspěvky na příslušné úrovni (určuje odsazení) identifikované parametrem *Uroven* a rekurzivně se volají pro zobrazení zanořených úrovní.

První funkce *ZobrazPrispevky* rekurzivně zobrazí příspěvky v plném znění k dané konferenci (identifikované parametrem *ID\_Konference*) a odsadí jednotlivé příspěvky pro zdůraznění jejich hierarchie (parametr funkce *Uroven*). Ke každému příspěvku se může zobrazit tlačítko *Editovat* (pokud je uživatel vlastník příspěvku) nebo *Odpovědět* (pokud má uživatel právo přispívání do diskusního fóra). Takto se chová pokud je parametr *ZobrazovacíMod* nastavený na hodnotu 2 (po stisku tlačítka *Zobrazit vše*). Pokud má tento parametr hodnotu 1 zobrazují se pouze příspěvky, které jsou zaškrtnuté v přehledu příspěvků (po stisku tlačítka *Zobrazit vybrané*).

Druhá funkce *ZobrazPrehledPrispevku* pracuje obdobně jako první funkce s tím, že zobrazuje pouze základní informace o příspěvcích a zobrazuje checkboxy, pomocí kterých je možné nechat si příspěvky zobrazit v plném znění po kliknutí na tlačítko *Zobraz označené*. Zajímavé na této funkci je, že vykresluje u každého příspěvku grafické znázornění jeho vztahu k ostatním příspěvkům. Toho je dosaženo předáváním parametrů *Uroven* a *PoleUrovni*.

Funkce projde parametr *PoleUrovni* od prvního indexu až do indexu menšího než *Uroven* a tam, kde je nastavené vykreslí obrázek vertikální čáry. Pokud *PoleUrovni* nastavené na daném indexu není, zobrazí obrázek průhledný se stejnou velikostí. Pokud je příspěvek na dané úrovni sám nebo je poslední vykreslovaný, zobrazí se obrázek ve tvaru písmene *L*. Jinak zobrazí obrázek ve tvaru písmene *t*.

#### 4.5.6 Propojení s Databází publikací

Jak již bylo řečeno ve specifikaci cíle, systém by měl generovat citační údaje o konferenci ve formátu *BibTeX*. Tyto údaje jsou dostupné po kliknutí na *info* u příslušné konference. Příklad citačních údajů :

```
@inproceedings{
  booktitle = "ICICT 2005: Proceedings of the ITI 3rd International Conference on Information &
  Communications Technology",
  year = {2005},
  month = "March",
  location = "San Jose, California",
}
```

, kde se *booktitle* složí jako “zkratka” “rok” : Proceedings of the “název konference”, *year* a *month* je určen začátkem konference a *location* je složeno z města a státu, kde se konference koná.

Informace o konferenci jsou nahrány z databáze a poté zřetězeny pro vytvoření tohoto řetězce.

#### 4.6 Testování

Otestoval jsem funkčnost celého systému včetně všech jeho nových částí. Provedl jsem také testování správného zobrazení na prohlížečích:

- *Mozilla Firefox 2.0.0.14*
- *Opera 9.22*
- *Konqueror v KDE 3.5*
- *Internet Explorer 7.0*
- *Internet Explorer 6.0*

Dále jsem systém validoval do *HTML 4.01 Strict* pomocí *W3C Validátoru*.

Systém byl nasazen na dvou nezávislých serverech. Oba dva byli veřejně dostupné a systém na nich běžel celou dobu vývoje (asi 8 měsíců).

## 5. BEZPEČNOST

Identifikace uživatele probíhá pomocí relace (session). Session je superglobální proměnná jazyka *PHP*. Obsah této proměnné je zachován při přechodu uživatele mezi stránkami systému. Řeší problém bezstavovosti protokolu *HTTP* a pomocí ní si může systém zapamatovat stav systému pro příslušného uživatele. Vlastní identifikace je prováděna pomocí řetězce tvořeného malými písmeny a číslicemi, který vygeneruje server. Tomuto řetězci se říká *SID* (*session id*) a k uživateli může být dopraven pomocí *cookie* nebo pokud nejsou *cookies* povolena nebo podporována, je dopraven jako součást adresy stránky. Klient se pomocí tohoto řetězce identifikuje při každém svém požadavku. Pokud by útočník chtěl získat přístup do aplikace, musel by odchytil tento řetězec v době, kdy uživatel se systémem pracuje nebo těsně poté. *PHP* interpreter totiž tyto identifikátory po uplynutí určitého času maže. K jejich zneplatnění také může dojít po explicitním vyžádání aplikace, například po odhlášení. Proti tomuto typu útoku je možné systém ochránit zapamatováním si klientovy *IP* adresy a případně dalších podrobností o jeho síťovém připojení. To by řešilo i problém použití dynamického přidělování *IP* adres nebo proxy serveru.

Pro bezpečné přihlašování do systému by bylo potřeba použít protokol *HTTPS*. *HTTPS* je nadstavba protokolu *HTTP*, která poskytuje zvýšenou bezpečnost před odposloucháváním či podvržením dat. Data tímto protokolem jsou přenášena pomocí *HTTP*, ale jsou šifrována pomocí *SSL* nebo *TLS*. [13]

Uživatelská hesla jsou ukládána jako 128 bitový *MD5* otisk. To je zamýšleno jako částečná ochrana proti odcizení tabulky uživatelů *uziv* a také proto, aby správce databáze neviděl přímo nešifrovaná hesla. Jedná se pouze o základní ochranu, protože pro tento typ šifry je možné poměrně snadno vytvořit řetězec, který generuje stejný otisk.

Systém jsem zabezpečil vůči nekorektním vstupům (*SQL Injection*) formulářových dat. Pokud by to nebylo provedeno, mohl by útočník například použitím vhodného řetězce jako hodnoty formulářového prvku nebo proměnné, která je součástí hlavičky, zobrazovat obsah databáze, který má být ukrytý veřejnosti. Zároveň jsem přidal výstupní kontrolu zobrazovaných dat pro eliminaci *HTML* tagů, které by prošly při vkládání do databáze a mohly by způsobit nepříjemnosti při zobrazení stránky.

## 6. INSTALAČNÍ MANUÁL

### Prekvizity

Pro instalaci systému je potřeba mít nainstalovaný webový server *Apache* s podporou *PHP 5.0* a *MySQL*. Dále je potřeba mít nainstalovaný databázový server *MySQL*. Pro zprovoznění automatického posílání emailů je potřeba mít nainstalován plánovač. Například to může být *Cron* popsaný v sekci *Serverové technologie*.

### Postup instalace

1. Všechny *PHP* a *JavaScripty* nahrajte společně s adresáři *css* a *graf* do adresáře ze kterého webový server načítá stránky. Typicky se tento adresář jmenuje *htdocs*.
2. Vytvořte schéma datového úložiště pomocí *DDL* skriptu s kódováním *latin2* (při jiném kódování se nebude správně zobrazovat Čeština). Tento skript je k nalezení v adresáři *sql* pod jménem *dml.sql*. Tato akce se nejjednodušeji provádí ve webovém rozhraní *MySQL*, zvaném *phpMyAdmin*, importem tohoto skriptu. Pokud nemáte webové rozhraní nainstalováno proveďte import přes příkazový řádek.
3. Stejným způsobem nainportujte testovací data, která jsou k nalezení také v adresáři *sql* pod jménem *pokusna\_data.sql*.
4. Nastavte parametry pro připojení skriptů k databázi pomocí funkce *mysql\_connect* na čtvrtém řádku souboru *db.php*.
5. Ujistěte se, že proměnná prostředí *\$\_ENV["debug\_mode"]* je nastavená na *false* a proměnná *\$\_ENV["emailova\_domena"]* je nastavena na doménu, do které budou posílány informativní emaily. Tyto proměnné jsou definovány ve skriptu *fun.php* na řádku 9 a 10.
6. V souboru *mail.php* upravte řádek 11, který definuje příjemce zpráv odeslaných z formuláře *Napište nám*.
7. Soubor *script\_pro\_planovac.php* umístěte do veřejnosti nepřístupné složky a nastavte v něm cestu k souboru *fun.php*, která je k nalezení na třetím řádku. Poté tento soubor naplánujte na každodenní spouštění (postup je popsán v sekci 3.2.3 *Cron*). To lze provést jedním z následujících způsobů:

První způsob spočívá v nakopírování aplikace do jednoho z adresářů */etc/cron.daily*, */etc/cron.hourly*, */etc/cron.weekly* a */etc/cron.monthly*. Již z názvů těchto adresářů je jasné, jak často se bude skript v těchto adresářích spouštět. Pro více specifické časy je potřeba použít utilitu *crontab*.

Po spuštění *crontabu* s parametrem *-e* se zobrazí textový editor se seznamem úloh. Každá úloha má šest parametrů oddělených mezerami nebo tabelátory. Tyto parametry jsou minuta, hodina, den v měsíci, měsíc, den v týdnu (0 - neděle, 1 - pondělí a 6 - sobota) a cesta k programu, který se má spustit. Místo kteréhokoliv z prvních pěti parametrů se může nahradit hvězdičkou, která znamená, že parametr nebyl definován.

Pro spouštění *PHP* skriptu je potřeba zjistit jestli je *PHP* zkompileováno jako *CGI* verze (zda existuje spustitelný program *php*). Pokud ne, musí se skripty spouštět pomocí programu *wget*, který jej zavolá podobně jako webový prohlížeč. Nevýhoda

tohoto způsobu je, že skript zbytečně vytěžuje webový server a tím snižuje i výkon celého serveru. Spuštění a následné provádění *PHP* skriptu z příkazové řádky je mnohem rychlejší.

Pokud tedy na serveru máte nainstalováno *PHP* i jako *CGI* verzi, máte dvě možnosti, jak *PHP* skript spustit. První možnost, která nepotřebuje zásah do skriptu, je zavolání *PHP* interpretu a předání parametru, který bude obsahovat cestu k *PHP* skriptu. To lze provést tímto zápisem (spouštění jednou denně v jednu hodinu ráno):

```
0 1 * * * /bin/php -f /www/server/ script_pro_planovac.php
```

Druhý způsob spočívá v úpravě *PHP* skriptu. Skript se pak stává samostatně spustitelným programem obdobně jako *CGI* verze. Důležité je uvést cestu k interpretu *PHP* do prvního řádku zdrojového kódu a přidělení práv pro spuštění tohoto skriptu.

```
#!/bin/php
```

Tento skript se pak *cronem* naplánuje následujícím způsobem:

```
0 1 * * * /www/server/ script_pro_planovac.php
```

V obou případech je nutné pamatovat, že cesta k vloženému souboru *fun.php* musí být uvedena absolutně. Jinak by mohlo docházet k chybám, protože interpret *PHP* by nenašel požadovaný soubor.[6]

## 7. ZÁVĚR

### 7.1 Výhled do budoucna

Do budoucna by se do systému mohlo přidat zabezpečení při přihlašování pomocí *HTTPS*. Jeho výhody jsou popsány v sekci *Bezpečnost*.

Dále by systém mohl být přepsán použitím nějakého šablonovacího systému například pomocí *Smarty*. Tento systém byl jako jediný přijat do rodiny oficiálních projektů *PHP*. Tímto by bylo dosaženo oddělení výkonné a zobrazovací části a přidávání nových funkcí do systému by bylo snazší a rychlejší.

### 7.2 Shrnutí

Po domluvě se zadavatelem byly konkretizovány nové požadavky na systém, které byly implementovány a byly odstraněny nedostatky původního řešení.

## 8. SEZNAM POUŽITÉ LITERATURY A OSTATNÍCH ZDROJŮ

- [1] PHP - Wikipedie, otevřená encyklopedie [7.6.2008]  
URL: <<http://cs.wikipedia.org/wiki/PHP>>
- [2] SQL - Wikipedie, otevřená encyklopedie [7.6.2008]  
URL: <<http://cs.wikipedia.org/wiki/SQL>>
- [3] JavaScript - Wikipedie, otevřená encyklopedie [7.6.2008]  
URL: <<http://cs.wikipedia.org/wiki/JavaScript>>
- [4] Apache - Wikipedie, otevřená encyklopedie [7.6.2008]  
URL: <<http://cs.wikipedia.org/wiki/Apache>>
- [5] MySQL - Wikipedie, otevřená encyklopedie [7.6.2008]  
URL: <<http://cs.wikipedia.org/wiki/Mysql>>
- [6] Jak na démona Cron - Interval.cz [7.6.2008]  
URL: <<http://interval.cz/clanky/jak-na-demonu-cron/>>
- [7] Relační databáze - Wikipedie, otevřená encyklopedie [7.6.2008]  
URL: <[http://cs.wikipedia.org/wiki/Normální\\_formy#Norm.C3.A1ln.C3.AD\\_formy](http://cs.wikipedia.org/wiki/Normální_formy#Norm.C3.A1ln.C3.AD_formy)>
- [8] Luke Welling, Laura Thomson: PHP a MySQL - rozvoj webových aplikací
- [9] Přednášky předmětu X36DBS [7.6.2008]  
URL: <<http://service.felk.cvut.cz/courses/X36DBS/materialy.html>>
- [10] Přednášky předmětu X36WWW [7.6.2008]  
URL: <<http://amun.felk.cvut.cz/x36www/>>
- [11] Bc. Jan Kemr: Bakalářská práce na téma Databáze konferencí
- [12] Hyper Text Markup Language - Wikipedie, otevřená encyklopedie [8.6.2008]  
URL: <<http://cs.wikipedia.org/wiki/HTML>>
- [13] HTTPS - Wikipedie, otevřená encyklopedie [8.6.2008]  
URL: <<http://cs.wikipedia.org/wiki/https>>

## 9. PŘÍLOHY

### 9.1 DDL skript pro vytvoření datového úložiště

```
-- Zruseni tabulek pokud existují. Tabulky musí být ruseny v tomto pořadí,  
-- protože toto pořadí respektuje vazby mezi tabulkami.  
DROP TABLE IF EXISTS osobkonf;  
DROP TABLE IF EXISTS forum;  
DROP TABLE IF EXISTS konftema;  
DROP TABLE IF EXISTS temauziv;  
DROP TABLE IF EXISTS tema;  
DROP TABLE IF EXISTS konference;  
DROP TABLE IF EXISTS uziv;  
  
-- Vlastní vytvoření tabulek podle ER schématu uložíte.  
CREATE TABLE tema  
(  
    id_tema INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    nazev VARCHAR(80) COLLATE latin2_general_ci DEFAULT 'Nazev tematu',  
    menu_zkr VARCHAR(50) COLLATE latin2_general_ci DEFAULT '',  
    PRIMARY KEY (id_tema)  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_czech_cs;  
  
CREATE TABLE uziv  
(  
    uid INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    uname VARCHAR(32) COLLATE latin2_general_ci UNIQUE NOT NULL DEFAULT '',  
    heslo VARCHAR(32) NOT NULL DEFAULT '',  
    datlog DATETIME DEFAULT NULL,  
    vychozi_predstih INTEGER DEFAULT 14,  
  
    vytvoreni BOOL DEFAULT false,  
    mazaniuprava BOOL DEFAULT false,  
    sprava BOOL DEFAULT false,  
    forum BOOL DEFAULT false,  
  
    PRIMARY KEY (uid)  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_czech_cs;  
  
CREATE TABLE konference  
(  
    id_konf INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    id_rodice INTEGER NULL DEFAULT 0,  
    id_uziv INT UNSIGNED NOT NULL,  
    zkratka VARCHAR(10) COLLATE latin2_general_ci DEFAULT '',  
    rok INT(4) DEFAULT '0',  
    nazev VARCHAR(100) COLLATE latin2_general_ci,  
    homepage VARCHAR(100) DEFAULT '',  
    stat VARCHAR(30) COLLATE latin2_general_ci DEFAULT '',  
    mesto VARCHAR(60) COLLATE latin2_general_ci DEFAULT '',  
    dat_vlozeni TIMESTAMP NOT NULL,  
    dat_zacatek DATE NOT NULL,  
    dat_konec DATE NOT NULL,  
    deadline DATE,  
    notification DATE ,  
    finalversion DATE ,  
    popis TEXT DEFAULT '',  
    archiv TINYINT(1) DEFAULT '0' NOT NULL,
```



```

INDEX idx_konference_id_uziv (id_uziv),
CONSTRAINT konf2uziv FOREIGN KEY (id_uziv) REFERENCES uziv(uid),
PRIMARY KEY (id_konf)
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_czech_cs;

CREATE TABLE konftema
(
id_konf INT UNSIGNED NOT NULL,
id_tema INT UNSIGNED NOT NULL,
INDEX idx_konfskup_id_konf (id_konf),
INDEX idx_konfskup_id_skup (id_tema),
CONSTRAINT konfskup2konf FOREIGN KEY (id_konf) REFERENCES konference(id_konf),
CONSTRAINT konfskup2skup FOREIGN KEY (id_tema) REFERENCES tema(id_tema)
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_czech_cs;

CREATE TABLE osobkonf
(
id_konf INT UNSIGNED NOT NULL,
id_uziv INT UNSIGNED NOT NULL,
posprispevku INT(11) NOT NULL default '0',
priprispevku INT(11) NOT NULL default '0',

forumnotify      BOOL DEFAULT FALSE,
editnotify       BOOL DEFAULT FALSE,

zacateknotify    BOOL DEFAULT FALSE,
zacatek_pouzit_predstih BOOL DEFAULT FALSE,
zacatek_predstih INTEGER DEFAULT 0,
deadlinenotify   BOOL DEFAULT FALSE,
deadline_pouzit_predstih BOOL DEFAULT FALSE,
deadline_predstih INTEGER DEFAULT 0,
finalnotify      BOOL DEFAULT FALSE,
final_pouzit_predstih BOOL DEFAULT FALSE,
final_predstih   INTEGER DEFAULT 0,

INDEX idx_oskonference_id_konf (id_konf),
INDEX idx_oskonference_id_uziv (id_uziv),
CONSTRAINT osobkonf2konf FOREIGN KEY (id_konf) REFERENCES konference(id_konf),
CONSTRAINT osobkonf2uziv FOREIGN KEY (id_uziv) REFERENCES uziv(uid)
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_czech_cs;

CREATE TABLE forum
(
id_item INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
id_reply INT(10) UNSIGNED NOT NULL,
id_konf INT(10) UNSIGNED NOT NULL,
id_uziv INT(10) UNSIGNED NOT NULL,
nazev VARCHAR(50) COLLATE latin2_general_ci,
text TEXT DEFAULT "",
dat_vlozeni DATETIME NULL DEFAULT 0,
dat_editace DATETIME NULL DEFAULT 0,
pocet_editaci INTEGER NULL DEFAULT 0,
CONSTRAINT forum2konf FOREIGN KEY (id_konf) REFERENCES konference(id_konf),
CONSTRAINT forum2uziv FOREIGN KEY (id_uziv) REFERENCES uziv(uid),
PRIMARY KEY (id_item)
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_czech_cs;

CREATE TABLE temauziv
(
id_tema INT UNSIGNED NOT NULL,
uid INT UNSIGNED NOT NULL,

forumnotify      BOOL DEFAULT FALSE,
konfnotify       BOOL DEFAULT FALSE,

```

```
interested BOOL DEFAULT FALSE,  
CONSTRAINT su2skup FOREIGN KEY (id_tema) REFERENCES tema(id_tema),  
CONSTRAINT su2uziv FOREIGN KEY (uid) REFERENCES uziv(uid)  
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_czech_cs ;
```

## **9.2 Obsah přiloženého DVD**

Přiložené *DVD* obsahuje původní systém včetně dokumentace v adresáři *puvodni* a rozšířený systém taktéž včetně dokumentace v adresáři *rozsireny*. Adresář s dokumentací se v obou adresářích jmenuje *dokumentace* a adresář s vlastní realizací se jmenuje *system*.