

**České vysoké učení technické v Praze
Fakulta elektrotechnická**



Bakalářská práce

Databáze konferencí

Jan Kemr

Vedoucí práce: Ing. Petr Fišer

**Studijní program: Informatika a výpočetní technika
leden 2006**

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Petru Fišerovi za jeho odborné rady, všestrannou pomoc a ochotu, s kterou práci vedl.

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) 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 10.ledna 2006

.....

Anotace

Cílem projektu je vytvořit komplexní webovou aplikaci pro evidenci konferencí.

Prvotní myšlenkou se stala potřeba sestavit funkční systém, který by jednotlivé uživatele informoval o dané konferenci a zpřehlednil tak možnost setkávání akademiků mezi sebou.

Systém je zpřístupněn pouze zaměstnancům ČVUT-FEL a na základě tohoto požadavku se do něj lze přihlásit pouze přes vnitřní fakultní databázi těchto osob. Konference obsažené v databázi se dělí do jednotlivých kategorií podle výzkumných skupin Katedry počítačů na ČVUT. Dynamičnost je podpořena přidáváním, editací a dalšími akcemi, které lze s konferencemi provádět.

K vytvoření databáze bylo použito relační databázové aplikace MySQL, nad kterou byla naprogramována aplikace v PHP. Do celého projektu byly dále zakomponovány XHTML, Javascript, CSS a nedílnou součástí je i grafický návrh celkového vzhledu webu.

Abstract

The main aim of the project was to create a web application to be used for an evidence of conferences.

The primary idea was a need to design a system, which would inform single users about existing conferences and thus would enable academics getting together.

The designed system is available for ČVUT-FEL employees only, thus it is allowed to log in to the system using the internal database of these people only.

Conferences included in this database are divided into separate categories by existing research groups in the Department of Computer Science and Engineering. Dynamism is supported by adding, editing and other actions involving conferences.

A relational database application MySQL was used, upon which a PHP application was programmed. XHTML, Javascript, and CSS was included into the whole project as well. An integral part of the web project is the graphics design.

Obsah

1 Rozbor vlastního systému	8
1.1 Úvod	8
1.2 Sestavení požadavků	8
1.3 Stávající systém	9
1.4 Pojetí mého systému	9
1.5 Rozšíření do budoucna	10
2 Popis řešení zadaného úkolu	11
2.1 Datový model	11
2.1.1 E – R model	11
2.1.2 Popis datového modelu	13
2.1.3 Normální formy	15
2.2 Funkční model	15
2.3 Uživatelské role	17
2.4 Technické řešení	17
2.4.1 Datový tok	17
2.4.2 Vybrané PHP funkce a části kódu	19
2.4.3 Javascript funkce	20
2.4.4 CSS	21
3 Srovnání a testování	22
3.1 Porovnání MySQL a PostgreSQL	22
3.1.1 Historie databází	22
3.1.2 Komunikace s PHP	22
3.2 Testování	23
3.2.1 Testování jednotlivých faktorů	23
3.2.2 Testování v prohlížečích	25
4 Budoucí vývoj	26
4.1 Zabezpečení	26
4.1.1 Session	26
4.1.2 Šifrování	28
4.1.3 Vstupy formulářových prvků	28
4.2 Nové moduly	29
5 Uživatelská příručka	30
5.1 Úvod	30
5.2 Přihlášení	30
5.3 Konference	31
5.3.1 Seznam	31
5.3.2 Archiv	32
5.3.3 Informace	32

5.3.4	Editace	33
5.3.5	Forum	34
5.3.6	Vyhledávání	35
5.3.7	Řazení	35
5.4	Přidání nové konference	35
5.5	Osobní konference	36
5.6	Novinky	36
5.7	Doplňkové funkce	37
	Závěr	38
	Použitá literatura a ostatní zdroje	39
	Příloha č. 1: Obsah příloženého cd	40
	Příloha č. 2: Výňatky ze zdrojového kódu	41

Kapitola 1

Rozbor vlastního systému

1.1 Úvod

V současné době nalezneme jen velice málo technických odvětví, které by prodělaly tak intenzivní vývoj jako oblast Internetu. Z původní myšlenky na vytvoření vojenského komunikačního prostředku jako součást sítě ARPANET se postupem času a rozkvětem technologií vyvinulo médium s mnoha využívanými prostředky. Pomineme-li mnoho dalších variant, které nám nabízí, jistě je jeden z hlavních atributů Internetu masová komunikace.

Určitou formou komunikace jsou, podle mého názoru, i různé internetové systémy, kupříkladu webové obchody.

Hlavní výhodou obchodování na Internetu je podle mého to, že se zkracuje cesta od výrobce k zákazníkovi, čímž se šetří peníze a výsledná cena může být nižší. Další neopomenutelná výhoda je v okamžitém vyhledání a porovnání nabídky různých firem, což se vám při nákupu v obchodním domě ne vždy povede. Tato možnost zákonitě musí vyvolat tlak na výrobce, kteří se budou muset snažit stále vylepšovat své služby a svou nabídku. Co víc bychom si my, zákazníci mohli přát. Jaké mají tyto aspekty vliv na koncovou cenu nabízeného zboží je zřejmé a není to obsahem této práce.

Příklad webového obchodu zde uvádím z toho důvodu, že mne inspiroval při výběru tématu mojí bakalářské práce. Čas je v dnešní době často tím nejcennějším faktorem našeho uspěchaného života, kterého se nám příliš nedostává. Každý způsob jeho ušetření je tedy jistě velice zajímavý.

Konferenční systém, postavený na přednostech, které nám Internet nabízí, může jistě přispět mnoha lidem k efektivnějšímu využití jejich času. Systém by jednotlivé uživatele informoval o dané konferenci a zpřehlednil tak možnost setkávání akademiků mezi sebou.

1.2 Sestavení požadavků

Požadavek na vytvoření tohoto systému přišel ze strany Katedry počítačů. Zahrnoval v sobě vytvoření databáze konferencí, možnost jejich archivace, práce uživatele s osobními konferencemi a jejich samotné přidávání. Celý systém měl být dynamický.

Po zamyšlení se nad různými typy řešení, vyvstalo několik základních otázek. Jaké zvolit skriptovací jazyky? Co použít za typ relační databáze? Jak rozmístit jednotlivé prvky v aplikaci tak, aby byla přehledná pro uživatele. A další podobné otázky.

Pro programové řešení jádra aplikace jsem měl možnost zvolit si mezi platformou ASP nebo PHP.

Relačních databází jsem měl na výběr hned několik. PostgreSQL, MySQL, MS SQL server, Oracle, SYBASE
Později jsem svůj výběr zúžil i na základě některých nevyhovujících vlastností, na první dvě jmenované. Tj. PostgreSQL a MySQL.

Jaké konkrétní prostředky jsem použil naleznete v kapitole č.2.

1.3 Stávající systém

Současný systém, po jehož shlednutí jsem usoudil o správnosti načasování k vytvoření nového, funguje na neobstojné bázi editace html kódu.
Požadavek tedy byl, vytvořit systém na bázi databáze.

Pro představu doposud využívané aplikace pro editaci konferencí uvádím adresu.
<http://cs.felk.cvut.cz/~fiserp/>

Kromě výzvy na předělání stávajícího systému, mě na realizaci této práce zaujala celková unikátnost projektu. Když jsem sbíral veškeré potřebné materiály k projektu, překvapilo mě, že jsem nebyl schopen najít profesionální webovou aplikaci s podobným obsahem.

1.4 Pojetí mého systému

Před vlastním návrhem řešení jsem se nejdříve zamyslel nad účelem, pro který je výsledný produkt určen. Je to především úspora času a elegantní řešení pro zpřehlednění stávajícího systému.

Celá aplikace je koncipována a připravena pojmout více evidovaných uživatelů. K detekci úspěšného přihlášení uživatele slouží zobrazení uživatelského jména a po kliknutí na něj se automaticky odhlašuje z aplikace. V rámci zkušební verze, která ještě nebyla propojena se systémem běžícím na fakultě, je odhlašování řešeno pouze nastavením session. (viz 4.1.1) Po brzkém propojení počítám s přesměrováním na logovací stránku, která už je součástí původní fakultní aplikace.

Celý systém jsem se snažil navrhnout, co možná nejdynamičtěji. Hlavní důraz je tu kladen na celkovou přehlednost a intuitivní jednání uživatele vstupujícího poprvé do aplikace.

Samotná aplikace obsahuje seznam výzkumných skupin na Katedře Počítačů, jejichž názvy jsou uchovávány v databázi. Při požadavku přidání/odebrání položky ze seznamu se tato změna musí udělat natvrdo přímo v databázi.

Jednotlivým skupinám přísluší určitý seznam jednotlivých konferencí, kde každá konference obsahuje atributy (Zkratka+rok, Deadline, Notifikace, Finální verze, Termín konání, Místo, Příspěvků). Tento seznam lze řadit (sestupně i vzestupně) podle výše uvedených atributů a také si u něj mohou zvolit, kolik konferencí se jim zobrazí na stránku. Mezi jednotlivými stránkami mohou pochopitelně procházet ve směru i proti.

Dále se lze dostat na podrobnější informace o konferencích, o které se budou zajímat hlavně akademici. Ty shledávám jako jedny z nejpodstatnějších. Každou konferenci lze libovolně editovat a lze na ní i reagovat pomocí fóra.

Pro každého uživatele je v aplikaci připravena část osobní konference, kde vidí informace o své osobě. Zde schraňuje svůj vlastní seznam konferencí, kterým nastavuje prioritu (obyčejná, zajímá mě, zajímá mě - reakce) v editačním menu konference a pak je i podle jednotlivých priorit filtruje.

Nedílnou součástí je v mé práci archiv konferencí. Do něj se konference přesune automaticky, pokud je starší než 31 dnů od aktuálního data nebo jí lze přesunout v editačním menu do archivu manuálně. Pro zpřehlednění archivu se v horní části nalézá tabulka s nabídkou konferencí, které se konaly v minulých letech ve stejný měsíc podle aktuálního data a pod ní umístěny odkazy s jednotlivými roky, v kterých konference probíhaly.

Do databáze lze pochopitelně konferenci i přidat. Docílí toho uživatel završením dvoufázové registrace nové konference. Vstupy formulářových prvků na registrační stránce jsou zabezpečeny javascriptem.

K doplňkovým, ale podstatným funkcím aplikace patří vyhledávání konferencí (požadovanou konferenci naleznete pomocí zadání části jejího názvu nebo zkratky.), napište nám (přípomínka či dotaz adresovaný přes centrální mail) a fórum.

Šikovným prvkem zajišťujícím lepší přehlednost je vypsání novinek. Pět posledních vložených konferencí zobrazených pod sebou.

1.5 Rozšíření do budoucna

Do budoucna (tedy nejlépe v rámci diplomové práce) bych se chtěl zaměřit na několik vylepšení.

Základní myšlenkou je uplatnění aplikace na dalších akademických půdách, popřípadě v soukromém sektoru. Na základě této skutečnosti se jako podstatné rozšíření nabízí správa, rozhraní a zabezpečení uživatele v rámci kompletní aplikace.

Obecně se tomuto problému věnuji již v této práci v kapitole 4.

Dále bych se chtěl věnovat upravení levého menu - skupiny pro požadavky další distribuce aplikace. Skupiny by se mohly nahradit tématy konferencí v rámci konkrétní firmy. Atd.

Pochopitelně, že počítám s rozšířením aplikace o další jazykové verze. Ze začátku o anglickou, popřípadě německou verzi. Dále podle potřeby.

Také by byla zajímavá možnost práce na WAPovém rozhraní a možnost informovat uživatele nejen přes email, ale také pomocí sms.

Kapitola 2

Popis řešení zadaného úkolu

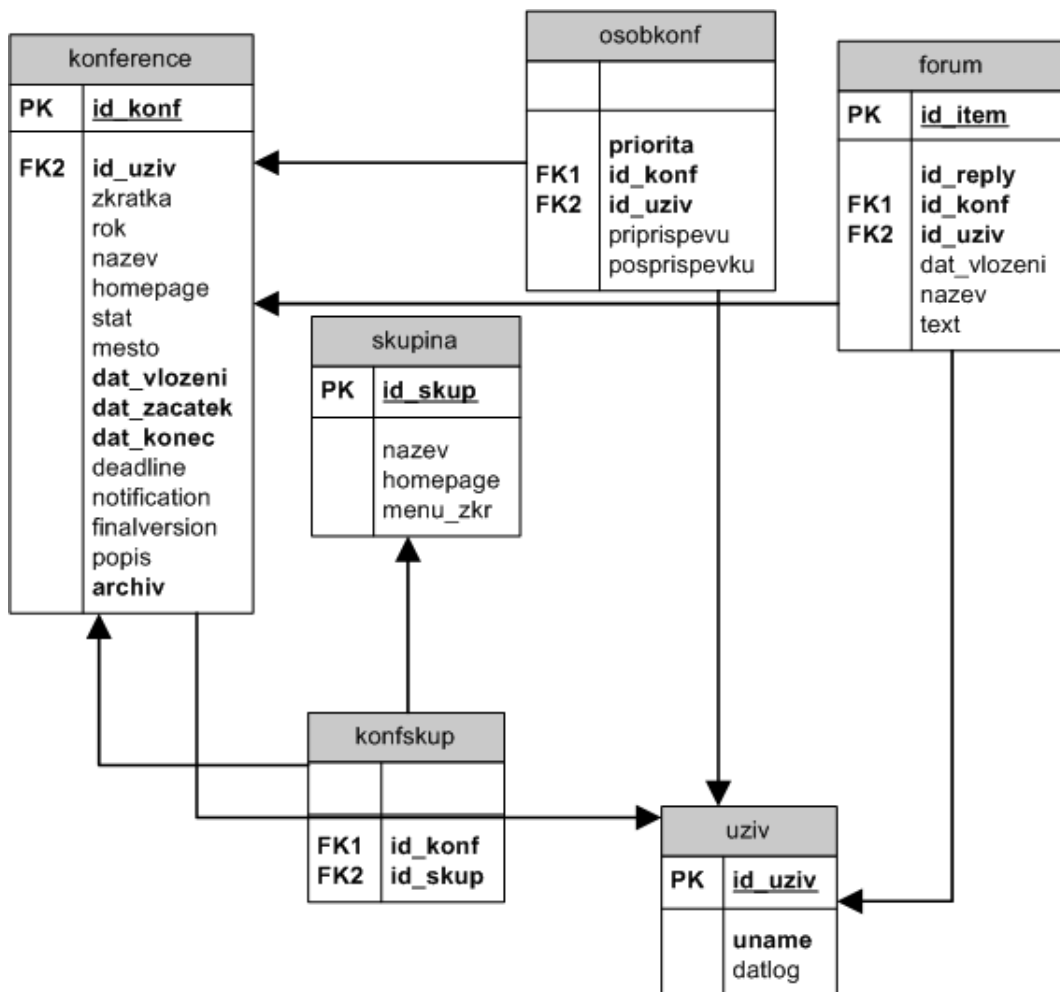
Dané úkoly jsem implementoval v níže uvedených aplikacích a programovacích jazycích. Využil jsem relační databázové aplikace MySQL. Nad vytvořenou databází jsem naprogramoval aplikaci v PHP 5.1, jakožto univerzálním skriptovacím jazyku vhodném pro vývoj webu. Jako jazyka k publikování hypertextu na webu bylo použito XHTML 1.0 Transitional doplněného o CSS. A v neposlední řadě jsem k vytvoření několika funkcí podporujících kompletní funkčnost této práce, použil klientského skriptovacího jazyka Javascript.

2.1 Datový model

2.1.1 E – R model

The Entity-Relationship (ER) model byl původně navržen Peterem v roce 1976 [Chen76] jako cesta k sjednocení sítě a pohledů relační databáze. Jednoduše řečeno, ER model je koncepční datový model, který zobrazuje reálný svět jako entity a vztahy. Základní komponentou modelu je Entity-Relationship diagram, který je použit k vizualizaci reprezentovaných datových objektů. Od doby, co Chen popsal svůj návrh, model byl rozšířen a nyní je společně užíván k designu databází.

Datový model tak, jak byl navržen a použit, je znázorněn na obrázku č. 2.1.



Obr. 2.1 – E-R model

2.1.2 Popis datového modelu

Relace jsou typu 0,1:N, či 1:N. Vazba M:N byla potřeba použít

dvakrát. Jednou mezi tabulkou *konference* a *skupina* (Výzkumná skupina). Konference může patřit do více různých skupin a naopak, jedna skupina samozřejmě obsahuje více různých konferencí. Podruhé mezi tabulkami *konference* a *uziv*(uživatel). Pod každého člověka, který založil záznam o konferenci nebo si ji nastavil jako osobní může spadat více konferencí a opačně, na každou konferenci připadá několik takových osob. V tomto případě se jedná o řešení osobních konferencí pro jednotlivé uživatele. Tento problém dvou vazeb M:N je řešen dekompozicí na nové tabulky *konfskup* a *osobkonf*.

Tabulky také neobsahují žádné redundantní položky. Každá tabulka má svůj vlastní celočíselný identifikátor, opatřený funkcí *autoincrement*.

Rozdíl mezi nenulovým a ostatními atributy znázorňuje v modelu tučně a obyčejné písmo. Tučně – nenulový atribut, normálně – zbylé atributy.

V tabulce *konference* jsou uloženy kompletní informace o jednotlivých konferencích umístěných v systému. *Uziv_id* je cizí klíč z tabulky *uziv* a říká který uživatel vložil konferenci.

V tabulce *uziv* jsou uchovávány obecné údaje, které je třeba registrovat pro každou osobu. V rámci aplikace se přihlašovací jméno(*uname*) uživatele načte z externího textového souboru. Aktuální datum (*datlog*) v sobě uchovává čas posledního přihlášení daného uživatele do systému. Tabulky *uziv* a *konference* jsou provázány dalším vztahem přes tabulku *osobkonf* a její atributy *posprispevku* a při *priprispevku* říkájí kolik uživatel poslal, či přijal jednotlivých příspěvků k daným konferencím. Tabulka *konfskup* provazuje navzájem tabulky *konference* a *skupina*. Posledně zmíněná tabulka obsahuje základní informace o výzkumných skupinách na Katedře počítačů. Tabulka *forum* je umístěna u informací (viz. info) k jednotlivé konferenci a současně obsahuje dva cizí klíče. *Id_konf*, *id_uziv*.

Table *konference*

id_konf {Int} - id daného typu konference (primární klíč)

id_uziv {Int} - id uživatele, který vložil konferenci (cizí klíč z tabulky *uziv*)

zkratka {VarChar} – zkratka příslušné konference

rok {Int} – rok, v kterém konference probíhá

nazev {VarChar} – celý název konference

homepage {VarChar} – domovská stránka

stat {VarChar} – stát, kde se konf. koná

mesto {VarChar} – město, kde se konf. koná

dat_vlozeni {Timestamp} - datum vložení konf.

dat_zacatek {Date} – začátek konf.

dat_konec {Date} - konec konf.

deadline {Date} - doba, do které se musí poslat první verze příspěvku

notification {Date} - doba, do které musí být zasláný článek schválen

finalversion {Date} - doba, do které se musí poslat finální verze článku po korektuře

popis {Text} – všeobecný popis konf.

archiv {TinyInt} – identifikátor, zda-li se konf. nachází v archivu

Table skupina

id_skup {Int} - id typu skupiny (primární klíč)
nazev {VarChar} – název výzkumné skupiny
homepage {VarChar}- domovská stránka
menu_zkr {VarChar} – zkratka pro levé menu v aplikaci

Table uziv

id_uziv {Int}- id uživatele (primární klíč)
uname {VarChar} – přihlašovací jméno uživatele
datlog {Date} – čas posledního přihlášení uživatele

Table forum

id_item {Int}- id uživatele (primární klíč)
id_reply {Int}- id odpovědi
id_konf {Int}- id konference, ke které patří konkrétní fórum (cizí klíč z tabulky konference)
id_uziv {Int}- id uživatele, který vložil reakci (cizí klíč z tabulky uziv)
dat_vlozeni {Date} – datum vložení příspěvku
nazev {VarChar} – název konkrétního příspěvku
text {Text} – vlastní text příspěvku

Table konfskup

id_konf {Int}- cizí klíč z tabulky konference
id_skup {Int} - cizí klíč z tabulky skupina

Table osobkonf

id_konf {Int} - cizí klíč z tabulky konference
id_uziv {Int} - cizí klíč z tabulky uziv
priorita {SmallInt} – priorita u osobní konference
posprispevku {Int} – počet příspěvků zaslaných k dané konferenci
pripispevku {Int}- počet přijatých příspěvků

2.1.3 Normální formy

Normální formy tabulek se používají pro lepší návrhy databázových systémů. Obecně platí, že čím je tabulka ve vyšší normální formě, tím kvalitněji je tabulka navržena. Zde jsou uvedeny jednotlivé formy, jejich význam a použití v rámci mé databáze.

1NF (první normální forma): Tabulka je v první normální formě, jestliže lze do každého pole dosadit pouze jednoduchý datový typ (jsou dále nedělitelné).

V mém případě není splněno, neboť například datový typ date není atomický.

2NF (druhá normální forma): Tabulka je ve druhé normální formě, jestliže je v první a navíc platí, že existuje klíč a všechna neklíčová pole jsou funkcí celého klíče (a tedy ne jen jeho částí).

V mém případě nejsou všechna neklíčová pole funkcí celého klíče. Proto není *2NF* splněna.

3NF (třetí normální forma): Tabulka je ve třetí normální formě, jestliže každý neklíčový atribut není tranzitivně závislý na žádném klíči schématu neboli je-li ve druhé normální formě a zároveň neexistuje jediná závislost neklíčových sloupců tabulky.

V mém případě, pokud by databáze splňovala *1NF*, splňovala by i *3NF*.

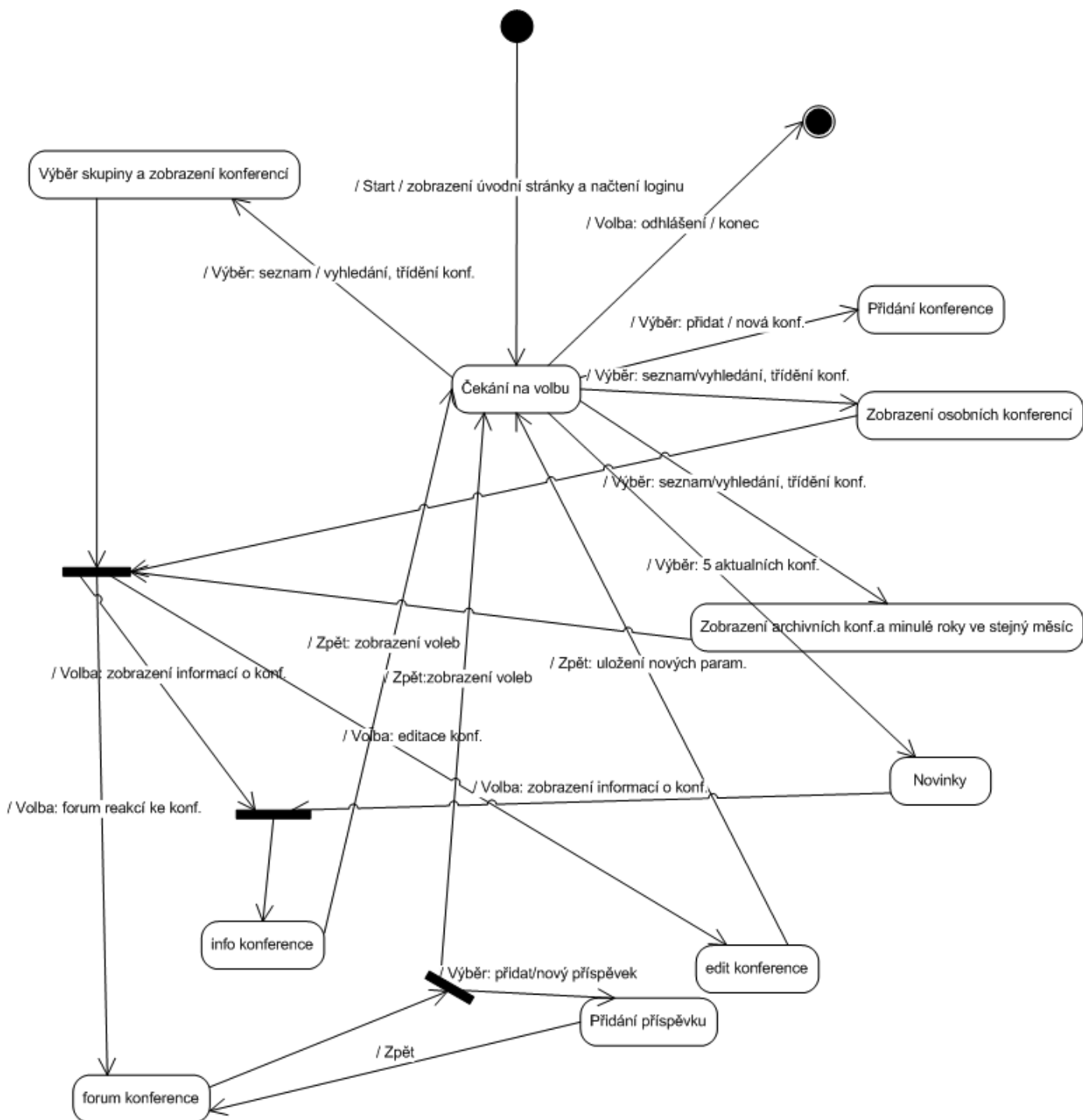
BCNF (Boyce-Coddova normální forma): Tabulka je v Boyce-Coddově normální formě, jestliže pro každou netriviální závislost $X \rightarrow Y$ platí, že X obsahuje klíč schématu R .

V mém případě je toto splněno.

2.2 Funkční model

Funkční diagram popisuje aktivitu uživatele pohybujícího se uvnitř webové aplikace.

Viz obr. 4.3



obr. 2.2 – funkční model

2.3 Uživatelské role

Aplikace se stane používaným nástrojem na Katedře počítačů ČVUT. Slouží pouze k potřebám uvnitř katedry a právě i z tohoto důvodu bylo prvním požadavkem, aby systém využívali pouze zaměstnanci univerzity. Tito uživatelé mají tudíž všichni stejná práva a není potřeba využívat rozdílné uživatelské role pro určitou skupinu.

Závěrem tohoto bloku tedy mohu říci, že požadavek pouze jedné role – uživatel posloužil ve výsledku k větší přehlednosti a stal se i částečným usnadněním vzhledem k celkové rozsáhlosti projektu.

Co se týká budoucí správy aplikace, předpokládám, že bude vytvořena pozice jakéhosi administrátora, který bude celý projekt nadále spravovat, popřípadě rozšiřovat.

2.4 Technické řešení

Projekt má pro všechny své funkce grafické uživatelské prostředí.

Měl by také bez problémů fungovat s kterýmkoli běžným prohlížečem.

Podporovaná rozlišení jsou 800x600 a větší. Použité kódování je iso-8859-2, český jazyk by tedy měl být bezproblémovou záležitostí.

Celá databáze je zpřístupněna pomocí skriptů, které se dají rozdělit do několika základních skupin.

První skupinou jsou skripty, které zobrazují seznamy. Příkladem by mohl být skript *seznam.php*. Jeho základem je stránka zobrazující dynamicky vytvořenou tabulku konferencí.

Další skupinou jsou skripty, které do databáze data přidávají (*konf.php*). Stránky většinou obsahují formulář, jehož vstup je kontrolován javascriptem. Formulář je odeslán dalšímu skriptu, který uloží položky do databáze.

Ve skriptu *user.php* se nastavují veškeré funkce spojené s uživatelem.

Do skupiny takzvaných doplňkových skriptů lze zařadit *forum.php* a *mail.php*, kde uživatel reaguje na jednotlivou konferenci a v druhém případě zasílá osobní dotaz na email.

Pomocné funkce lze nalézt ve skriptu *fun.php*.

Propojení aplikace s databází nalezneme ve skriptu *db.php*.

Javaskriptové funkce nalezneme v souborech *konf.js* a *forum.js*.

2.4.1 Datový tok

Uživatel, jehož přihlášení ověří školní aplikace propojená s databází pracovníků fakulty vstupuje do aplikace Konference.

Jeho uživatelské jméno se ve zkušební verzi načítá z externího textového souboru, kam se po přihlášení zapsalo. V budoucnu se pomocí proměnné (tuto proměnnou obsahuje fakultní skript, který je vložen do skriptu *user.php* aplikace Konference) předá dále. Tuto uživatelské jméno (*uname*) společně s *uid* uživatele a dobou posledního přihlášení (*datlog*) se zapíše do tabulky *uziv*. Nastaví se session

proměnná, v které jsou uloženy *uid* a *uname* a pomocí superglobálního pole `$_SESSION[]` se předávají tyto uživateli informace. V aplikaci se zobrazí uživatelské jméno, po kliknutí na něj se odešle požadavek na odhlášení. Nyní je tato akce řešena natavením *uid* na nulu a *uname* na „nepřihlášen“ pomocí *sessions*. Později bude uživatel přeměrován na logovací stránku, která již nebude součástí aplikace. Při každém novém přihlášení uživatele, který už má vytvořený záznam v tabulce *uziv* se přepíše *datlog*.

Uživatel nyní pohlíží na celou aplikaci, která má rozmístění jednotlivých prvků následující. Na levé straně nahoře nalezneme seznam výzkumných skupin. Položky, které obsahuje, se vyberou z tabulky *skupina*. V pravé části se obsah liší na základě kliknutí na jeden ze čtyř hlavních odkazů – *seznam*, *osobní*, *archiv*, *přidej*.

První tři mají společné prvky, které si popíšeme jako první. Popis *přidej* přijde až později.

Po kliknutí na seznam se z tabulky *konference* vypíše seznam všech konferencí pro všechny skupiny. Jednotlivé konference jsou řazeny pod sebou v tabulce a jejich sloupce obsahují atributy (*id_konf*; *zkratka*; *rok*; *deadline*; *notification*; *finalversion*; *dat_zacatek*; *dat_konec*; *mesto*; *stat*; *priprispevku* - suma tohoto atributu, její alias *prispevku*; *posprispevku* - suma tohoto atributu, její alias *prispevku*; *priorita* – pokud se jedná o osobní konf.)

Stejným způsobem se vypíšou konference, které patří do osobních a do archivu. Zde uvedu způsob, podle kterého rozlišují jednotlivé kategorie, v kterých vypisují určité množství konferencí. Tyto kategorie (např. osobní nebo seznam konferencí pro jednu výzkumnou skupinu) jsou rozlišeny na základě těchto proměnných (*join* – sql: JOIN, *wh* – sql:WHERE, *ob* – sql:ORDER BY, *ofs* – sql:LIMIT, *rows* – sql:LIMIT), které jsou součástí sql dotazu.

Uvedu jeden příklad pro všechny. Chci, aby se mi po kliknutí na jednu z výzkumných konferencí, zobrazil seznam konferencí jí odpovídající. Potom musím do proměnných uložit následující:

```
$join="LEFT JOIN konfskup USING (id_konf)"; - propojení s tabulkou konfskup a rozpoznání, které konference patří do kterých skupin a obráceně.
```

```
$wh="id_skup='$sk'";
```

Do *wh* se ještě přidá druhá část dotazu, omezující zobrazení konferencí, které nejsou starší 31 dnů od aktuálního data a současně mají atribut *archiv* roven nule (když je *archiv* roven jedné, tak je konference v archivu).

Do proměnné *ob* uložíme jméno atributu, podle kterého chceme řadit konference.

Teď se blíže podíváme na jednotlivou konferenci, co všechno s ní můžeme provádět za akce. V první řadě tu máme možnost vložení nové konference. V první fázi vyplníme základní informace o konferenci a v druhé popis se zaškrtnutím výzkumných skupin, do kterých konference patří. Tyto data z formulářových prvků se uloží do tabulek *konference* a *konfskup*. Nakonec je zde možnost přidání konference do osobních s vybranou prioritou. Po vybrání priority se vytvoří záznam v tabulce *osobkonf*, kde se navíc atributu *priorita* přiřadí nenulová hodnota.

Vloženou konferenci máme možnost editovat. Můžeme jí přidat do osobních s odpovídající prioritou (viz. výše uvedený popis) nebo naopak vyřadit z osobních.

Smažeme záznam z tabulky *osobkonf* podle *id_uziv* uživatele a *id_konf*. Uživatel nastavuje počet příspěvku, které poslal a přijal. Pokud zadává uživatel poprvé tento počet, automaticky se konference zařadí do osobních (do tabulky *osobkonf*). Po další editaci těchto atributů (*posprispevku*, *priprispevku*) se pouze přepíše jejich hodnota.

V rámci editačního menu lze konferenci ještě přesunout do archivu (atribut *archiv* se v tab. *konference* nastaví na jedna) a editovat její ostatní parametry, které se přepíšou v tabulce *konference* na aktuální.

Když zobrazím informace o konferenci, z tabulky *konference* se vypíšou všechny atributy konference kromě archivu. Podle *id_uziv* se dozvíme jméno uživatele, který konferenci vložil (*id_uziv* je provázané s atributem *uid* v tabulce *uziv*). Informace uzavírá pořadí výzkumných skupin, do kterých konference patří. Tyto údaje získáme z tabulky *konfskup*.

Reagovat na konferenci lze pomocí fóra. Nový příspěvek se vloží do tabulky *forum*. Pokud chceme příspěvek editovat, přepíšeme jeho parametry včetně aktuálního času vložení. Příspěvky zobrazíme jejich vypsáním z tabulky *forum*.

Dostáváme se pomocným prvkům v archivu. Dotaz na zobrazení konferencí z minulých let ve stejný měsíc vypíše z tabulky *konference* seznam konferencí, jejichž rok je menší než aktuální a zároveň měsíc je roven aktuálnímu měsíci. Rozdělení konferencí do jednotlivých roků zajišťuje dotaz, který vybere roky z tabulky *konference* a vypíše je seřazené vzestupně.

Prvek zajišťující lepší přehlednost je vypsání novinek. Pět posledních vložených konferencí zobrazených pod sebou se vypíše z tabulky *konference*.

2.4.2 Vybrané PHP funkce a části kódu

PHP je dnes velmi rozšířená technologie umožňující snadné programování na straně serveru (server-side programming). Toho lze využít k tvorbě různých interaktivních webových stránek.

Počátky vývoje PHP sahají do roku 1994, kdy Rasmus Lerdorf vytvořil jednoduchý systém evidence přístupů ke svému webu, nejdříve v jazyce Perl, poté v C. Ten se rozšířil mezi další uživatele, kteří přicházeli s požadavky na vylepšení. Vznikl tak systém Personal Home Page Tools, později Personal Home Page Construction Kit.

Vlastní práce byla psána v PHP 5.1

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

Na začátku každého skriptu se testuje, zda jsou skripty volané z *index.php* (centrální skript, kde je nastavená proměnná *indexphp* na *true*)

```
$pri=((isset($_GET["pri"]) && is_numeric($_GET["pri"]) &&
$_GET["pri"]>=0 && $_GET["pri"]<=3) ? $_GET["pri"] : 0);
```

Přiřazení vlastností do proměnné *pri* (priorita), která je používána v dalších částech skriptu.

Funkce `debug($msg)`

zapiše do souboru *debug.txt* datum, jméno prohlížeče, a zprávu, která se definuje v následující funkci při zavolání funkce `debug ($msg)`.

Funkce `sqlchyba ($msg, $que)`

zavolá funkci `debug ($msg)` a pro parametr `msg` vypíše SQL chybu:(tu nastaví funkce `mysql_error()`) a pro parametr `que` vypíše SQL dotaz: do *debug.txt*. Vypíše tuto hlášku - Při zpracování Vašeho požadavku došlo bohužel k chybě. Vraťte se prosím o krok zpět a zkuste to znovu.

Funkce `connect ()`

zajistí připojení k MySQL databázi.

Funkce `query ($query_str)`

zavolá funkci `mysql_query ($query_str, $dbConn)`, která provede dotaz na aktuálním spojení v aktivní databázi na serveru a vrátí identifikátor výsledku. V případě neúspěchu funkce `query ($query_str)` zavolá funkci `sqlchyba` a nakonec zařídí odpojení od aktivní databáze.

Funkce `query_id ($query_str)`

je stejná jako `query ($query_str)`, ale navíc obsahuje proměnnou `id`, v které je uloženo id poslední přidané položky

Funkce `query_ar ($query_str)`

je stejná jako `query ($query_str)`, ale navíc obsahuje proměnnou `ar`, v které je uložen počet upravených řádků.

2.4.3 Javascript funkce

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk z dílny Netscape. Není užitečný jako samostatný jazyk, ale je navržen pro snadné vložení do jiných produktů a aplikací, např. webových prohlížečů.

Pro svou popularitu se stal základem pro evropský standard ECMAScript. Jeho napodobeninou je i JScript v MSIE od Microsoftu.

Funkce `check_date (d,m,y)`

vrací hodnotu `true`, pokud jsou splněny podmínky počtu dnů pro jednotlivé měsíce. Funkce počítá i s přestupným rokem. Parametry `d`, `m`, `y` - den, měsíc a rok.

Funkce `check_add1 ()`

vytvoří dialogové okno `alert`. Ošetřuje vstupní formulářová pole. Pokud je formulářový prvek prázdný, uloží varovnou hlášku do proměnné `m`. Takto otestuje všechny vstupy, hlášky přičítá do proměnné `m`. Nakonec zavolá funkci `alert` s parametrem `m`.

Funkce `select_all (x)`

zaškrtně všechny zaškrťovací formulářové prvky ve `form_add2`.

Funkce `invert_all (x)`

invertuje výběr zaškrtnutých zaškrťovacích formulářových prvků ve form_add2.

Funkce `forum_reply(id,nazev)`

nastaví název do formuláře v reakci na příspěvek ve fóru podle daného id příspěvku.

2.4.4 CSS

CSS (Cascading Sytle Sheets) neboli kaskádové styly vznikly jako souhrn metod pro úpravu vzhledu stránek. První návrh normy byl zveřejněn v roce 1994, v roce 1996 byla pak vydána specifikace CSS 1, v roce 1998 CSS 2, nyní se pracuje na verzi CSS 3.

CSS se využívá k formátování obsahu HTML, XHTML a XML dokumentů. Ve srovnání s formátováním pomocí atributů v HTML formátovací schopnosti rozšiřuje. Styly umožňují přesně určit, jak bude který element vypadat. Narozdíl od atributů stylem můžeme definovat jednotný vzhled elementu pro celý dokument (např. že všechny nadpisy úrovně 1 budou červené) a to jediným zápisem pro příslušný element (nikoli v každém tagu příslušného elementu). Stejně tak můžeme pomocí stylu určit odlišné formátování pro třeba jen jediný výskyt určitého elementu. Tím se jednak zbavíme velkého množství kódu, jednak se tento kód stane mnohem přehlednější. Navíc pokud se jednou rozhodneme změnit například barvu písma všech odstavců, bude to pro nás otázka několika málo vteřin, měnit každý atribut u každého elementu v HTML by byla katastrofa. Jeden styl můžeme snadno použít pro libovolné množství stránek.

V mém případě využívám dvou css souborů. *Style.css a print.css*, kde první obsahuje veškeré informace o vzhledu aplikace a druhý z nich zajišťuje podobu stránky, tak jak bude vypadat pro tisk. Oba soubory jsou nalinkovány do souboru *index.php* těmito html tagy.

```
<link rel="stylesheet" type="text/css" media="screen"
href="css/style.css">
<link rel="stylesheet" type="text/css" media="print"
href="css/print.css">
```

Dále uvedu několik příkladů pravidel pro jednotlivé elementy, které mají být formátovány.

```
body {
position:relative; min-width:780px; width:100%;
border:0; margin:0; padding:0; font:70%/140% verdana, sans-serif;
color:black; background:#BFC0C3 url(../graf/body.gif) repeat-x
left top;
}
```

Selektor `body`(vlastní tělo dokumentu) obsahuje výše uvedenou deklaraci.

```
#minwidth {
width:770px; height:1px; overflow:hidden;
}
```

Identifikátor `minwidth` určuje minimální šířku stránky. Stránka je tímto koncipována na minimální rozlišení 800x600.

Kapitola 3

Srovnání a testování

V této kapitole se hodlám zaměřit na základní rozdíly mezi relačními databázemi MySQL a PostgreSQL fungujícími na bázi server-klient. Toto téma jsem zvolil na základě původního požadavku, aby samotná aplikace běžela pod databází Postgres. Během práce na ní jsem usoudil použít MySQL hlavně z důvodu placeného web hostingu, na kterém běží testovací verze a kde není podporován Postgres.

Neměl by chybět ani test aplikace na různých platformách a kompatibilita s různými prohlížeči.

3.1 Porovnání MySQL a PostgreSQL

Otázka srovnání MySQL a PostgreSQL 8.1. souvisí s jejich rychlostí. Třebaže nyní Postgres je na tom s rychlostí přibližně stejně, dřívější verze byly známy jako pomalé. V následujícím srovnání se pokusím vystihnout jiné rysy než rychlost, která je v tomto případě z obecného pohledu zavádějící.

3.1.1 Historie databází

PostgreSQL relační databázový systém (RDBMS) vzešel z POSTGRES projektu na Univerzitě v Berkeley. Profesor Michael Stonebraker odstartoval projekt v roce 1986 kvůli transformaci původního projektu Ingres.

Snahou projektu Postgres bylo vytvořit relačně-objektovou DB. Objektově orientovaná DB umožňuje uživatelům definici vlastních metod, pomocí kterých se přistupuje k datům a manipuluje s nimi.

Po ukončení projektu v letech 1994-1995 se dva studenti zmíněné univerzity Jolly Chen a Andrew Yu rozhodli dále pracovat na akademickém kódu Postgresu a přejmenovali původní Postgres na Postgres95. V létě roku 1996 přechází Postgres95 do open source podoby a je přejmenován na PostgreSQL. Tento je vyvíjen dodnes. Osobně nevím o OpenSource produktu, který by byl tak přímo a výrazně svázán s historií nějakého počítačového oboru a tak moc ho ve své historii ovlivnil.

Prioritou pro vytvoření MySQL se stala pro lidi, co ho psali, používat mSQL ke spojení k jejich vlastním low-level datovým strukturám. Objevili, že mSQL postrádal některé vlastnosti a rychlost, kterou vyžadovali a rozhodli se napsat vlastní. Takto začal život MySQL jako produktu.

3.1.2 Komunikace s PHP

Zde vyberu několik základních funkcí, které používá PHP 5.0 pro komunikaci s oběma databázemi.

Funkce `mysql_connect([string hostname [:port] [:/path/to/socket] [,string username [, string password]])`
otevře spojení s MySQL Serverem.

Funkce `mysql_create_db(string database name [, resource link_identifier])`
vytvoří MySQL databázi.

Funkce `mysql_db_query(string database, string query [, resource link_identifier])`
vybere databázi a provede na ní dotaz.

Funkce `mysql_error([resource link_identifier])`
vrátí chybový text z poslední MySQL funkce nebo prázdný znak, když se chyba nevykytla.

Funkce `mysql_fetch_array(resource result [, int result_type])`
načte výsledné řádky jako asociativní pole, numerické pole nebo obojí.

Funkce `mysql_num_rows(resource result)`
vrací číslo řádků.

Pro PostgreSQL existují identické funkce.

Jsou to `pg_connect()`, `pg_errormessage()`, `pg_fetch_array()`, `pg_numrows()` a další, které zde pro velké množství neuvádím.

Aktuální verze PHP 5.0 a 5.1.2 se mimo jiné liší přidáním nových funkcí pro v3 protokol (začal se využívat v PostgreSQL 7.4.). Například funkce `pg_query_params()` - provedení parametrizovaných dotazů

3.2 Testování

Testování je velice důležitou součástí vývoje a pokud jej zanedbáme, můžeme splakat nad výdělkem.

Testována byla demo verze běžící na <http://konference.pixart.cz>.

3.2.1 Testování jednotlivých faktorů

Faktory se navzájem odlišují ve způsobu testování a nástrojích k tomu určených. Obecně se dá říci, že k testování většiny z nich lze použít neplacené online utility a analýzy, případně příslušný freeware. Například testování použitelnosti je však záležitostí mnohem náročnější.

Funkčnost odkazů

Ověření, že odkazy v aplikaci fungují. Využil jsem službu W3C – Link Checker. Tester v tomto případě neohlásil žádnou chybu.(Valid links!)

Přístupnost

Tato aplikace by měla splňovat různá kritéria přístupnosti. Nabízejí se následující standardy a pravidla: WCAG 1.0 (Web Content Accessibility Guidelines 1.0), Section 508 (americký zákon o přístupnosti webových stránek financovaných z

veřejných zdrojů), Manifest Dogma W4 (Worth World Wide Web) a Blind Friendly Web.

Základní a velice důležitou vlastností všech přístupných webů, kterou by však měly plnit všechny internetové stránky, je dostatečný kontrast mezi barvami textu a pozadí. K otestování této vlastnosti jsem použil nástroj Kontrast barev na Sově v síti (<http://sovavsiti.cz>). Colorblind Web Page Filter mi umožnil zobrazit stránky tak, jak je vidí barvoslepi uživatelé s různými druhy poruch. Tento test prošel bez problémů.

Datová velikost stránek a jednotlivých objektů a rychlost načítání

K otestování této vlastnosti jsem využil jeden z nejlepších nástrojů Web Page Analyzer (<http://www.websiteoptimization.com/>).

Předkládám některé výsledky:

Pro rychlost připojení 56K, 128K, 1.44Mbps se stránky stahují 23, 4.39, 0.56 sekund. Největší velikost 22.6KB má objekt – graf/logo.jpg. Většina analýzy proběhla v rámci norem. Překročena byla pouze celková velikost a množství obrázků, velikost css souboru.

Použitelnost

Testování použitelnosti je věc mimořádně složitá, touto problematikou se zabývají celé firmy. V mém případě se tento faktor naplno projeví, až po zavedení demo verze do praxe.

Optimalizace pro vyhledávače

Optimalizace pro vyhledávače (SEO - *Search Engine Optimization*) je samostatným a dosti složitým oborem. Proto jsem testoval jen některé faktory. Beru také v potaz, že aplikace bude přístupná jen na půdě školy, ale v rámci budoucího rozšíření zde tyto faktory uvádím.

Hustota klíčových slov - je přesně měřitelnou veličinou a traduje se, že její hodnoty pro nejvýznamnější slova by se měly pohybovat v rozmezí mezi 3 až 7 procenty. V mém případě mi nástroj Keyword density analyzer (<http://www.ranks.nl/tools/spider.html>) z totálních 312 slov zjistil, že nejvíce používané slovo je *2006* s hustotou 10.6%. Můj požadavek na klíčové slovo *konference* byl vyhodnocen s hustotou výskytu 1.28%.

Stránky pohledem robota - testování metainformací a prohlížení stránek v takové podobě, v jaké je pravděpodobně vidí vyhledávací roboti, kteří na ně přicházejí. Stránky z pohledu robota mi ukázal nástroj Poodle Predictor (<http://www.gritechnologies.com/tools/spider.go>) a Meta Tag Analyzer (<http://www.widexl.com/remote/search-engines/metatag-analyzer.html>) provedl vyhodnocení metaelementů a klíčových slov na stránkách. Tag *title* v tomto nástroji vůbec neuspěl. Hodnotil jeho významnost vůči zbylému obsahu stránek.

Pozice ve vyhledávačích – testování proběhlo za pomoci nástroje Search Engine Position Checker (<http://www.webmaster-toolkit.com/search-engine-position-checker.shtml>). Při zadaném klíčovém slovu *konference* se aplikace logicky nenachází na některé z padesáti nejvyšších příček nejpoužívanějších vyhledávačů.

3.2.2 Testování v prohlížečích

V této části uvádím seznam jednotlivých prohlížečů, v kterých byla aplikace testována a pro které byla odladěna.

Firefox 1.0.6, 1.0.7

Konqueror v KDE 3.3, 3.4

Konqueror v Gnome 3.4

Opera 7.54, 8.02

Mozilla 1.7.11

Internet Explorer 6.0

Test aplikace proběhl i na známém textovém prohlížeči Lynx v prostředí Linuxu. Testován byl vzhled a funkčnost stránek. Protože Lynx podporuje kaskádové styly byly všechny funkce konference plně přístupné a funkční. Tento prohlížeč byl zvolen spíše pro zajímavost. V dnešní době vyspělého hardwaru se s textovým prohlížečem v běžných případech nesetkáme.

Kapitola 4

Budoucí vývoj

Jednou z deviz mého projektu by do budoucna mělo být rozšíření funkcí pro uživatele. Už nyní jsem mojí práci částečně rozšířil vzhledem k výchozímu zadání o budoucí požadavek přístupu více uživatelů do aplikace a jejich evidenci. V databázi jsem vytvořil tabulku *uziv*, která v sobě prozatím ukládá základní informace spojené s uživatelem.

Možnost registrace nového uživatele, celkové zlepšení zabezpečení aplikace pomocí relací (sessions), vlastní kódování hesla uživatele a další rozšíření aplikace se stanou náplní další verze této práce.

Dalším zlepšením, které také rozeberu v této kapitole, bude lepší kontrola a ošetření vstupních formulářových prvků.

4.1 Zabezpečení

4.1.1 Session

Vzhledem k tomu, že informace v systému obsažené jsou velice osobní, je jistě jejich ochrana důležitou budoucí součástí projektu.

Z řešení, která se v dnešní době běžně používají a zajišťují poměrně vysoký stupeň bezpečí, jsem vybral relace (sessions).

Session proměnná je speciální proměnná v PHP, jejíž obsah je uchováván i při přechodu uživatele mezi jednotlivými stránkami. Řeší problém bezstavovosti protokolu HTTP, udržíme pomocí ní informace o stavu aplikace a práci uživatele s ní. Session například používají téměř všechny e-mailové servery. Jejich použití je tedy všude tam, kde je potřeba monitorovat, či ověřovat uživatele, kupříkladu s využitím jeho IP adresy. Nutno podotknout, že mechanismus session od prvních přídatných doplňků pro PHP3 doznal řady změn a od PHP verze 4.1+ je práce se sessions standardně dobře funkční a použitelná.

Prvním krokem pro použití session je inicializace pomocí funkce `session_start()`. Tato funkce nejdříve zkontroluje, zda bylo aplikaci předáno jméno nějaké již probíhající session. Pokud ne, interpret PHP zahájí novou relaci: vytvoří nový identifikátor relace. V opačném případě je znovu spuštěna aktuální session, která byla předána. Každý uživatel, který vstoupí na stránku, tedy obdrží unikátní číslo. Nazveme ho ID, resp. SID (session ID). Často to nebývá pouze číslo, ale vyskytují se v něm i nečíselné znaky, např. `a5cda6d5587acdd643ab59bb`. Tímto číslem se bude uživatel identifikovat, a proto musíme ve všech odkazech předávat tento parametr. PHP implicitně předává SID v proměnné `PHPSESSID`. K nastavení slouží sekce *Session* a položka *session.name* v souboru *php.ini*. Ukládání čísla relace v URL je pouze jeden ze způsobů jejího předávání. PHP nabízí také ukládání potřebných dat do cookie. Všechna data jsou uložena v superglobálním poli `$_SESSION[]`. Protože `session_start()` ve

většině případů vytváří cookie, měla by být spouštěna ještě předtím, než jsou klientovi odeslána jakákoliv data.

Máme tedy spuštěnu session a k dispozici pole `$_SESSION[]`. Zde se již naplno projeví výhody nových verzí PHP a superglobálních polí. Pokud chceme přidat do session nějaká data, uděláme to zcela jednoduše a přirozeně tak, že je přidáme do pole `$_SESSION[]`. Chceme-li například předávat mezi aplikacemi proměnnou `user` s hodnotou `10`:
`$_SESSION["user"]=10`. Zcela ve stejném duchu se nese modifikace dat v session. Opět stačí změnit hodnotu daného prvku pole. Pokud chceme některá data ze session vymazat, stačí zrušit příslušný řádek. Následující kód umožní vyjmutí proměnné `user` ze session: `unset($_SESSION["user"]);`.

Nyní již umíme spustit session a modifikovat její obsah. To nejdůležitější a také nejnáročnější (alespoň zdánlivě) nás však stále čeká. Jak se v předchozím textu zmiňuji, musíme nějakým způsobem předávat id naší session. Nejlepší případ nastane, pokud můžeme použít cookie. Ne všichni uživatelé ale mají cookies povoleny, proto by aplikace měla být schopna předávat id session i pomocí URL. Zatímco ve starších verzích se o vše musel starat programátor sám, nyní se o vše postará PHP.

Životnost každé relace končí, pokud uživatel opustí korektně stránku (například se odhlásí), nebo vyprší-li její časový limit. Je zřejmé, že na serveru běží démon, který kontroluje aktivní relace a maže již neaktivní. Neaktivní relace se nazývají garbage (smetí).

Nastavení časového limitu, způsobu vytvoření, zápisu, čtení a mazání relace se provádí pomocí této funkce:

```
session_set_save_handler (open, close, read, write, destroy, gc)
```

Procedury `open`, `close`, `read`, `write`, `destroy` a `gc` (garbage collection) jsou definované funkce pro práci s relacemi. Jak bylo řečeno, relace se inicializuje funkcí `session_start()`. Ukončení relace se provádí funkcí

```
session_destroy(). Aby mohla být session úspěšně smazána, je podstatné ji
```

```
nejdříve spustit pomocí session_start(). Funkce session_destroy()
```

```
nesmaže položky pole $_SESSION[], o to se programátor musí postarat sám,
```

```
například pomocí $_SESSION=array(); nebo unset($_SESSION["user"]); .
```

Měli bychom se vyhnout smyčkám. Mohlo by dojít k zahlcení serveru. Toho by se také dalo docílit, pokud by někdo nějakým robotem neustále vytvářel nové relace.

Možnou ochranou je v tomto případě použití databáze a kontrola IP adresy.

Pokud se v průběhu práce s aplikací změní IP adresa klienta, odepřeme mu k aplikaci přístup. To může způsobit určité obtíže uživatelům, kterým se skutečně během práce změnila IP. Při zahájení session do ní uložíme také IP adresu uživatele a tu kontrolujeme při každém požadavku klienta. Problémem mohou být uživatelé připojení přes proxy, nebo skrytí ve vnitřní síti (maškaráda). I tomu lze čelit tak, že do proměnné, kterou si uložíme do session, ale i do té, s kterou již následně porovnáváme, vložíme vše, co lze o IP adrese klienta zjistit. Tak lze rozlišit i více uživatelů přistupujících k aplikaci přes jednu proxy.

```
$IPadresa=$REMOTE_ADDR;  
$IPadresa.="@".$HTTP_X_FORWARDED_FOR;  
$IPadresa.="@".$HTTP_FORWARDED;  
$IPadresa.="@".$HTTP_CLIENT_IP;  
$IPadresa.="@".$X_HTTP_FORWARDED_FOR;
```

Pro opravdu bezpečnou aplikaci je třeba použít, kromě všech předchozích pravidel, i zabezpečený protokol HTTPS. Tím se stane aplikace odolnou i proti různým pokusům se zmíněným nastavením proxy, se změnou konfigurace webového serveru, i změnou DNS. Celkově proti odposlechu komunikace, ne už však proti napadení serveru. K čemu by však byla naše session, když už se útočník dostal přímo ke skriptům na serveru.

V mém případě jsou sessions využity a pomocí superglobálního pole

```
$_SESSION[] se předávají uživateli informace uid, uname.
```

V rámci zkušební verze, která ještě nebyla propojena se systémem běžícím na fakultě, je odhlašování řešeno:

```
$_SESSION["uid"]=$uid=0;  
$_SESSION["uname"]=$uname="nepřihlášen";
```

Pro budoucí rozšíření bych rád použil rozšířeného zabezpečení popsaného v předešlém textu. Tj. kontrola IP adresy a HTTPS.

4.1.2 Šifrování

V rámci zabezpečení bude také použita funkce MD5, která spočítá hodnotu hashovací funkce pro zadaný text.

Algoritmus MD5 vygeneruje pro libovolný řetězec 128-bitové číslo, které je s velmi vysokou pravděpodobností pro každý řetězec jedinečné. Algoritmus se používá zejména pro tvorbu digitálních podpisů apod.

V aplikaci Konference bude použit pro kódování přenášených přihlašovacích jmen a hesel.

4.1.3 Vstupy formulářových prvků

Jako velmi důležitá složka bezpečnosti je ošetření vstupů formulářových prvků.

Na tento problém se pochopitelně pamatoval, kde pokud uživatel vkládá novou konferenci, musí zadat potřebné parametry do formulářových vstupů. Pro zabezpečení těchto vstupů využívám javascriptových funkcí, které jsou popsány podrobně ve druhé kapitole. Problém zde nastává, pokud má uživatel ve svém prohlížeči vypnutý javascript.

V rámci tzv. user friendly aplikace se pro uživatele stává daleko příjemnější, pokud pouze odklikne varovné javascriptové okno upozorňující na chybějící parametry v registraci konference. Avšak jak bylo zmíněno, neřeší to vypnutý javascript v prohlížeči. Tento problém lze jednoduše odstranit přidáním potřebných funkcí v PHP.

Proto mi do budoucna vyvstává řešení, které spojuje tyto dva požadavky – user friendly a bezpečnost. Navrhuji, že vytvořím navíc funkce pro vstupy i v PHP. Na počátku stránky index.php bude javascriptová funkce, která pokud je js zapnutý,

nastaví proměnnou, s kterou pracuje i funkce napsaná v PHP. Tuto proměnnou použije PHP v další funkci, která rozliší jestli zavolat potřebné funkce v PHP pro formulářové vstupy nebo nikoli.

4.2 Nové moduly

Veškeré nové prvky, které může aplikace obsahovat, se odvíjejí od konkrétních potřeb daného subjektu, pro který může být dále využitelná.

Pro požadavek registrace nového uživatele zde bude vytvořena registrační stránka. Jejím obsahem jsou základní informace o uživateli – jméno, příjmení, adresa, telefon, atd.

V rámci potřeby přidání nové položky do levého (výzkumné skupiny), zde bude vytvořena stránka, pomocí které položku přidáme.

V archivu konferencí se nyní objevuje položka – konference v minulých letech ve stejný měsíc podle aktuálního data. Zde jsou vypsány konference na základě sql dotazu, který obsahuje: vypiš všechny konference, které začínají stejným měsícem jako je ten aktuální a současně mají menší rok než ten aktuální. Tuto zpřehledňující položku v budoucnu upravím tak, aby vypisovala konference za minulé roky s tolerancí +-14dní od aktuálního data. Nyní totiž může nastat případ, že budu mít konferenci 30.-31.1. a pak se to posune na 1.2., nevidím to.

Kapitola 5

Uživatelská příručka

Obsah:

- Úvod
- Přihlášení
- Konference
 - Seznam
 - Archiv
 - Informace
 - Editace
 - Forum
 - Vyhledávání
 - Řazení
- Přidání nové konference
- Osobní konference
- Novinky
- Doplňkové funkce

5.1 Úvod

Právě se Vám dostává do rukou uživatelská příručka k webové aplikaci. Aplikace slouží k evidenci konferencí na Katedře počítačů. K jejich přidávání, archivaci, vyhledávání. V této příručce se jen zběžně seznámíte s aplikací, tak abyste byli schopni základních úkonů.

5.2 Přihlášení

Přihlášení se do webové aplikace je velmi snadné. Stačí pouze vyplnit váš login a heslo a následně potvrdit, buď klávesou enter či tlačítkem přihlásit se. Pamatujte, že přihlásit se můžete pouze, pokud jste jako uživatel veden v centrální databázi zaměstnanců Katedry počítačů.

Po přihlášení se dostáváte na úvodní stránku aplikace. Jako další indikátor, že se jedná opravdu o přihlášeného uživatele, slouží zobrazení jeho loginu v horní pravé části aplikace – “Uživatel - login“



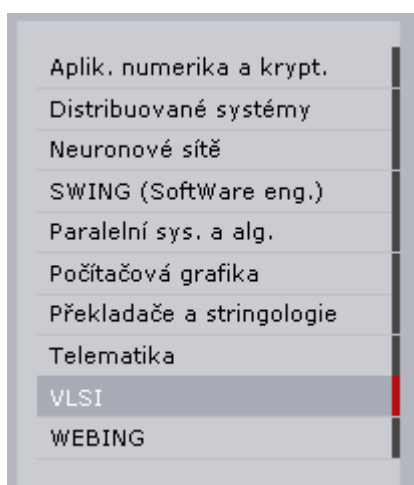
obr 5.2.1 - přihlášený uživatel

5.3 Konference

5.3.1 Seznam

Po přihlášení do databázové aplikace Konference se vám automaticky načte seznam všech evidovaných konferencí na Katedře počítačů.

Na základě vámi zvolené výzkumné skupiny lze výběr zúžit o seznam konferencí pro zadanou kategorii. Seznam výzkumných skupin je umístěn v levé horní části webové aplikace a obsahuje 10 položek (Aplikovaná numerika a kryptologie, Distribuované systémy, Neuronové sítě, SWING, Paralelní systémy a algoritmy, Počítačová grafika, Překladače a stringologie, Telematika, VLSI, WEBING).



obr 5.3.1 - seznam výzkumných skupin

V souhrnném seznamu v tabulkovém zobrazení jednotlivé konference obsahují tyto nejdůležitější atributy (podrobnější popis jejich významu – viz 2.1.2) - Zkratka+rok, Deadline, Notifikace, Finální verze, Termín konání, Místo, Příspěvků. V pravé části každého řádku tabulky naleznete tři odkazy. První odkaz *info* vás naviguje na podrobné informace o jednotlivé konferenci (viz. 5.3.3). Po kliknutí na druhý odkaz *edit* lze editovat vámi vybranou konferenci (viz. 5.3.4) a díky poslednímu odkazu se prokliknete na forum konference (viz. 5.3.5). Po kliknutí na odkaz konkrétní zkratky a roku se dostáváte přímo na domovskou stránku konference.

Výše jmenované atributy lze podle potřeby řadit. (viz. 5.3.7)

Zkratka+rok	Deadline	Notifikace	Finální verze	Termín konání	Místo	Příspěvků	
DIAGO@ 2006	30.05.2005	01.01.2006	06.01.2006	31.01.2006 - 01.02.2006	Kouty nad Desnou, ČR	0	info edit forum
IWTPV 2006	17.04.2006	18.04.2006	19.04.2006	22.04.2006 - 23.04.2006	Prague, ČR	0	info edit forum
Poster 2006	09.04.2006	15.04.2006	19.04.2006	18.05.2006 - 18.05.2006	Praha, ČR	0	info edit forum

[zpět](#) [nahoru](#)

obr 5.3.2 - seznam konferencí

5.3.2 Archiv

Po kliknutí na ikonu archiv se automaticky dostáváte do archivu konferencí. Uspořádání jednotlivých konferencí je totožné se seznamem (viz. 5.3.1). Opět zde nalezneme tabulkové zobrazení jednotlivých konferencí řazených pod sebou s příslušnými atributy, které také lze řadit (viz. 5.3.7).

Pro zpřehlednění archivu se v horní části nalézá tabulka s nabídkou konferencí, které se konaly v minulých letech ve stejný měsíc podle aktuálního data. Pokud konference tuto vlastnost splňují, tabulka se zobrazí a obsahuje tyto atributy - Zkratka+rok a Termín konání. V případě, že vlastnost nesplňuje ani jedna, tabulka není zobrazena.

Dalším přehledným prvkem je umístění odkazů s jednotlivými roky, v kterých konference probíhaly. Po kliknutí na rok se vám zobrazí seznam pouze těch údajů, které v onen rok probíhaly.

Zkratka+rok	Termín konání	
ABCD_2003	12.01.2003 - 09.01.2003	obnovit
fd_2005	01.01.2005 - 01.12.2005	obnovit
fa_2006	01.01.2004 - 01.01.2004	obnovit

Zkratka+rok	Deadline	Notifikace	Finální verze	Termín konání	Místo
ABCD_2003	21.10.2005	26.10.2001	15.11.2005	12.01.2003 - 09.01.2003	Praha

obr 5.3.3 – archiv konferencí

5.3.3 Informace

Jako uživatel zjistíte veškeré informace o vámi zvolené konferenci kliknutím na odkaz info, který naleznete v pravé části každého řádku tabulky.

Informace o konferenci POSTER	
id	336
vložil	kemrli
zkratka	Poster
ročník	2006
název	POSTER
homepage konference	http://radio.feld.cvut.cz/conf/poster2006/
místo konání	Praha, ČR
datum vložení	14.01.2006
termín konání	18.05.2006 - 18.05.2006
deadline	09.04.2006
notifikace	15.04.2006
finalní verze	19.04.2006
popis	
Mezinárodní studentská konference POSTER 2006	
konference patří do skupin	
Neuronové sítě	http://ncg.felk.cvut.cz
SWING (SoftWare engineering.)	https://swing.felk.cvut.cz/cms/
VLSI	http://service.felk.cvut.cz/vlsi
zpět na seznam	

obr 5.3.4 – informace o konferenci

5.3.4 Editace

Odkaz *edit* vás přesune do editační stránky pro jednotlivou konferenci. Tu lze rozdělit na dvě základní části dle voleb. Ve spodní části se nám nabídnou předvyplněné informace na vstupu formulářových prvků. Vstupy jsou totožné jako při registraci nové konference. (viz 5.4). V horní části jsou čtyři základní prvky, které uživateli umožní manipulaci s konferencí. Lze jí přidat do osobních nebo naopak vyřadit z osobních. Dále je zde možnost přesunutí do archivu. Při přidání do osobních vybíráte zároveň i konkrétní prioritu. Pokud chcete přesouvat do archivu, nutnou podmínkou je potvrzení vašeho úmyslu zaškrtnutím formulářového prvku zaškrťovací políčko. V neposlední řadě lze zadat osobní počet odeslaných a přijatých konferencí. Po přidání jakéhokoli počtu přijatých/odeslaných se konference automaticky zařadí do osobních. V seznamu se tyto počty reakcí od všech uživatelů sčítají. Tyto základní čtyři prvky v horní části jsou rozšířeny o prvek *obnovit pro tento rok*. Ten se zobrazí pokud má konference datum začátku jiný rok než rok aktuálního data.

1/2 Základní údaje

Osobní seznam
 Přidat do osobních | priorita: obyčejná

Vyřadit z osobních

Vaše příspěvky do konference
 přijato: 1 | posláno: 2 | Změnit

Archiv
 ← potvrzení úmyslu | Přesunout do archivu

Editace konference

Název: POSTER

Zkratka: Poster

Rok: 2006

Město: Praha | Stát: ČR

Termín konání: Od: 18 05 2006
 Do: 18 05 2006

Deadline: 09 04 2006

Notifikace: 15 04 2006

Finální verze: 19 04 2006

Homepage: http://radio.feld.cvut.cz/conf/poster2006/

ODKURČOVAT >>

obr 5.3.5 – editace konference

5.3.5 Fórum

Kliknete-li na odkaz *fórum*, dostáváte se na stránku fóra konference. Lze přidávat jednotlivé příspěvky, které se řadí pod sebe. Konkrétní příspěvek lze opět editovat nebo na něj reagovat pomocí odkazu *replay*. Daná odpověď se zařadí pod onen příspěvek.

POSTER - forum

Přidat příspěvek

Název:

ODESLAT

:: <start :: < předchozí :: 1 :: > další :: > konec ::

15.01.2006 kemrj1 doporučení

[replay](#) | [edit](#)

Zajímavá konf. minulý ročník byl přínosem

obr 5.3.6 – editace konference

5.3.6 Vyhledávání

V horní části aplikace naleznete formulářové textové pole. Vámi požadovanou konferenci naleznete pomocí zadání části jejího názvu nebo zkratky.

Například chcete vyhledat konferenci *Poster*

Klepněte do pole *Vyhledávání*

1. Zadejte klíčové slovo *Poster*
2. Stiskem klávesy Enter zahájíte vyhledávání.

Výsledek hledání se vám zobrazí ve středu aplikace.



obr 5.3.7 – vyhledávání konference

5.3.7 Řazení

V celé aplikaci lze provádět dva typy řazení v závislosti na attributech. První typ třídění se nalézá v souhrnném seznamu všech konferencí, popřípadě jednotlivých skupin a druhý typ lze nalézt v osobních konferencích uživatele.

V seznamu konferencí dosáhneme seřazení jednotlivých položek tak, že klikneme na jméno atributu (Zkratka+rok, Deadline, Notifikace, Finální verze, Termín konání, Místo, Příspěvků) v horní části tabulky. Druhý způsob jak tyto atributy seřadit, je použití formulářového prvku umístěného nad samotným seznamem konferencí. V rozevíracím poli stačí vybrat vámi požadovaný atribut, dle kterého se chystáte řadit.

U osobních konferencí může každý uživatel svým vlastním konferencím přiřazovat prioritu (všechny, obyčejná, zajímá mě, zajímá mě - reakce). Ve svém seznamu pak může ponechat pouze ty konference, které odpovídají zadané prioritě. Opět využíváte rozevíracího pole formuláře.

5.4 Přidání nové konference

Konferenci přidáváte pomocí vstupů do formulářových prvků a celý proces je rozdělen na dvě fáze.

První část registrace nové konference spočívá v zadání následujících povinných údajů – *Název, Zkratka, Rok, Město, Termín konání(od, do), Deadline, Notification, Final version, Homepage*.

Po stisknutí tlačítka pokračovat se dostáváte do fáze, v které zadáváte následující vstupy – *Popis konference, Výběr skupiny, do které konference patří*. Tento výběr je rozšířen o tlačítka *vybrat všechny, invertovat výběr*. Zakončení celého procesu registrace je volba možnosti přidání nové konference do osobních s odpovídající prioritou.

Vypsání všech zadaných parametrů je identifikátorem úspěšného dokončení registrace.

5.5 Osobní konference

Jednotlivou konferenci přidám do osobních dvěma způsoby. Během vytváření nové konference(5.4) mohu rovnou vybrat, zda zahrnu novou konferenci do osobního seznamu. Druhou možnost naleznete na editační stránce konference (5.3.4), kde po stisknutí tlačítka tuto možnost vykonáte.

V osobních konferencích přihlášený uživatel vidí jako první svoje uživatelské jméno, pod kterým se nachází tři základní informace spadající pod uživatele. *Poslaných příspěvků, přijatých příspěvků a datum posledního přihlášení.* Počet poslaných a přijatých příspěvků se uživateli automaticky počítají při jakémkoliv jejich potvrzení v editaci konference. (5.3.4)

Pod těmito vlastnostmi naleznete řazení (5.3.7), které obsahuje i řazení dle priority, které můžete přidělovat svým vlastním konferencím.

Jednotlivé priority:

obyčejná: zelená barva

zajímá mě: modrá barva

zajímá mě – reakce: červená barva - na tuto konferenci už jsem nějakým způsobem reagoval.



The screenshot shows a user profile for 'kemrj1' with the following statistics:

- Poslaných příspěvků: 2
- Přijatých příspěvků: 1
- Poslední přihlášení: 2006-01-21 12:22:27

Navigation controls include: Str. 1 z 1, start, předchozí, 1, další, konec, Položek: 5, OK.

Sorting options: Řazení: Zkratka+rok, Priorita: Všechny.

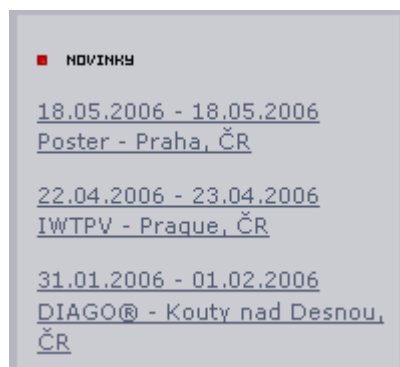
Zkratka+rok	Deadline	Notifikace	Finální verze	Termín konání	Místo
Poster 2006	09.04.2006	15.04.2006	19.04.2006	18.05.2006 - 18.05.2006	Praha,

Legend: Vysvětlivky: ■ Obyčejná konference ■ Zajímá mě - konference ■ Zajímá mě reakce - konference

Obr 5.5 – osobní konference

5.6 Novinky

V levé spodní části uživatel nalezne novinky. Seznam pěti naposledy uživateli vložených konferencí. Pokud na jednotlivou položku kliknete, dostanete se na podrobné informace o konferenci.



Obr 5.6 – novinky

5.7 Doplnkové funkce

Pokud má uživatel jakékoli připomínky či dotazy, může je adresovat přes centrální mail, kde mu na ně osoba k tomu určená odpoví. Tento formulář naleznete v zápatí stránek pod odkazem *Napište nám*.

Dalšími užitečnými pomůckami pro uživatele se mohou stát dva odkazy v pravé horní části stránek – *tisk* a *mapa stránek*.

A screenshot of a contact form titled 'Napište nám'. Below the title is a note: '*Pokud chcete dostat odpověď, zadejte prosím Váš email.' The form contains two input fields: 'Jméno:' and 'Email:'. Below these is a large text area labeled 'Text zprávy'. At the bottom of the form is a red button with the text 'ODESLAT DOTAZ'.

Obr 5.7 – napište nám

Závěr

Po rozhovoru s pedagogem působícím na ČVUT-FEL Katedře počítačů a shlednutí stávajícího systému, jsem sestavil soubor požadavků na program typu Konferenční systém. Produkt je navržen tak, aby byla databáze zpracována pokud možno optimálně. Celá aplikace je naprogramována ve skriptovacím jazyce PHP. Použity jsou také prvky kaskádových stylů a Javascriptu. Ověření práv uživatele není součástí aplikace. K přihlašování a následnému ověřování uživatele se využívá interní databáze katedry. Po úspěšném přihlášení se login uživatele zapíše do externího souboru, z kterého ho pak moje aplikace načte a dále s ním pracuje. O správu se stará uživatel s pravomocemi administrátora. Celá aplikace byla podrobně otestována na různých platformách v rámci různých webových prohlížečů.

Výsledkem je systém, který je jistě již nyní schopen obstát a fungovat na půdě Katedry Počítačů. Tento provoz samozřejmě může poukázat na některé detaily, které bude třeba případně vyřešit jiným, pro praxi vhodnějším způsobem. Dle mého názoru by se mělo jednat pouze o detaily.

Pro jeho dynamičnost není pochyby, že při minimálních změnách a rozšíření, lze tento systém aplikovat i jinde. Po rozšíření, které jsem popsal v předcházejícím textu, lze tento systém finančně zhodnotit a poskytnout tak tuto aplikaci i jiným institucím. Samozřejmě, že jedna z prvotních myšlenek byla distribuce této aplikace i na jinou akademickou půdu než ČVUT.

Použitá literatura a ostatní zdroje

- [1] Jaroslav Pokorný – Dotazovací jazyky,
Science 1994
- [2] Přednášky DBS [3.9.2005]
URL: <<https://service.felk.cvut.cz/courses/X36DBS/prednaska>>
- [3] Přednášky TI [3.9.2005]
URL: <<https://service.felk.cvut.cz/courses/36TI/TIN-Pred-PPT.html>>
- [4] Pokorný, J., Halaška, I.: Databázové systémy,
Praha: Vydavatelství ČVUT, 2002
- [5] Eric Meyer o CSS : ovládněte kaskádové styly! / Eric Meyer. -- Vyd. 1. --
Brno : Zoner Press, 2004
- [6] CSS kaskádové styly : kompletní průvodce / Petr Staníček. -- Vyd. 1. --
Praha : Computer Press, 2003
- [7] Eric Meyer o CSS : pokračujeme s kaskádovými styly profesionálně! /
Eric Meyer. -- Vyd. 1. -- Brno : Zoner Press, 2005
- [8] Článek o session [21.11.2002]
URL: <<http://interval.cz/>>
- [9] Testování webových stránek [31.03.2004]
URL: <<http://interval.cz/>>
- [10] Programovací a značkovací jazyky pro web – přehled [2004]
URL: <<http://www.webtvorba.cz/>>
- [11] PHP [2006]
URL: <<http://www.php.net/>>
- [12] Historie relačních databází [19.10.2001]
URL: <<http://http://www.root.cz/clanky/historie-relacnich-databazi/>>

Příloha č. 1

Obsah přiloženého CD

Na CD jsou kompletně všechny skripty realizující systém.

Program Easyphp, který vytvoří web server localhost, na němž se dá aplikace spustit. Také je zde databáze demo. K nahlédnutí je i databázový model a funkční diagram.

Přiložena je i elektronická podoba Bakalářské práce.

Příloha č. 2

Výňatky ze zdrojového kódu

create.sql: (vytvoření databáze)

```
CREATE TABLE skupina (  
    id_skup INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    nazev VARCHAR(80) DEFAULT 'Nazev skupiny',  
    homepage VARCHAR(80) DEFAULT 'Neexistuje domovska stranka',  
    menu_zkr VARCHAR(50) DEFAULT '',  
    PRIMARY KEY (id_skup)  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_general_ci;  
  
CREATE TABLE uziv (  
    uid INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    uname VARCHAR(32) NOT NULL DEFAULT '',  
    datlog DATETIME DEFAULT NULL,  
    PRIMARY KEY (uid)  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_general_ci;  
  
CREATE TABLE konference (  
    id_konf INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    id_uziv INT UNSIGNED NOT NULL,  
    zkratka VARCHAR(10) DEFAULT '',  
    rok INT(4) DEFAULT '',  
    nazev VARCHAR(100),  
    homepage VARCHAR(100) DEFAULT '',  
    stat VARCHAR(30) DEFAULT '',  
    mesto VARCHAR(60) 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_general_ci;  
  
CREATE TABLE konfskup (  
    id_konf INT UNSIGNED NOT NULL,  
    id_skup INT UNSIGNED NOT NULL,  
    INDEX idx_konfskup_id_konf (id_konf),  
    INDEX idx_konfskup_id_skup (id_skup),  
    CONSTRAINT konfskup2konf FOREIGN KEY (id_konf) REFERENCES  
    konference(id_konf),  
    CONSTRAINT konfskup2skup FOREIGN KEY (id_skup) REFERENCES  
    skupina(id_skup)  
  
    ) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_general_ci;
```

```

CREATE TABLE osobkonf (
    prioritita SMALLINT NOT NULL,
    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',
    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_general_ci;

```

```

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,
    dat_vlozeni DATETIME,
    nazev VARCHAR(50) COLLATE latin2_general_ci,
    text TEXT DEFAULT '',
    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_general_ci;

```

[index.php](#): (úvodní stránka)

```

<?php //ISO 8859-2 = +ěščřžýáíéúů
session_start();
$indexphp=true;
require("fun.php");
require("uziv.php");

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-2">
<meta http-equiv="content-language" content="cs">
<meta name="author" content="Design/Code: Jan Kemr[Pixart -
www.pixart.cz]; e-mail: info@pixart.cz; Bakalářská práce 2005">
<meta name="copyright" content="(c) 2005, ČVUT, FEL, Výpočetní technika;
service.felk.cvut.cz;">
<meta name="description" content="konference, vlastní konference, archiv
konferencí, výzkumné skupiny">
<meta name="keywords" content="CS.FELK.CVUT.CZ | Konference, Katedra
výpočetní techniky a programování">
<meta name="robots" content="all, follow">
<title>Konference CVUT|Katedra Pocitacu</title>

<link rel="shortcut icon" href="http://konference.pixart.cz/favicon.ico"
type="image/x-icon">
<link rel="stylesheet" type="text/css" media="screen"
href="css/style.css">
<link rel="stylesheet" type="text/css" media="print"

```

```

        href="css/print.css">
<script language="JavaScript">
<!--
function SymError() {
    return true;
}

window.onerror = SymError;

var SymRealWinOpen = window.open;

function SymWinOpen(url, name, attributes) {
    return (new Object());
}

window.open = SymWinOpen;

//-->
</script>

</head>
<body>
<div id="minwidth"><!-- workaround (min-width) -->
</div><!-- /MIN-WIDTH -->
<div id="shopbar">
    <p><a href="http://konference.pixart.cz/" name="top">
    </a>
    <span id="accesslink" class="noprint"> | <a
    href="http://konference.pixart.cz">přeskočit na navigaci
    </a></span></p>
</div><!-- konec shopbar -->
<form id="toolbar" class="noprint" action="index.php?seznam="
method="POST">
    <p>

        <div id="grafmenu">
        <ul>
        <li id="list"><a href="index.php?seznam=" title="seznam
        konferencí"><i>Seznam</i></a></li>
        <li id="personal"><a href="index.php?seznam=&per=1" title="osobní
        konference"><i>Osobní</i></a></li>
        <li id="add"><a href="index.php?add1=" title="přidej
        konferenci"><i>Přidej</i></a></li>
        <li id="archiv"><a href="index.php?seznam=&arc=1" title="archiv
        konferencí"><i>Archiv</i></a></li>
        <li id="search">
        <input id="searchtype" name="search" value="" type="text">
        <input src="graf/lupa.gif" alt="vyhledat v názvech a zkratkách"
        type="image">
        </li>
        </ul>
        </div><!-- konec grafmenu-->

        <span style="margin: 0.3em; position: absolute; left:
        550px;">vyhledat</span>
        <span style="margin: 0.3em; position: absolute; right: 90px;"><a
        href="index.php?logout=">Uživatel: <?echo($uname);?></a></span>
        <span style="margin: 0.3em; position: absolute; right: 60px;">
        <input type="image" src="graf/printer.gif" width="20" height="17"
        onClick="window.print()"></span>
        <span style="margin: 0.3em; position: absolute; right: 30px;"><a
        href=""></a></span>

    </p>
</form><!-- konec toolbar -->

<div id="main">

```

```

<div id="content"><!--vrstva pro obsah stredu-->
  <div id="content-in"><!--misto pro nabidku zbozi-->
    <div id="content-in2">
      <br>
      <?
        if(isset($_GET["info"]) ||
          isset($_GET["add1"]) ||
          isset($_GET["add2"]) ||
          isset($_GET["add3"]) ||
          isset($_GET["edit1"]) ||
          isset($_GET["edit2"]) ||
          isset($_GET["edit3"]) ||
          isset($_GET["pers"]) ||
          isset($_GET["arch"]) ||
          isset($_GET["pris"]) ||
          isset($_GET["renew"]))
            ) require("konf.php");

        else if(isset($_GET["forum"])) require("forum.php");
        else if(isset($_GET["mail"])) require("mail.php");
        else require("seznam.php");
      ?>
      <br class="clear">
      <p class="clear"><br><br><br><br><br><br><br><br><br>
      </div><!-- konec content-in2 -->
    </div><!-- konec content-in -->

</div><!-- konec content -->
<div id="sidebar" class="noprint">
  <div id="sidebar-in">
    <div id="sidebar-in2">
      <div id="skupiny">
        <?
          if(mysql_num_rows($dbQuery=query("SELECT id_skup,menu_zkr FROM
            skupina"))>0)
            {
              echo("<ul>\n");
              while($dbRS=mysql_fetch_array($dbQuery))
                {
                  echo("<li><a href='index.php?seznam=&sk=$dbRS[0] '$dbRS[1]</a></li>\n");
                }
              echo("</ul>");
            }
          else echo("žádné skupiny nejsou k dispozici");
        ?>
      </div><!-- konec skupiny-->
      <br><br><br><br><br><br><br><br>
    </div><!-- konec sidebar-in2-->
    
  <!-- novinky... pet nejaktualnejsich-->
  <div id="sidebar-in2">
    <div id="news">
      
    <? /***NOVINKY***/
      if(mysql_num_rows($dbQuery=query("SELECT
        id_konf,dat_zacatek,dat_konec,zkratka,mesto,stat FROM konference
        ORDER BY dat_zacatek DESC LIMIT 5"))>0)

```

```

        {
            while($dbRS=mysql_fetch_array($dbQuery))
                echo("<p class='itemn'><a
                    href='index.php?info=&id=$dbRS[0] '>".xsqldate($dbRS[1])." -
                    ".xsqldate($dbRS[2])."<br> $dbRS[3] - $dbRS[4], $dbRS[5]</a></p>\n");
        }
    ?>

</div><!-- konec news-->
<br><br><br><br><br><br><br>
</div><!-- konec sidebar-in2-->


</div><!-- konec sidebar-in -->
</div><!-- konec sidebar -->
</div><!-- konec main -->

<div id="znak" class="noprint">


</div><!-- konec znak -->

<div id="footer">
<address class="clear">[Aktualizace: 1.&nbsp;&nbsp;&nbsp;2.&nbsp;&nbsp;&nbsp;2006]<span
class="noprint"> -

<a href="index.php?mail=">napište nám</a> - </span>Copyright &nbsp;&nbsp;&nbsp;2005
<a href="http://feld.cvut.cz">ČVUT - FEL</a>, Katedra počítačů; <span
class="noprint"><a href="http://www.pixart.cz">code&design -
pixart</a></span></address>

</div><!-- konec FOOTER -->

</body></html>

```

seznam.php: (seznam konferencí)

```

<?php //ISO 8859-2 = +ěščřžýáíéúů

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

?>
<?

/*archiv*/
$arc=(isset($_GET["arc"]) && is_numeric($_GET["arc"]) ? $_GET["arc"] :
0);

/*osobni pohled*/
$per=(isset($_GET["per"]) && $_GET["per"]==1);

/*priorita*/
$pri=((isset($_GET["pri"]) && is_numeric($_GET["pri"]) && $_GET["pri"]>=0
&& $_GET["pri"]<=3) ? $_GET["pri"] : 0);

$pri=((isset($_POST["pri"]) && is_numeric($_POST["pri"]) &&
$_POST["pri"]>=0 && $_POST["pri"]<=3) ? $_POST["pri"] : $pri);

/*vyhledavani*/
$search=(isset($_GET["search"]) ? rawurldecode($_GET["search"]) : "");
$search=(isset($_POST["search"]) ? $_POST["search"] : $search);

/*skupina*/
$sk=0;

if(isset($_GET["sk"]) && is_numeric($_GET["sk"]) && $_GET["sk"]>0)

```

```

{
$sk=$_GET["sk"];
$wh="id_skup='$sk'";
$join="LEFT JOIN konfskup USING (id_konf)";
if(mysql_num_rows($dbQuery=query("SELECT nazev FROM skupina WHERE $wh
LIMIT 1"))>0)
{
$dbRS=mysql_fetch_array($dbQuery);
$title=$dbRS[0];
$wh.=" AND konference.archiv='0' AND TO_DAYS(NOW())-
TO_DAYS(dat_zacatek)<=31";
}
else $title="žádná skupina";
}
else if($per)
{
$title=$uname;
$wh="osobkonf.id_uziv='$uid'";
if($pri) $wh.=" AND priorita='$pri'";
$join="LEFT JOIN osobkonf USING (id_konf)";
}
else if($arc)
{
$title="archiv konferencí";
$wh="(konference.archiv='1' OR TO_DAYS(NOW())-
TO_DAYS(dat_zacatek)>31)";
if($arc>2000) $wh.=" AND YEAR(dat_zacatek)='$arc'";
$join="";
}
else if($search!="")
{
$title="vyhledávání: $search";
$wh="nazev LIKE '%$search%' OR zkratka LIKE '%$search%'";
$join="";
}
else
{
$title="všechny skupiny";
$join="";
$wh="konference.archiv='0' AND TO_DAYS(NOW())-
TO_DAYS(dat_zacatek)<=31";
}
}

//trideni
$ob_pole=array("rok,zkratka","deadline","notification","finalversion",
"dat_zacatek","stat,mesto","prispevku");
$sort=((isset($_GET["sort"]) && is_numeric($_GET["sort"]) &&
abs($_GET["sort"])>0 && abs($_GET["sort"])<=7) ? $_GET["sort"] : 1);
$sort=((isset($_POST["sort"]) && is_numeric($_POST["sort"]) &&
abs($_POST["sort"])>0 && abs($_POST["sort"])<=7) ? $_POST["sort"] :
$sort);

if(isset($_POST["desc_x"])) $sort=-abs($sort);
$ob=$ob_pole[abs($sort)-1].($sort<0 ? " DESC" : "");

//strankoveni
$ofs=(isset($_GET["ofs"]) && is_numeric($_GET["ofs"]) ? $_GET["ofs"] :
0);
$rows=(isset($_GET["rows"]) && is_numeric($_GET["rows"]) ? $_GET["rows"]
: 5);
$rows=(isset($_POST["rows"]) && is_numeric($_POST["rows"]) ?
$_POST["rows"] : $rows);
if(mysql_num_rows($dbQuery=query("SELECT count(1) FROM konference $join
WHERE $wh"))>0)
{
$dbRS=mysql_fetch_array($dbQuery);

```

```

        $pg=ceil($dbRS[0]/$rows);
        $ap=ceil($ofs/$rows);
    }
?>

<h3 class="clear" id="zacatek"><? echo($title); ?></h3>

<?
if($per) { /**jen pro OSOBNI***/
    $datlog="N/A";
    if(mysql_num_rows($dbQuery=query("SELECT
sum(priprispevku),sum(posprispevku) FROM osobkonf WHERE
id_uziv=$uid"))>0)
    {
        $dbRS=mysql_fetch_array($dbQuery);
        $priprispevku=($dbRS[0]=="" ? 0 : $dbRS[0]);
        $posprispevku=($dbRS[1]=="" ? 0 : $dbRS[1]);
    }
    if(mysql_num_rows($dbQuery=query("SELECT datlog FROM uziv WHERE
uid=$uid"))>0)
    {
        $dbRS=mysql_fetch_array($dbQuery);
        $datlog=$dbRS[0];
    }
}
?>

<p class="osuup"></p>
<p class="osuc">Poslaných příspěvků:
<? echo($posprispevku); ?>
<br>Přijatých příspěvků: <? echo($priprispevku); ?>
<br>Poslední přihlášení:<br><? echo($datlog); ?></p>
<p class="osudown"></p>
<?
    } ?>

<p class="clear"></p>
<form name="rows_form" id="sort_form" action="<?
echo("index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$ofs&sort=$
sort"); ?>" method="post">
<span style="font-size:9px;"><strong>&nbsp;&nbsp;  Str.
<? echo($ap+1); ?> z <? echo($pg); ?></strong></span>
<span style="font-size:9px;">

    <? echo(" :: "); //oddelovac
        //prvni
        echo($ap>0 ? "<a
href='index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=0&rows=$row
s&sort=$sort'><img src=\"graf/arrows/sipl_.gif\" alt=\"&lt;\&gt;start</a>\"
: "<b><img src=\"graf/arrows/sipl.gif\" alt=\"&lt;\&gt;start</b>");

        echo(" &nbsp;&nbsp;&nbsp; "); //oddelovac

        //predchozi
        echo($ap>0 ? "<a
href='index.php?seznam=&arc=$arc&per=$per&pri=$pri&&sk=$sk&ofs=".($ofs-
$rows)."&rows=$rows&sort=$sort'><img src=\"graf/arrows/sipll_.gif\"
alt=\"&lt;&lt;\&gt;&nbsp;&nbsp;&nbsp;předchozí</a>\" : "<b><img
src=\"graf/arrows/sipll.gif\" alt=\"&lt;&lt;\&gt;&nbsp;&nbsp;&nbsp;předchozí</b>");

        echo(" :: "); //oddelovac

        //cisla stranek
        for($i=0;$i<$pg;$i++)

            echo(($i*$rows!=$ofs ? "\n<a
href='index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&rows=$rows&ofs=
".($i*$rows)."&sort=$sort'>&nbsp;&nbsp;&nbsp;" :
"\n<b>&nbsp;&nbsp;&nbsp;").($i+1).($i*$rows!=$ofs ? "&nbsp;&nbsp;&nbsp;</a>\" : "&nbsp;&nbsp;&nbsp;</b>");

```

```

        echo(" :: "); //oddelovac

        //dalsi

        echo($ap<$pg-1 ? "<a
href='index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=".($ofs+$rows)."&rows=$rows&sort=$sort'">&nbsp;další</a>" : "<b>&nbsp;další</b>");

        echo(" &nbsp; "); //oddelovac

        //posledni

        echo($ap<$pg-1 ? "<a
href='index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=".($pg-1)*$rows."&rows=$rows&sort=$sort'">konec</a>" : "<b>konec</b>");

        echo(" :: "); //oddelovac

?>
</span>
<label for="jmeno">Položek &nbsp; </label>
<SELECT name="rows" class="inputbox" size="1">
    <OPTION <? echo($rows==5 ? "SELECTED" : ""); ?> value=5>5</OPTION>
    <OPTION <? echo($rows==10 ? "SELECTED" : ""); ?> value=10>10</OPTION>
    <OPTION <? echo($rows==15 ? "SELECTED" : ""); ?> value=15>15</OPTION>
    <OPTION <? echo($rows==20 ? "SELECTED" : ""); ?> value=20>20</OPTION>
    <OPTION <? echo($rows==25 ? "SELECTED" : ""); ?> value=25>25</OPTION>
    <OPTION <? echo($rows==30 ? "SELECTED" : ""); ?> value=30>30</OPTION>
</SELECT>
<input type="image" src="graf/buttons/button_ok.gif" width="33"
height="19" id="submit">
<br/>
</form>

<? /*Řazení podle jednotlivých parametrů*/
$a=abs($sort); ?>
<table class="noborder"><tr>
    <td>
        <INPUT type=hidden value=add name=act>
        <form action="
<?
echo("index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$ofs&rows=$
rows"); ?>" method="post">
    <span><b>&nbsp;&nbsp;Řazení</b></span>
    <input name="desc" type="hidden" value="0">
    <SELECT name=sort class="inputbox" size="1">
        <OPTION value=1 <? echo($a==1 ? "SELECTED" : ""); ?
        >>Zkratka+rok</OPTION>
        <OPTION value=2 <? echo($a==2 ? "SELECTED" : ""); ?
        >>Deadline</OPTION>
        <OPTION value=3 <? echo($a==3 ? "SELECTED" : ""); ?
        >>Notifikace</OPTION>
        <OPTION value=4 <? echo($a==4 ? "SELECTED" : ""); ?
        >>Finalní verze</OPTION>
        <OPTION value=5 <? echo($a==5 ? "SELECTED" : ""); ?
        >>Termín konání</OPTION>
        <OPTION value=6 <? echo($a==6 ? "SELECTED" : ""); ?
        >>Místo konání</OPTION>
        <OPTION value=7 <? echo($a==7 ? "SELECTED" : ""); ?
        >>Počet příspěvků</OPTION>
    </SELECT>
    <input type="image" src="graf/arrows/sipn.gif" name="asc">
    <input type="image" src="graf/arrows/sipd.gif" name="desc">
</td>

```



```

<? if($per) { /**jen pro OSOBNI***/ ?>
    <td>
        <form name="prio_form" action="<?
echo ("index.php?seznam=&arc=$arc&per=$per&rows=$rows&ofs=$ofs&sort=$sort"
); ?>" method="post">
        <p><b>&nbsp;  Priorita</b>
        <SELECT name="pri" class="inputbox">
        <OPTION <? echo($pri==0 ? "SELECTED" : ""); ?>
            value=0>Všechny</OPTION>
        <OPTION <? echo($pri==1 ? "SELECTED" : ""); ?> value=1
            class="prio1">Obyčejná</OPTION>
        <OPTION <? echo($pri==2 ? "SELECTED" : ""); ?> value=2
            class="prio2">Zajímá mě</OPTION>
        <OPTION <? echo($pri==3 ? "SELECTED" : ""); ?> value=3
            class="prio3">Zajímá mě - reakce</OPTION>
        </SELECT>
        <input type="image" src="graf/arrows/sippg.gif">
        </form>
    </td></tr></table>

<?     } ?>

<?     /**jen pro ARCHIV***/
        if($arc)
        {
            if(mysql_num_rows($dbQuery=query("SELECT
id_konf,dat_zacatek,dat_konec,zkratka,rok FROM konference WHERE
MONTH(dat_zacatek)=MONTH(curdate()) AND
YEAR(dat_zacatek)<YEAR(curdate())"))>0)
            {
                echo("<table id=\"listconf_archiv\" >".
                    "<tr style=\"background:white;\">".
                    "<th colspan=3 align='center'>Minulé roky ve stejný měsíc".
                    "</th></tr>");
                echo("<tr class=\"textbg-3\">".
                    "<td>Zkratka+rok</td><td>Termín konání</td>".
                    "<td>&nbsp;  </td></tr>");

                $n=0;
                while($dbRS=mysql_fetch_array($dbQuery))
                {
                    echo("<tr class=\"textbg-$n\">".
                        "<td><a
href='index.php?info=&id=$dbRS[0]&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$
ofs&rows=$rows&sort=$sort'>$dbRS[3] $dbRS[4]</a>
                        </td>".
                        "<td>".xsqldate($dbRS[1])." -
                        ".xsqldate($dbRS[2])."</td>".
                        "<td><a href='index.php?renew=&id=$dbRS[0]'>obnovit</a></td>".
                        "</tr>");
                    $n=1-$n;
                }
                echo("</table>");
            }

            /**odkazy na jednotlivé roky**/

            if(mysql_num_rows($dbQuery=query("SELECT year(dat_zacatek) FROM
konference WHERE konference.archiv='1' OR TO_DAYS(NOW())-
TO_DAYS(dat_zacatek)>31 GROUP BY year(dat_zacatek) ORDER BY
dat_zacatek"))>0)
            {
                echo("<table id=\"listconf_archiv\" >".
                    "<tr style=\"background:white;\">".
                    "<th align='center'>Archiv podle minulých let</th></tr>");

                echo("<tr class=\"textbg-3\"><td>");

```

```

while ($dbRS=mysql_fetch_array($dbQuery))
{
    echo("<a
href='index.php?seznam=&arc=$dbRS[0]&per=$per&pri=$pri&sk=$sk&ofs=$ofs&ro
ws=$rows&sort=$sort'>$dbRS[0]</a>&nbsp;");
}
echo("</td></tr></table>");
}

if(mysql_num_rows($dbQuery=query("SELECT
konference.id_konf,zkratka,rok,deadline,notification,finalversion,dat_zac
atek,dat_konec,mesto,stat,(SELECT sum(priprispevku) FROM osobkonf WHERE
id_konf=konference.id_konf),(SELECT sum(posprispevku) FROM osobkonf WHERE
id_konf=konference.id_konf) AS prispevku".($per ? ",osobkonf.priorita" :
"")." FROM konference $join WHERE $wh ORDER BY $ob LIMIT $ofs,$rows"))>0)
{
    ?>
    <table id="listconf">
    <tr class="textbg-3">
    <th class="sirka"><a href="<?
echo("index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$ofs&rows=$
rows&sort=1"); ?>">Zkratka+rok</a></th>
    <th><a href="<?
echo("index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$ofs&rows=$
rows&sort=2"); ?>">Deadline</a></th>
    <th><a href="<?
echo("index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$ofs&rows=$
rows&sort=3"); ?>">Notifikace</a></th>
    <th><a href="<?
echo("index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$ofs&rows=$
rows&sort=4"); ?>">Finální verze</a></th>
    <th><a href="<?
echo("index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$ofs&rows=$
rows&sort=5"); ?>">Termín konání</a></th>
    <th><a href="<?
echo("index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$ofs&rows=$
rows&sort=6"); ?>">Místo</a></th>
    <th><a href="<?
echo("index.php?seznam=&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$ofs&rows=$
rows&sort=7"); ?>">Příspěvků</a></th>
    <th>&nbsp;</th>
    </tr>
    <?
    $n=0;
    while($dbRS=mysql_fetch_array($dbQuery))
    {
        echo("<tr class='".($per ? "prio".$dbRS[12] : "textbg-$n").">".
            "<td".($per && $pri==0 ? " class='prio".$dbRS[11]."' :
            "").">".
            "<a
href='index.php?info=&id=$dbRS[0]&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$
ofs&rows=$rows&sort=$sort'>$dbRS[1] $dbRS[2]</a></td>". //zkratka+rok
            "<td> ".xsqldate($dbRS[3])." </td>". //deadline
            "<td> ".xsqldate($dbRS[4])." </td>". //notifikace
            "<td> ".xsqldate($dbRS[5])." </td>". //finalni verze
            "<td> ".xsqldate($dbRS[6])." - ".xsqldate($dbRS[7])." </td>".
            //termin konani
            "<td> $dbRS[8], $dbRS[9] </td>". //misto
            "<td> ".$dbRS[10]==" ? 0 : $dbRS[10])." / ".$dbRS[11]==" ? 0 :
            $dbRS[11])." </td>".
            "<td class='odk'><a
href='index.php?info=&id=$dbRS[0]&arc=$arc&per=$per&pri=$pri&sk=$sk&ofs=$
ofs&rows=$rows&sort=$sort'>&nbsp;info</a><br>".
    "<a href='index.php?edit1=&id=$dbRS[0] &per=$per'>&nbsp;edit</a><br>".
    "<a href='index.php?forum=&id_konf=$dbRS[0]'">&nbsp;forum</a></td>".
    "</tr>";
    $n=1-$n; //to zaruci ze se bude v $n stridat 0 a 1

        }
?>
    </table>
<?
    }
?>

    <p id="up_and_back">
         <a href="">zpět</a>
         <a href="#zacatek">nahoru </a>
    </p>
<? if($per) { /***jen pro OSOBNI***/ ?>
<p id="fleft">Vysvětlivky:&nbsp;&nbsp;&nbsp;</p>
<p id="fleft">&nbsp;&nbsp;&nbsp;Obyčejná konference&nbsp;&nbsp;&nbsp;</p>
<p id="fleft">&nbsp;&nbsp;&nbsp;Zajímá mě - konference&nbsp;&nbsp;&nbsp;</p>
<p id="fleft">&nbsp;&nbsp;&nbsp;Zajímá
mě reakce - konference&nbsp;&nbsp;&nbsp;</p>
<?    }    ?>

```

uziv.php: (komunikace s uživatelem)

```

<?php //ISO 8859-2 = +ěščřžýáíéú
if(!isset($indexphp) || (isset($indexphp) && !$indexphp) ||
isset($_GET["indexphp"]))
echo("<script>document.location='index.php';</script>");
if(isset($_GET["logout"]))
    { // "odhlaseni"
    $_SESSION["uid"]=$uid=0;
    $_SESSION["uname"]=$uname="nepřihlášen";
    }
if(isset($_SESSION["uid"]))
    { //pokud je nastavena session promenna - vezmeme udaje z ni
    $uid=$_SESSION["uid"];
    $uname=$_SESSION["uname"];
    }
else if($fr=@fopen("login.txt","r")) { //jinak cteme uziv.jmeno ze
souboru
$uname=fread($fr,100); //maximalne 100 znaku
fclose($fr);

if(mysql_num_rows($dbQuery=query("SELECT uid FROM uziv WHERE
uname='$uname' LIMIT 1"))>0)
    { //a podle uziv.jmena bud dohledame uid
    $dbRS=mysql_fetch_array($dbQuery);
    $_SESSION["uid"]=$uid=$dbRS[0];
    }

```

```
    $_SESSION["uname"]=$uname;
    query("UPDATE uziv SET datlog=NOW() WHERE uid='$uid';"); //nastavnime
datum loginu
}
//nebo nove uid vytvorime a priradime
else $uid=query_id("INSERT INTO uziv(uid,uname,datlog) VALUES
(NULL, '$uname',NOW());");
}
else { //a pokud neni ani session ani soubor - uzivatel neni prihlasen
$_SESSION["uid"]=$uid=0;
$_SESSION["uname"]=$uname="nepřihlášen";
}
?>
```