

Slovenská technická univerzita v Bratislave  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY  
Katedra informatiky a výpočtovej techniky

## System na podporu posudzovania projektov

Pôvodne: *Softvérová podpora organizovania  
vedeckej konferencie*

Tímový projekt

### **Tím 4:**

Peter Kósa (kosa@decef.elf.stuba.sk)

Peter Lopeň (loopenpet@decef.elf.stuba.sk)

Richard Richter (virgo@fornax.sk)

Róbert Trebula (trebular@decef.elf.stuba.sk)

Ján Zázrivec (zazrivec@decef.elf.stuba.sk)

10. máj 2001

# Zadanie<sup>1</sup>

Pre úspešné zorganizovanie vedeckej konferencie treba zvládnuť niekoľko typických aktivít, ako sú:

- prezentácia konferencie;
- prihlasovanie účastníkov na konferenciu;
- zasielanie a oponovanie príspevkov;
- vytlačenie zborníka, apod.

Hoci sa vedecké konferencie líšia svojím obsahom a veľkosťou, podstata činností spojených s organizovaním konferencie je podobná.

Cieľom projektu je vytvoriť softvérový systém podporujúci organizovanie vedeckých konferencií. Organizátorom a programovému výboru má systém pomôcť prezentovať, plánovať a riadiť vedeckú konferenciu. Systém musí zohľadniť rôzne alternatívy pri organizovaní konferencie a má byť konfigurovateľný podľa špecifických požiadaviek a potrieb konferencie. Systém musí diferencovať používateľov v závislosti od role, ktorú hrajú vo vedeckej konferencii.

Systém musí umožniť základné aktivity pri organizovaní konferencie, ako sú:

- prihlasovanie účastníkov na konferenciu;
- elektronické posielanie a prijímanie príspevkov;
- evidovanie údajov o prihlásených príspevkoch a účastníkoch.

Systém musí poskytnúť podporu programovému výboru konferencie pri spoločnej komunikácii a rozhodovaniach týkajúcich sa najmä oponovania a výberu príspevkov. Treba uľahčiť najmä:

- priradovanie oponentov ku príspevkom;
- posielanie príspevkov na oponovanie;
- monitorovanie aktuálneho stavu v procese oponovania.

Vytvorený produkt by mal poslúžiť pri organizovaní medzinárodnej konferencie ADBIS 2002.

---

<sup>1</sup>Pôvodné zadanie sa týkalo *Softvérovej podpory organizácie vedeckej konferencie*. Od kapitoly 5 Projektovej dokumentácie je však zadanie modifikované.

## Úvod – štruktúra dokumentácie

Súbor dokumentov, ktoré nasledujú, je výsledkom práce tímu Con4U, tímu č. 4 v školskom roku 2001/02. Počas práce na projekte došlo k zmene jeho cieľového nasadenia, čo sa nutne muselo odraziť aj v dokumentácii (ako bude zrejmé z Projektovej dokumentácie). Upozorňujeme na to, aby nedošlo k nedorozumeniam, prečo sa tak veľa rozpráva o konferencii.

### Štruktúrovanie dokumentov

Odvzdávaná práca obsahuje nasledujúce dokumenty:

- **Projektová dokumentácia**, ktorá sa zaoberá postupom riešenia nášho projektu. Dokumentuje analýzu, špecifikáciu, návrh a implementáciu aj s prípadnými revíziami (ako je napr. podstatná kapitola 5, kde sa popisuje úprava špecifikácie).
- **Používateľská príručka** k systému Con4U – konkrétne pre jeho inštanciu *Záverečné projekty 2001/02*. Obsahuje aj kapitolu o inštalácii systému.
- **Posudok práce konkurenčného tímu po etape 1** – tj. posudok, ktorý sme napísali my na prácu tímu Confess.
- **Posudok našej práce po etape 1**, ktorý vypracoval tím Confess. Tento posudok sa zaoberá naším dokumentom po kapitolu 4.1 včítane.
- **Naše vyjadrenie sa k posudku po etape 1.**
- **Posudok práce konkurenčného tímu po etape 2.**
- **Posudok na našu prácu po etape 2.**
- **Dokumentácia k riadeniu projektu**, ktorá obsahuje všetky zápisy, ponuku, plány a ostatné dokumenty, ktoré s riadením súvisia.
- Prípadné ďalšie dokumenty k riadeniu, ktoré vznikli ako samostatné šablóny, bude na ne odkazované z dokumentácie k riadeniu projektu.

Dokumenty sú v práci zoradené v uvedenom poradí. Dokumenty budú od seba oddelené farebnými papiermi, pričom každý z dokumentov má buď výrazný titul alebo samostatnú titulnú stranu. Každý samostatný dokument má samozrejme vlastné číslovanie strán.

Slovenská technická univerzita v Bratislave  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY  
Katedra informatiky a výpočtovej techniky

System na podporu  
posudzovania projektov  
Tímový projekt  
Projektová dokumentácia

**Tím 4:**

Peter Kósa (kosa@decef.elf.stuba.sk)

Peter Lopeň (loopenpet@decef.elf.stuba.sk)

Richard Richter (virgo@fornax.sk)

Róbert Trebula (trebular@decef.elf.stuba.sk)

Ján Zázrivec (zazrivec@decef.elf.stuba.sk)

10. máj 2001

# Obsah

<b>1</b>	<b>Úvod</b>	<b>i</b>
1.1	Úvod k podpore organizovania konferencie . . . . .	i
1.2	Úvod k podpore posudzovania projektov . . . . .	i
1.3	Použité notácie . . . . .	i
<b>2</b>	<b>Analýza</b>	<b>1</b>
2.1	Jestvujúce systémy . . . . .	1
2.1.1	Confman . . . . .	1
2.1.2	Edas . . . . .	2
2.1.3	Puma . . . . .	2
2.1.4	Witan Web . . . . .	3
2.2	Technologická analýza . . . . .	4
2.2.1	Služba WWW . . . . .	4
2.2.2	Platformy . . . . .	4
2.2.3	Webové prehliadače a servery . . . . .	5
2.2.4	HTML editory . . . . .	5
2.2.5	Kaskádové štýly (CSS) . . . . .	6
2.2.6	Design webovských stránok . . . . .	6
2.2.7	Podpora pre webovské aplikácie . . . . .	7
2.2.8	Databázové servery . . . . .	12
2.2.9	Závery technologickej analýzy . . . . .	12
<b>3</b>	<b>Špecifikácia</b>	<b>14</b>
3.1	Kontext systému . . . . .	14
3.2	Funkcie systému . . . . .	14
3.2.1	Funkcie systému z pohľadu oponenta . . . . .	14
3.2.2	Funkcie systému z pohľadu autora . . . . .	15
3.2.3	Funkcie systému z pohľadu účastníka . . . . .	15
3.2.4	Funkcie systému z pohľadu predsedu . . . . .	16
3.3	Model údajov . . . . .	16
3.3.1	Popis jednotlivých dátových entít: . . . . .	16
3.3.2	Popis relácií: . . . . .	17
3.4	Používatelia systému . . . . .	18
3.4.1	Anonymný používateľ . . . . .	18
3.4.2	Účastník konferencie . . . . .	18
3.4.3	Autor príspevku . . . . .	18
3.4.4	Oponent – člen programovej komisie . . . . .	19
3.4.5	Administrátor – predseda programovej komisie . . . . .	19
<b>4</b>	<b>Návrh</b>	<b>20</b>
4.1	Hrubý návrh dátového modelu . . . . .	20
4.1.1	Popis atribútov neväzobných dátových entít . . . . .	20
4.1.2	Popis atribútov väzobných dátových entít . . . . .	22
4.2	Architektúra systému . . . . .	22
4.3	Moduly systému . . . . .	23
4.3.1	Web-rozhranie . . . . .	23

4.3.2	Mail-rozhranie . . . . .	24
4.3.3	Databázové rozhranie . . . . .	24
4.4	Návrh formulárov pre web-rozhranie . . . . .	24
4.4.1	Štandardné formuláre . . . . .	26
4.4.2	Registration of Authors . . . . .	26
4.4.3	Registration of Participant . . . . .	26
4.4.4	My Registration . . . . .	27
4.4.5	My Papers . . . . .	28
4.4.6	My Reviews . . . . .	29
4.4.7	Administration . . . . .	30
4.5	Návrh dátového modelu – verzia 2 . . . . .	35
4.5.1	Logický model údajov . . . . .	35
4.5.2	Fyzický model údajov . . . . .	38
4.6	Revízia návrhu . . . . .	40
4.7	Návrh modulov systému . . . . .	41
4.7.1	Modul Registration . . . . .	41
4.7.2	Modul Papers (Author) . . . . .	41
4.7.3	Modul Reviews (Reviewer) . . . . .	42
4.7.4	Modul Administration (Administrator) . . . . .	42
4.7.5	Modul CustomPages . . . . .	45
4.7.6	Modul Phormation . . . . .	46
4.7.7	Modul spoločných funkcií (include.php) . . . . .	46
<b>5</b>	<b>Špecifikácia požiadaviek 2</b>	<b>47</b>
5.1	Zmeny v terminológii . . . . .	47
5.2	Zoznam požiadaviek . . . . .	47
5.3	Stav požiadaviek na konci projektu . . . . .	51
5.4	Dôsledky pre návrh . . . . .	52
<b>6</b>	<b>Implementácia</b>	<b>53</b>
6.1	Implementovaný dátový model . . . . .	53
6.2	Popis modulov a formulárov . . . . .	53
6.2.1	Knižnica phormation . . . . .	53
6.2.2	Kaskádne štýly . . . . .	56
6.2.3	Podporné PHP skripty . . . . .	56
6.2.4	Implementácia formulárov . . . . .	58
6.3	Vlastnosti systému (properties) . . . . .	64
6.4	Konfiguračné poznámky . . . . .	65
6.5	Navrhované vylepšenia, chyby . . . . .	65
<b>7</b>	<b>Test systému</b>	<b>67</b>
7.1	Akceptačný test Systému na odovzdávanie projektov . . . . .	67
7.2	Testovanie nepoužitých funkcií . . . . .	67
<b>8</b>	<b>Záver</b>	<b>69</b>

# 1 Úvod

Tento dokument popisuje naše riešenie zadania *Systém na podporu posudzovania projektov*<sup>2</sup>, ktoré je súčasťou našej práce na predmete Tímový projekt. Obsah prvej etapy (analýza, špecifikácie a hrubý návrh) je náplňou kapitol 2 až 4.1. V kapitolách 4.2 až 4.5 včítane je opísaný výsledok našej práce v rámci etapy 2 – ide o korekcie prvej etapy a o používateľskú príručku k prototypu. Od kapitoly 4.6 ďalej nasleduje dokumentácia k etape 3 (letný semester).

## 1.1 Úvod k podpore organizovania konferencie

Organizovanie vedeckej konferencie ponúka množstvo problémov. Náročnosť organizovania konferencie v dobe globalizácie – v dobe informačnej spoločnosti – neustále rastie. Dnes je možné zvládnuť aj veľké konferencie pomerne pružne – vyžaduje si to ale, samozrejme, iné prostriedky. Dnes organizátori konferencií disponujú možnosťami, ku ktorým ich predchodcovia nemali prístup – a to stačí ísť o jediné desaťročie dozadu. Elektronická komunikácia účastníkov pred samotnou konferenciou – najmä využitie technológie WWW – umožňuje zvládnuť prípravu konferencie za kratší čas, resp. za ten istý čas si môžu účastníci konferencie vymeniť viacej dôležitých informácií.

Ak je možné zvýšiť množstvo informácií, ktoré sa všetky zúčastnené strany dozvedia, vznikne potreba tieto informácie spracovať. Je možné, že sa o všetko postará skupina ľudí – samotní organizátori. Rovnako tak je ale možné, že časť úloh bude automatizovaná, pričom človek zasiahne, len ak to bude nutné. Je jasné, že nenastala doba, kedy stroje budú posudzovať príspevky posielané do konferencie, ale je možné napríklad priradiť príspevku kľúčové slová a na základe nich uľahčiť výber oponenta, alebo takýto výber priamo navrhnúť.

Z naznačeného vyplýva, že vhodný informačný systém by mohol byť neoceniteľným pomocníkom pri organizovaní vedeckej konferencie.

## 1.2 Úvod k podpore posudzovania projektov

Pri riešení problematiky sme sa zamerali na posudzovanie, čo bolo na KIVT FEI STU v konečnom dôsledku využité na podporu procesu odovzdávania a posudzovania projektov (pilotná prevádzka *Záverečné projekty 2001/02*). K tejto zmene došlo najmä pre istú podobnosť procesov prebiehajúcich na konferencii počas procesu akceptácie príspevkov a procesov prebiehajúcich na škole pri odovzdávaní takýchto projektov. Bližšie sa problematike venujeme v časti pojednávajúcej o nových požiadavkách – viď kapitolu 5.<sup>3</sup>

## 1.3 Použité notácie

Pri väčšine diagramov, ktoré sa objavajú v tejto dokumentácii, sa držíme notácie UML.

V kapitole 3.1 sme použili diagram, ktorého notácia je popísaná priamo na mieste.

Napokon pri popise formulárov a požiadaviek sme použili štruktúrovaný text, ktorého komponenty sú tiež bližšie vysvetlené priamo na mieste.

---

<sup>2</sup>Pôvodne *Softvérová podpora organizovania vedeckej konferencie*.

<sup>3</sup>Toto je možné považovať aj za náhradu časti pojednávajúcej o zmenách oproti predchádzajúcej etape. Tieto sú totiž (najmä v používaných termínoch) dosť zrejme – opäť viď kapitola 5.

## 2 Analýza

### 2.1 Jestvujúce systémy

Pred diskusiou so zákazníkom o požiadavkách na systém boli preštudované požiadavky, ktoré boli dodané ako súčasť zadania. Boli identifikované tieto základné požiadavky:

- plánovanie vedeckej konferencie;
- riadenie vedeckej konferencie;
- prezentovanie vedeckej konferencie;
- konfigurovateľnosť systému.

Vykonalí sme analýzu existujúcich systémov, ktoré sa zaoberajú manažmentom vedeckých konferencií. Ako hlavný zdroj referencií poslúžila stránka – ACM Special Interest Group Governing Board: Summary of Conference Management Software<sup>4</sup>. Táto stránka je síce z roku 1999, takže niektoré linky a nej boli nefunkčné, poskytla však pomerne veľký počet odkazov na jednotlivé systémy na manažovanie konferencií. Pri štúdiu existujúcich produktov sme postupovali v poradí, v akom boli na tejto stránke uvedené a pri každom produkte sme si všímali najmä nové rysy systému, ktoré prinášali nové dátové, funkčné či iné prvky.

V nasledujúcich podkapitolách bude stručný popis jednotlivých systémov so zameraním na nové vlastnosti, ktorými sa tento produkt odlišuje a ktoré budú zapracované do celkového zoznamu možných vlastností systému na menežovanie vedeckej konferencie.

#### 2.1.1 Confman

Nórsky systém Confman<sup>5</sup>, ktorého posledná verzia je z roku 2000, sa javí ako jeden z najprepracovanejších, čo sa týka používateľnosti. Je veľmi dobre konfigurovateľný a pomerne jednoducho poňatý.

Okrem funkcií bežných pre všetky takéto systémy, má navyše tieto pozoruhodné vlastnosti:

- riadené posielanie mailov účastníkom konferencie;
- on-line oponovanie príspevkov;
- abstrakty článkov na webe;
- pri registrácii účastníkov konferencie je možné vypňovať aj položky týkajúce sa platenia poplatkov za účasť;
- posudok je tu poňatý ako formulár s prednastavenými otázkami a možnými odpoveďami (vo forme select-boxov). Tieto otázky a odpovede sú konfigurovateľné.

V systéme sa dá identifikovať niekoľko typov používateľov s príslušnými prípadmi použitia:

- **autor príspevku** pridáva svoj príspevok do systému a je informovaný o postupe pri jeho oponovaní a prijímaní;

---

<sup>4</sup><http://www.acm.org/sigs/sgb/summary.html>

<sup>5</sup><http://confman.unik.no/~confman/ConfMan/>



- **člen programovej komisie** posudzuje jednotlivé príspevky a odporúča/neodporúča ich na prijatie;
- **účastník konferencie** podáva prihlášku s vyplnením predvolených (konfigurovateľných) položiek;
- **administrátor** riadi zasielanie mailov autorom príspevkov, priraduje články oponentom, prezerá a riadi prihlásených účastníkov na konferenciu. Navyše konfiguruje všetky konfigurovateľné položky v rôznych častiach systému.

Systém ako taký sa môže nachádzať v jednom zo stavov:

- prijímanie príspevkov a posudkov;
- hlasovanie komisie o tom, ktoré príspevky prijať a ktoré nie;
- finálna fáza a konečné rozhodnutie.

Zmeny aktuálnej fázy vykonáva samozrejme administrátor.

### 2.1.2 Edas

Systém Edas<sup>6</sup> z newyorskej Columbia University nemá na webe funkčnú demonštračnú verziu, takže informácie sme získavali len z dokumentácie.

Oproti už preštudovanému systému Confman sa líši v týchto smeroch:

- namiesto fáz konferencie sú tu **stavy príspevku** – podaný abstrakt, podaný príspevok, priradený na posudzovanie, posúdený, prijatý/neprijatý na konferenciu;
- člen programovej komisie si môže sám vybrať, ktoré príspevky bude oponovať;
- automatické priradovanie príspevkov oponentom podľa miery zhody kľúčových slov príspevku a preferencií oponenta;
- v systéme je možné podávať opravy (corrections) jednotlivých príspevkov po ich odoslaní, aj ako reakciu na posudky.

### 2.1.3 Puma

Systém Puma – Publication Management System<sup>7</sup> z inej newyorskej univerzity – Cornell University – bol pôvodne len prípadovou štúdiou použitia ich vlastného databázového systému Predator, no napriek tomu je na dobrej úrovni. Na web-stránke projektu sú odkazy na demonštračnú verziu systému, sú však nefunkčné.

V tomto systéme sú oddelené rozhrania (interfaces):

- **oponent** – posudzuje články;
- **administrátor** – priraduje články oponentom.

---

<sup>6</sup><http://www.cs.columbia.edu/~hgs/edas/edas.html>

<sup>7</sup><http://www.cs.cornell.edu/database/predator/puma/puma.html>

Pozoruhodnou vlastnosťou systému Puma je, že sa cez tento systém nedajú posilať samotné príspevky. Všetky príspevky do systému vkladá ručne administrátor. Tento nedostatok je vysvetľovaný tým, že poslanie článku cez web-rozhranie by bolo príliš pomalé a nestihlo by sa vykonať počas jedného behu CGI skriptu. Toto je potrebné uvážiť a otestovať.

Posudky sú modifikovateľné aj po odoslaní. Sú tu zavedené tzv. **vlákna posudkov** (review threads). Jedná sa o to, že sa vytvára diskusia oponentov o jednom článku, ktorá sa zaznamenáva do stromovej štruktúry priradenej k tomuto článku.

Oponent – nazývaný rozhodca – do systému zadáva svoje preferencie, články akého typu by chcel (prípadne nechcel) oponovať. Administrátor má možnosť vytvárať si správy vo forme poradí príspevkov podľa hodnotenia oponentov a to podľa rôznych kritérií.

Systém je „mäkký“, čo sa týka dodržiavania termínov skončenia jednotlivých fáz, takže je možné aj dodatočné posielanie posudkov, hlasov a podobne, ktoré však už v závislosti na prebiehajúcej fáze neovplyvnia výsledky skončených fáz.

Systém sa zaoberá **konfliktnými posudkami** – ak sú na jeden príspevok radikálne odlišné posudky od niekoľkých oponentov, tieto konflikty sa musia vyriešiť. Práve na to sú vhodné diskusie k jednotlivým článkom. Z takejto diskusie by malo vyplynúť, ktorý z posudkov je správnejší, prípadne kto z oponentov svoj posudok zreviduje.

Puma má podrobne rozčlenené fázy – stavy systému:

1. posielanie abstraktov;
2. posielanie príspevkov;
3. posielanie preferencií oponentov;
4. priradenie oponentov príspevkom;
5. posielanie posudkov;
6. rozriešenie konfliktných posudkov;
7. príprava na stretnutie programovej komisie – generovanie reportov podľa rôznych kritérií;
8. aplikácia výsledkov stretnutia programovej komisie – posielanie oznamu o prijatí/neprijatí príspevku autorom.

#### 2.1.4 Witan Web

Produkt Witan Web<sup>8</sup> z kanadskej Národnej výskumnej rady je v krajine svojho vzniku hojne používaný. Prináša niekoľko nových myšlienok, ktoré by mohli byť zapracované do návrhu nášho systému.

Pretože je nasadzovaný v multijazyčnej krajine, zaoberá sa znakovými sadami a jazykmi dokumentov.

Oponenti môžu prostredníctvom systému Witan Web žiadať o konzultáciu, pomoc od iných oponentov týkajúcu sa niektorého posudzovaného príspevku.

Skoro v každom formulári je pole „dodatočné informácie“, kam sa zapisujú všetky záznamy pre ktoré nie sú predvolené otázky.

Zaujímavosťou je položka „študentský príspevok“ vo formulári pridávania článku.

Položky formulárov v tomto systéme nie sú konfigurovateľné priamo prostredníctvom web-rozhrania, čo je asi najväčší nedostatok tohoto systému.

---

<sup>8</sup><http://witanweb.iit.nrc.ca>

## 2.2 Technologická analýza

Po začiatkovej analýze zadania sa ako nevyhnutná ukázala potreba hlbšie sa oboznámiť s technologickými postupmi a metódami umožňujúcimi realizovať riešenie ako klient/server aplikáciu. Z tohto dôvodu sa v nasledujúcom texte zaoberáme jednotlivými technológiami a skúmame ich z hľadiska použitia v navrhovanom systéme.

### 2.2.1 Služba WWW

Základnou technológiou, ktorú použijeme v riešení, je služba World Wide Web (WWW). Táto služba vznikla na podo CERNu v rokoch 1989–90 a je založená na troch pojmoch: **HTML**, **HTTP** a **URL**.

**HTML – Hypertext Markup Language** HTML je značkovací jazyk založený na SGML na popis štruktúry webovských stránok. Je platformovo nezávislý. Prvá formálna špecifikácia je HTML 2.0, ktorá obsahuje prostriedky pre základné formátovanie a štrukturovanie dokumentu, možnosť vkladať obrázky a vytvárať formuláre. Z dôvodu neštandardizovaného rozširovania výrobcami prehliadačov vzniklo v roku 1996 konzorcium W3C, ktoré vybralo spoločnú podmnožinu rozšírení a schválilo HTML 3.2.

V súčasnosti je aktuálna verzia HTML 4.0, ktorá predstavuje veľký skok vpred. Z najdôležitejších vylepšení treba spomenúť podporu kaskádových štýlov, skripty vložené do stránky, multimedialne objekty, rozšírené možnosti tabuliek, podporu viacerých jazykov, rozšírenie formulárov a rámy. Takmer kompletnú podporu HTML 4.0 majú prehliadače Internet Explorer 4.0 a 5.0 a tiež Netscape Navigator. V roku 1999 vzniká XHTML 1.0. Ide o štandard HTML 4.0 upravený tak, aby stránky v XHTML predstavovali XML dokumenty.

**HTTP – Hypertext Transfer Protocol** HTTP predstavuje protokol na prenos stránok medzi webovským serverom a prehliadačom. Ide o jednoduchý aplikačný a bezstavový protokol vystavaný nad protokolom TCP modelom požiadavka/odpoveď, čo prináša niektoré nevýhody pre webovské aplikácie. Existuje niekoľko verzií HTTP 0.9, HTTP 1.0, HTTP 1.1 – najnovší, podporovaný vo väčšine serverov a prehliadačov.

**URL – Uniform Resource Locator** URL je adresa, ktorá jednoznačne identifikuje zdroj v Internete. Nie sú obmedzené len na službu WWW, ale pokrývajú celé spektrum služieb (WWW, FTP, e-mail, telnet). Počas komunikácie servera a prehliadača sa pomocou protokolu HTTP vymieňajú požiadavky používateľa (URL) a odpovede servera (HTML).

### 2.2.2 Platformy

Náš projekt bude klient/server aplikácia. Preto musíme uvažovať, na akom operačnom systéme bude bežať klientská a serverová časť systému. Pri ďalšej analýze budeme uvažovať, ktoré techniky sú prístupné pre ktorú platformu. Existujú dve veľké rodiny operačných systémov: Unix a Windows.

**Windows** Do tejto rodiny patria operačné systémy Windows 9x/ME, Windows NT/2000/XP. Tieto operačné systémy panujú v oblasti stolných počítačov vďaka ich pohodlnému používaniu. Majú zabudovaný webovský prehliadač Microsoft Internet Explorer. Windows NT podporuje procesory x86, DEC Alpha, MIPS R4000, PowerPC.

**Unix** Pojem UNIX predstavuje veľkú rodinu operačných systémov. Veľmi významným predstaviteľom tejto rodiny je Linux. Je voľne šíriteľný s otvoreným zdrojovým kódom, čo podporuje jeho rozšírenie. Dnes predstavuje robustné a spoľahlivé riešenie. Podporuje obrovskú škálu procesorov. Bežne sa v jeho distribúciach dodáva prehliadač Netscape Navigator, server Apache, niektoré distribúcie obsahujú dokonca aj databázové servery MySQL alebo PostgreSQL. V oblasti serverov je veľmi populárny pre jeho dostupnosť, spoľahlivosť a rýchlosť.

### 2.2.3 Webové prehliadače a servery

V predchádzajúcej kapitole sa niekoľkokrát vyskytli slová ako prehliadač alebo webový server. Je preto vhodné, aby sme sa teraz zaoberali na trhu dostupnými prehliadačmi a servermi – a to predovšetkým z pohľadu rôznych platforiem.

#### WWW prehliadače

- **Microsoft Internet Explorer** patrí medzi najrozšírenejšie prehliadače na platforme Windows. V súčasnosti je aktuálna verzia 5.5.
- **Netscape Navigator** existuje vo verziách pre platformu Windows aj Unix. Vývojári ocenia XML, JavaScript, HTML4 a CSS podporu. Aktuálnou je verzia 6.0.
- Jeden z novších prehliadačov predstavuje **Opera 5.0**, ktorá ponúka podporu pre väčšinu webovských štandardov (HTML 4, XML, SSL, CSS, JavaScript a DOM). Existuje pod operačné systémy Windows aj Unix.

V ďalšom texte sa budú uvedené technológie skúmať z pohľadu týchto prehliadačov.

#### WWW servery

- K najrozšírenejším WWW serverom patrí bezpochyby server **Apache** (60% podiel na trhu), ktorého verzia 1.3 existuje pre platformy Windows aj Unix. Je voľne šíriteľný a jeho licencie umožňuje aj komerčné využitie. Má modulárnu architektúru a je ľahko rozšíriteľný. Je robustný a bezpečný.
- Medzi ďalšie servery, ktoré budeme v nasledujúcom texte uvažovať, patria **Internet Information Server 5.0 (IIS)**, určený na komerčné nasadenie a **Personal Web Server (PWS)** vhodný pre jednoduchšie a menej rozsiahle aplikácie (prácu off-line). Oba sú určené pre operačný systém Windows. IIS je primárny server pre prostredie Windows NT verzia Server aj Workstation. Dynamické stránky nie sú ponúkané pomocou CGI, ale používa sa programovacie rozhranie Internet Server Application Programming Interface (ISAPI). Aplikácie ISAPI sú v podstate dynamické knižnice. Príklad ich použitia predstavujú aplikácie ASP.

### 2.2.4 HTML editory

Z dôvodu uniformného výstupu a zjednodušenia tvorby HTML stránok je vhodné sa ešte pred začatím samotných prác rozhodnúť, aký spôsob tvorby použijeme. HTML kód sa dá samozrejme zapisovať pomocou obyčajných textových editorov ako je Notepad v prostredí Windows, či Vim alebo Emacs pod Unixom. Ak však HTML stránky vytvárame často je vhodné uľahčiť si prácu použitím editoru obsahujúceho podporu pre písanie HTML kódu. Z tohto pohľadu môžeme editory rozdeliť do niekoľkých skupín:

**Textové editory s podporou HTML** Obľúbenými predstaviteľmi tejto skupiny sú editory **HomeSite** a **HTML-Kit**, ktoré boli vytvorené špeciálne pre potreby jazyka HTML. Zvýraznenie syntaxe ponúkajú aj klasické editory ako **Vim** a **Emacs**.

**Štrukturálne editory** Oproti predchádzajúcej skupine ponúkajú dôležitú vlastnosť navyše – neumožňujú vytvárať syntakticky nesprávnu stránku. Do miesta ktoré editujeme môžeme vložiť iba tie tagy, ktoré sa tam môžu objaviť. Nestane sa nám teda, že stránka obsahuje skrížené elementy alebo chýbajúce ukončovacie tagy. Veľmi dobrým programom z tejto kategórie je **Wedit**. V praxi je možné použiť aj kombináciu editoru Emacs a nadstavby **PSGML**.

**WYSIWYG editory** Táto kategória zahŕňa editory, s ktorých podporou môžeme vytvárať HTML stránky bez znalosti HTML. WYSIWYG editor sa ovláda podobne ako textový procesor – píšeme text, môžeme meniť písmo, zarovnanie, vkladať obrázky a podobne. Výsledok sa však ukladá vo formáte HTML a počas editácie sa zobrazuje približne tak, ako bude zobrazený v prehliadači. Problémom týchto editorov je, že zaostávajú za vývojom HTML, a tiež to, že generovaný HTML kód je pre následnú ručnú editáciu neprehľadne sformátovaný. Najznámejším editorom z tejto skupiny je Microsoft FrontPage. Podobnú funkčnosť poskytujú aj Netscape Composer.

### 2.2.5 Kaskádové štýly (CSS)

Cascading style sheets (CSS) je deklaratívny jazyk odstraňujúci niektoré nevýhody klasického HTML. Štýly umožňujú definovať spôsob zobrazenia každého elementu na stránke. Štýl však nie je priamo súčasťou textu stránky, a tak môže byť zápis stránky prehľadnejší a dobre štruktúrovaný. Navyše štýly umožňujú definovať jednotný vzhľad určitého elementu v celom dokumente jedným spôsobom – nemusíme ho opakovať u každého elementu. Pokiaľ štýl uložíme do externého súboru, môže ho využívať viacero stránok naraz. Definícia designu všetkých stránok je potom na jednom mieste. Pri požiadavke na zmenu vzhľadu stránok stačí upraviť štýl a zmeny sa automaticky uskutočnia vo všetkých stránkach. Dobrú podporu CSS majú prehliadače Internet Explorer 4.0 a 5.0, Netscape Navigator 6.0 a Opera. Z týchto dôvodov by bolo vhodné použiť túto techniku aj v nami navrhovanom riešení.

### 2.2.6 Design webovských stránok

Na tomto mieste uvádzame niekoľko základných pravidiel, ktoré by sa mali dodržať pri návrhu webovského interfejsu celej aplikácie:

#### Grafický design

- Vhodné kombinácie farieb;
- farba pozadia a textu by mala byť zvolená tak, aby sa text dobre čítal;
- nepoužívať obrázky na pozadí;
- ako farba pozadia je vhodná biela a svetlé odtiene ďalších farieb;
- u všetkých grafických objektov uvádzať alternatívne textové popisy.

## Organizácia stránok

- Hypertext umožňuje vytvoriť ľubovoľnú štruktúru, ale človek je zvyknutý na konzervatívnejšie usporiadania – väčšinou na hierarchiu;
- na každej stránke by mala byť navigačná lišta;
- jedna stránka by mala obsahovať logický celok informácií;
- stránka by mala byť dobre štruktúrovaná pomocou nadpisov a zoznamov;
- obsah by sa mal oddeliť od vzhľadu a mali by sa použiť CSS.

## Odkazy

- Vyhnúť sa technickým pojmom;
- odkazy by mali byť interaktívne (napr. meniť farbu);
- v rámci jedného servera používať relatívne odkazy.

## Dokument

- Každá stránka by mala obsahovať identifikačné údaje;
- malo by byť jasné, na ktorom serveri sa nachádzame;
- používať metadata na lepšiu klasifikáciu a vyhľadávanie.

### 2.2.7 Podpora pre webovské aplikácie

Pokiaľ chceme prostredie webovského prehliadača využiť ako platformu pre tvorbu aplikácie, máme k dispozícii mnoho možností. V zásade sa však dajú rozdeliť na dve skupiny podľa toho, kde sa vykonáva kód aplikácie. Jednu z nich predstavujú aplikácie, ktoré bežia v samotnom prehliadači (klientovi). V tejto skupine sú k dispozícii nasledujúce riešenia:

#### 1. Klientské skripty – JavaScript, VBScript a ďalšie

Priamo do kódu HTML stránky môžeme zapisovať krátke programy v jednoduchých jazykoch ako je JavaScript, JScript alebo VBScript. Tieto sa využívajú predovšetkým na kontrolu údajov vyplnených používateľom a na zvýšenie interaktivity stránky. Jazyk JavaScript je dokonale štandardizovaný ako ECMAScript. Podpora uvažovaných prehliadačov je nasledovná:

- **JavaScript 1.0** – Netscape 2.0 a IE 3.0;
- **JavaScript 1.1** – Netscape 3.0 a IE 4.0;
- **JavaScript 1.2** – Netscape 4.0 a čiastočne aj IE 4.0;
- **JavaScript 1.3** – Netscape 4.5 a IE 5.0.
- **VBScript** pracuje iba v Internet Exploreri.

**Dynamické HTML (DHTML)** Dynamické HTML stavia na využití klientských skriptov. Navyše je teraz celá stránka a ostatné komponenty prehliadača prístupná pomocou objektového modelu dokumentu (DOM). Obsah stránky sa dá interaktívne meniť aj po jej načítaní, čo umožňuje vytvárať skutočne prispôsobivé užívateľské rozhranie. Dynamické HTML je podporované prehliadačmi Internet Explorer 4.0 a 5.0 ako aj Netscape Navigatorom od verzie 4.0.

**Java applety** Java applet je program napísaný v jazyku Java, ktorému sa vyhradí časť HTML stránky. Na tejto časti si applet hospodári úplne vo vlastnej réžii. Preto sa applety hodia pre aplikácie, kde je potrebná vysoká interaktivita pri komunikácii s používateľom a klasické klientské skripty už nestačia. Java applet má prístup k väčšine knižnic jazyka Java, a preto môže na komunikáciu s ostatnými aplikáciami používať takmer ľubovoľný protokol a umožňuje vytvoriť multimediálne aplikácie (grafika, animácia, hudba). Použitie appletov je obmedzené na novšie prehliadače, ktoré ich však vo väčšine podporujú.

**ActiveX komponenty** Použitie ActiveX komponentov je rovnaké ako u Java appletov. Rozdiel je v tom, že ActiveX komponenty sú spustiteľné iba na platforme Windows. Nižšia prenositeľnosť je však vynahradená väčším výkonom oproti Java.

## 2. Skriptovanie na serveri

Druhú skupinu predstavujú riešenia, pri ktorých je aplikácia vytvorená tak, že nejaký program na serveri dynamicky podľa potreby generuje HTML kód. Klientské technológie sú v tomto prípade použité prevažne na zlepšenie funkčnosti užívateľského rozhrania a pri zvyšovaní interaktivity stránok. Navyše, klientské technológie nemajú prístup k databázam, ktoré sú uložené na serveri. Generovanie stránky v dobe požiadavky tak umožňuje reagovať na konkrétne podmienky a požiadavky užívateľa.

**Server Side Includes (SSI)** SSI sú jednoduché príkazy, ktoré sa zapisujú priamo do HTML stránky ako komentár. Umožňujú vykonávanie jednoduchých činností, ako je vloženie iného súboru do stránky alebo vypísanie dátumu poslednej modifikácie dokumentu. Dnes podporujú SSI všetky bežne používané servery.

**CGI skripty** CGI skripty majú svoj názov odvodený od rozhrania CGI – Common Gateway Interface. Toto rozhranie definuje spôsob, ktorým webový server komunikuje s programami a umožňuje jeho prepojenie s ľubovoľným programovacím jazykom. Pokiaľ si užívateľ vyžiada od servera stránku, ktorá má byť vygenerovaná CGI skriptom, predá server skriptu parametre zadané užívateľom a spustí ho. Skript potom na svojom štandardnom výstupe vygeneruje odpoveď pre prehliadač v jazyku HTML. Táto odpoveď je zachytená webovým serverom a zaslaná prehliadaču. CGI skripty sú najvšeobecnejším a najuniverzálnejším riešením pre webové aplikácie. Dajú sa písať v ľubovoľnom jazyku – najpoužívanejšie sú príkazové shelly, Perl, C/C+. Ich nevýhodou je však vysoká prevádzková réžia. Pre každú požiadavku musí byť vytvorený nový proces, v ktorom sa spúšťa CGI skript. Preto sa toto riešenie nedá použiť pre servery, na ktoré pristupuje niekoľko desiatok užívateľov súčasne. Podporu pre rozhranie CGI má dnes každý webový server. Pokiaľ sa na tvorbu skriptov použije jazyk nezávislý

na platforme, môžeme výslednú aplikáciu použiť na webovských serveroch v Unixe i vo Windows. Pri použití CGI skriptov by sa mal klásť zvýšený dôraz na bezpečnosť.

**FastCGI skripty** FastCGI skripty sú snahou o zrýchlenie vykonávania CGI skriptov. Skripty upravené pre toto rozhranie sú pri prvej požiadavke zavedené do pamäti. Ďalšie požiadavky už obsluhuje program, ktorý už je zavedený v pamäti. FastCGI skripty napísané v jazyku C sú dnes asi najvýkonnejším riešením. Sú používané predovšetkým v prostredí Unixu.

**ISAPI a NSAPI moduly** ISAPI je špeciálne rozhranie pre komunikáciu medzi webovským serverom a aplikáciou. Umožňuje, aby bol program neustále zavedený v pamäti a nemusel sa znovu načítavať a spúšťať pre každú požiadavku. ISAPI sa používa so serverom IIS aj keď toto rozhranie podporujú aj niektoré ďalšie servery. Servery firmy Netsape používajú vlastné rozhranie NSAPI.

**Active Server Pages (ASP)** ASP je technológia Microsoftu určená pre tvorbu webovských aplikácií. Má modulárnu štruktúru, takže umožňuje použitie takmer ľubovoľného programovacieho jazyka. Microsoft štandardne ponúka VBScript a JScript. Od ďalších firiem sa dajú získať interpreteri Perlu a Pythonu. V ASP sa dajú bez problémov používať COM objekty. Nevýhodou ASP je ich úzke zviazanie s technológiami Microsoftu. ASP beží len v prostredí IIS a PWS.

**ColdFusion** ColdFusion je vyvinutý Allaire. Jeho syntax je veľmi podobný syntaxi HTML, čo uľahčuje začínajúcim programátorom webovských stránok jeho používanie. Dnes je dostupný aj pre Windows NT aj pre Linux. Jeho nevýhodou je, že nie je voľne šíriteľný.

**Personal Home Pages (PHP)** PHP je jednoduchý, voľne šíriteľný a pritom výkonný skriptový jazyk. Zapisuje sa priamo do HTML kódu a umožňuje prirodzeným spôsobom kombinovať HTML s príkazmi. PHP sa dá prevádzať na ľubovoľnom webovom serveri (platforma Windows alebo Unix), aj keď najlepších výsledkov sa dosiahne v spojení so serverom Apache (PHP je skompilované ako modul a je trvale uložené v pamäti). Navyše PHP obsahuje priamo podporu pre prácu s najrozšírenejšími databázovými servermi. Aktuálna je verzia 4.0.

**Servlety v Java** Java applety umožňujú beh javovských aplikácií v prehliadači. Podobne servlety zase umožňujú beh Javy na serveri a tým jej využitie pre webové aplikácie. Servlety predstavujú robustné riešenie a vďaka CORBE môžu komunikovať s ďalšími časťami informačného systému (databázy, ...).

**Java Server Pages (JSP)** Pri použití JSP sa podobne ako v ASP alebo v PHP priamo do HTML kódu zapisujú príkazy. V prípade JSP sú zapisované v jazyku Java. Špeciálny servlet zabezpečuje aby bola JSP stránka vždy po svojej modifikácii automaticky priložená do byte-code. To uľahčuje vývoj a na druhej strane je aplikácia značne



výkonná, pretože skripty sú vyvolávané už v preloženom tvare. Ide o pomerne novú technológiu, ktorý zataaľ podporujú iba servery od firmy Sun a Apache.

Všetky vyššie uvedené technológie môžeme rozdeliť do dvoch skupín podľa toho, či je pred použitím potrebné previesť program do binárnej podoby alebo či ide o interpretované jazyky. Pokiaľ vo vývoji aplikácie použijeme jazyk ako C/C++ alebo Java, je potrebné pred použitím aplikáciu skompilovať. To spomaľuje vývoj, ale prináša zrýchlenie počas behu aplikácie. Pri interpretovaných jazykoch ako PHP, JavaScript, Perl je situácia opačná. Rýchlejší vývoj je kompenzovaný pomalším vykonávaním. Úbytok rýchlosti však väčšinou nie je nijak významný vzhľadom na fakt, že skripty webovských aplikácií bývajú pomerne jednoduché.

### 3. Porovnanie jazykov pre klientské skripty

V prípade, ak by sme sa nejakú časť aplikácie rozhodli implementovať pomocou skriptov vykonávaných u klienta v prehliadači, je vhodné porovnanie dostupných skriptovacích jazykov.

**JavaScript** JavaScript je kompaktný, platformovo nezávislý, na objektovom prístupe založený jazyk na vývoj aplikácií pre Internet. Pôvodne bol navrhnutý firmou Netscape Communications Corporation a po jeho úspechu prišiel aj Microsoft s podobným jazykom pod názvom JScript. Príkazy JavaScriptu sú priamo vložené v HTML stránke a sú interpretované prehliadačom.

**VBScript** VBScript je skriptovací jazyk alebo presnejšie skriptovacie prostredie implementované podmnožinou programovacieho jazyka Visual Basic. Výsledkom zjednodušenia je produkt, ktorý je ľahko použiteľný. Je to asi aj najväčšia výhoda oproti JavaScriptu. Veľká sila VBScriptu spočíva v možnosti sprístupniť množstvo ActiveX komponentov. To je súčasne dôvodom jeho najväčšieho nedostatku – je podporovaný iba v prehliadačoch IE.

### 4. Porovnanie jazykov pre serverovské skripty

V prípade voľby skriptov na strane servera ako technológie ktorú bude naše riešenie využívať je potrebné bližšie preskúmať aj jednotlivé jazyky, ktoré prichádzajú do úvahy. Hľadiská, z ktorých sa treba na tieto jazyky pozeráť sú nasledujúce:

- **Výkonnosť** – je zrejmé, že kompilované jazyky sú rýchlejšie ako interpretované. Časť z toho je spôsobená potrebou zavedenia kópie interpretera nevyhnutného na spustenie programu napísaného v skriptovacom jazyku, ako je napríklad Perl. Ďalšie zrýchlenie kompilovaných programov je aj dôsledkom použitia optimalizácií závislých na platforme, pre ktorú je program kompilovaný. Väčšinou však rozdiel vo výkonnosti nie je rozhodujúcim faktorom pre voľbu jazyka.
- **Bezpečnosť** – vo všeobecnosti najviac bezpečnostných problémov pochádza z programovacích chýb a nedostatočného pochopenia prostredia, v ktorom skript existuje a beží.
- **Spôľahlivosť** – mal by sa použiť jazyk, ktorého stabilita sa na použitej platforme dokázala. Je vhodné, ak je jazyk stabilný na viacerých platformách.

- **Udržovateľnosť** – toto je často parameter, na ktorý sa berie najväčší ohľad pri voľbe programovacieho jazyka. Veľa programátorov používa interpretované jazyky z dôvodu, že sú ľahšie čitateľné, ľahšie na pochopenie, jednoducho udržiavateľné a testovateľné. Umožňujú inkrementálny, iteratívny vývoj a testovanie namiesto cyklu vytvorenie/modifikovanie – preloženie – testovanie – krokovanie. Tieto faktory môžu dramaticky urýchliť vývoj.
- **Prenositel'nosť** – prenositeľný kód je základom dobre navrhnutého programu. Aj v tomto prípade vychádzajú interpretované jazyky ako dobrá voľba, pretože väčšina interpreterov existuje pre viacero platforiem.

**Perl** Perl je interpretovaný jazyk optimalizovaný pre prehľadávanie textových súborov a získavanie informácií z týchto súborov a vytvorenie ďalšieho textového súboru na základe týchto informácií. Je tiež vhodný pre množstvo úloh súvisiacich so správou systému. Je vytvorený tak, aby bol skôr praktický (ľahko používateľný, efektívny, kompletný) ako krásny. Je dostupný a stabilný na platformách Unix aj Windows. Spolu s webovským serverom Apache vytvárajú výkonnú a populárnu kombináciu. Jedným z najznámejších projektov využívajúcich tieto dve technológie je projekt `mod_perl`, ktorý patrí k najstarším a najúspešnejším. Umožňuje vloženie Perl interpretera do Apache a tým sprístupňuje interné informácie servera programom v perle.

**Python** Python je prenositeľný, interpretovaný, objektovo-orientovaný jazyk. Má elegantnú syntax a malé množstvo vstavaných vysoko úrovňových údajových typov. Môže byť systematicky rozširovaný pridávaním nových modulov implementovaných v kompilovanom jazyku ako je C alebo C++. Tieto rozširujúce moduly môžu definovať nové funkcie a premenné alebo tiež nové dátové typy. Obsahuje úplnú možinu operácií s reťazcami (vrátane regulárnych výrazov) a odľahčuje používateľa od správy pamäte. Tieto a ďalšie vlastnosti ho robia ideálnym pre vývoj prototypov a ostatných ad-hoc programovacích úloh. Obsahuje tiež generický spôsob, ako konvertovať objekt do prúdu bajtov a opačne, čo môže byť použité pri tvorbe roznych distribuovaných modelov. Je k dispozícii na platforme Unix, Windows, DOS, OS/2 a Macintosh a existuje ako modul do webovského servera Apache.

**Tcl** Tcl je výkonný skriptovací jazyk použiteľný na väčšine dostupných patforiem. Ponúka všetky typické vysoko úrovňové programovacie prostriedky ako sú: reťazce premennej dĺžky, asociatívne polia, zoznamy, štruktúry, regulárne výrazy a schopnosť definovať procedúry priamo počas vykonávania programu. Všetky dátové štruktúry sú dynamicky alokované. Tcl je programovací jazyk určený na vkladanie do iných aplikácií ako konfiguračný a rozširujúci jazyk. Je navrhnutý tak, aby bol ľahko rozšíriteľný pomocou nových primitív napísaných v jazyku C. Tcl používa rovnakú reprezentáciu pre dáta aj pre program, čo umožňuje uloženie programu do premenných, súborov alebo databázy. Taktiež podporuje udalosťami riadené programovanie a IPC medzi viacerými Tcl aplikáciami.

**PHP** Jednou z veľa výhod PHP je jeho rýchlosť. Ak ASP používa VBScript, spustí COM objekt. Ak píše niečo, používa iný COM objekt, ďalší, ak pristúpi k databáze,

ďalší pre súborový systém, atď. V PHP všetko beží v jednom adresnom priestore, čo znamená väčšiu rýchlosť, lebo nevzniká potreba komunikovať s iným objektom v inom procese.

Oproti Perlu bol PHP vyvinutý pre generovanie dynamických webovských stránok, nie na všeobecné spracovanie textových súborov, a preto má zabudovaných veľa špeciálnych prostriedkov (napr. prístup k objektom HTML stránky cez premenné).

Pre programátorov skúsených v Java/JavaScript/C/C++/Perl je ľahko zvládnuteľná gramatika jazyka, nakoľko je blízka gramatike týchto jazykov. Ďalšou výhodou je integrácia s MySQL. PHP má veľa prostriedkov na správu a udržiavanie MySQL databázy.

### 2.2.8 Databázové servery

Ako jedna z častí v nvrhovanom riešení bude pravdepodobne vystupovať databáza. Je preto vhodné vykonať analýzu aj v tejto oblasti. Medzi najrozšírenejšie produkty v tejto oblasti patria Oracle, MS SQL Server, Sybase, Progress. Vzhľadom na fakt, že sa jedná o drahé softvérové produkty, zameriame sa skôr na analýzu voľne šíriteľných databázových serverov, medzi ktoré patria napríklad MySQL alebo PostgreSQL.

**MySQL** MySQL je jednoduchý databázový SQL server. Je priamo podporovaný PHP. Umožňuje na jednom počítači pracovať s viacerými databázami. Existuje pre platformu Windows aj Unix. Medzi hlavné výhody patrí:

- schopnosť obsluhovať neobmedzené množstvo používateľov súčasne;
- kapacita na uchovanie 50 000 000 záznamov;
- veľmi rýchle vykonávanie príkazov;
- ľahký a efektívny systém používateľských privilégií.

MySQL má navyše širokú používateľskú základňu, čo zabezpečuje, že kód je dobre testovaný.

**PostgreSQL** PostgreSQL je databázový server takmer úplne kompatibilný so štandardom SQL92. Podporuje teda väčšiu podmnožinu jazyka SQL ako MySQL. Sever beží na Unixe, klientské programy existujú pod Unix aj Windows. Podporuje vnorené príkazy SELECT, pohľady a transakcie. Obsahuje interfajsy pre C/C++, Perl, PHP, Python, TCL, bash, ODBC. Jeho sila spočíva vo veľkej rozširovateľnosti: používateľom definované údajové typy, operátory, funkcie. Taktiež podporuje dedenie a má bohatú množinu dátových typov: boolean, časové intervaly, viac dimenzionálne polia a 1D geometrické typy. Navyše je dobre zdokumentovaný. Jeho najväčšou nevýhodou oproti MySQL je rýchlosť. Ďalšou nevýhodou je fakt, že nie je hierarchický – databázy môžu obsahovať iba tabuľky a nie ďalšie databázy. Podobne ako MySQL aj PostgreSQL je priamo podporovaný PHP.

### 2.2.9 Závěry technologickej analýzy

Pre náš projekt sme si vybrali kombináciu produktov uvedenú v tabuľke 2. Hlavné kritériá nášho výberu v poradí podľa dôležitosti boli:

- Platformová nezávislosť, prenositeľnosť;

webovský server	<b>Apache</b>
skriptovanie na strane servera	<b>PHP</b>
skriptovanie na strane klienta	<b>JavaScript</b>
databáza	<b>MySQL</b>

Tabuľka 2: Vybrané produkty pre náš systém

- efektívnosť;
- nízka cena tvorby a používania;
- jednoduchá implementácia.

Webovský server Apache je dostupný na oboch platformách (dokonca bezplatne) a má veľmi dobré ukazovatele efektívnosti.

Jazyk PHP je priamo podporovaný serverom Apache ako jeho modul pre obidve platformy a je najrýchlejším jazykom pre interpretované skripty. Programátori nášho tímu už majú skúsenosti s jeho používaním.

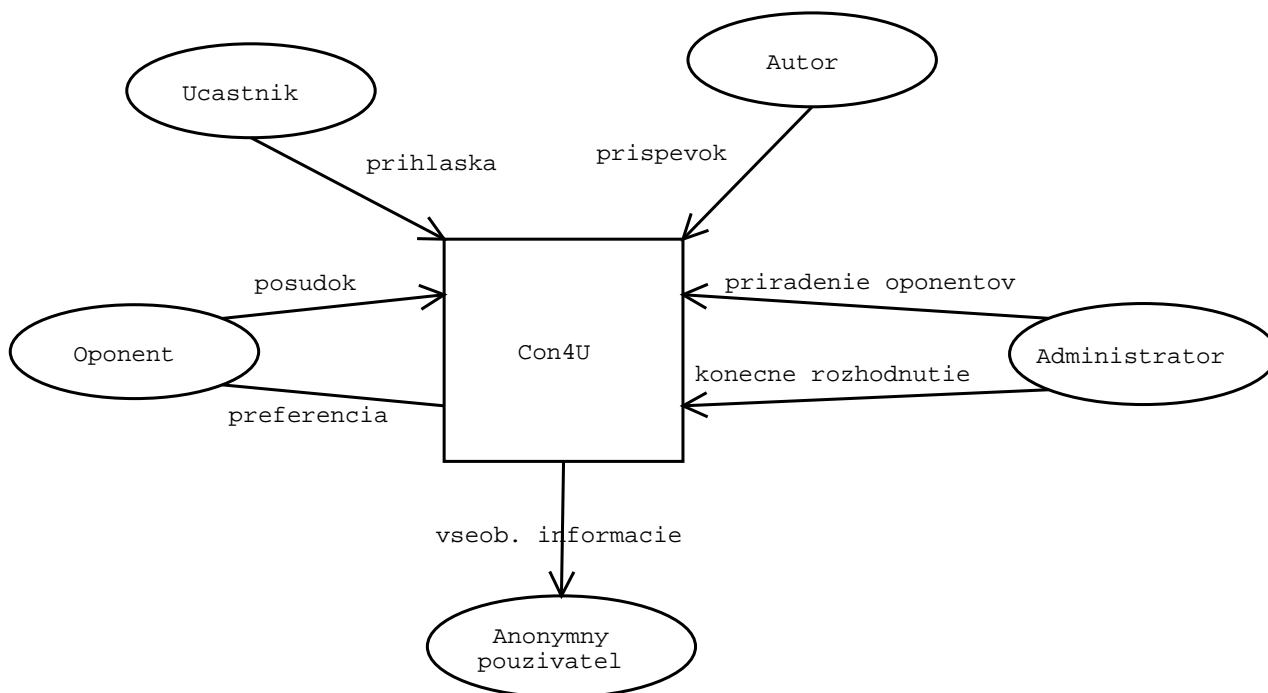
Databázový systém MySQL, tak isto ako PHP a Apache, je platformovo nezávislý. Je veľmi rýchly aj pri väčšom množstve dát. Všetky vyššie uvedené prostriedky možno získať bezplatne.

JavaScript je jediný skript na strane klienta, ktorý je podporovaný všetkými analyzovanými prehliadačmi.

## 3 Špecifikácia

### 3.1 Kontext systému

Obrázok 1 zobrazuje kontext systému – jeho okolie a ľudí, ktorý s ním prichádzajú do kontaktu. Systém je znázornený obdĺžnikom uprostred, okolné entity oválmi, vstupy a výstupy šípkami.



Obrázok 1: Kontext systému

Kontext vyplýva v mnohom zo zadania – máme tu typických používateľov systému (**účastník**, **oponent**, **anonymný používateľ**, **administrátor** a **autor** príspevku), ako aj údaje, ktoré jasne vyplývajú z požadovaných funkcií systému. Ak má viesť systém prijímať príspevky od jednotlivých autorov, objaví sa tu vstup nazvaný **príspevok**. Podobne od účastníka konferencie sa prijíma **prihláška**. Oponent zadáva svoje **preferencie**, čo sa týka kľúčových slov (tie popisujú jednotlivé témy) a zasiela **posudok** na pridelený príspevok. Anonymný používateľ sa dostane iba k **všeobecným informáciám**. Napokon administrátor systému – čo je vlastne predseda komisie – má rozhodujúce slovo pri **pridelení oponentov** na príspevky a vykonáva **konečné rozhodnutia** vo väčšine riešených otázok.

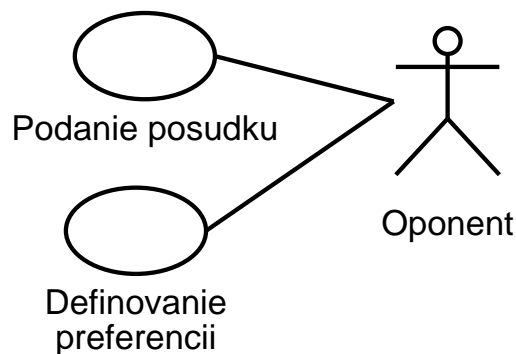
Funkcie, cez ktoré jednotliví používatelia prístupujú k systému, sú popísané v nasledujúcej časti, potom nasleduje popis modelu údajov a napokon sú špecifikovaní používatelia – tým by sa mali vyjasniť pojmy, ktoré sú v kontextovom diagrame použité.

### 3.2 Funkcie systému

#### 3.2.1 Funkcie systému z pohľadu oponenta

Systém bude realizovať rozhranie, pomocou ktorého sa budú môcť oponenti:

- Prihlásiť a definovať svoje preferencie. Preferencie oponenta vyjadrujú znalosti v danej

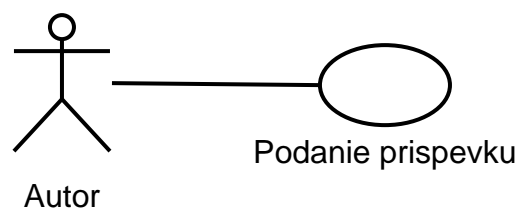


Obrázok 2: Akcie oponenta

oblasti, ktorá je definovaná kľúčovými slovami. Tieto preferencie bude možné vybrať z preddefinovanej množiny tém.

- Podanie vlastného posudku na pridelený článok. Pri tejto akcii oponent zároveň vyplní hodnotiace kritéria, ktoré budú stanovené pri konfigurácii systému.

### 3.2.2 Funkcie systému z pohľadu autora

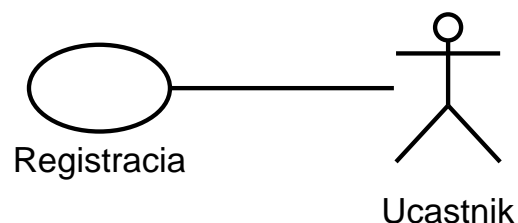


Obrázok 3: Akcie autora

Rozhranie v prípade prístupu autora bude umožňovať:

- Podanie príspevku do systému. Pri podávaní sa zároveň zvolí téma príspevku – bude možné vybrať si z definovaných kľúčových slov, ktoré vhodne popisujú obsah príspevku.

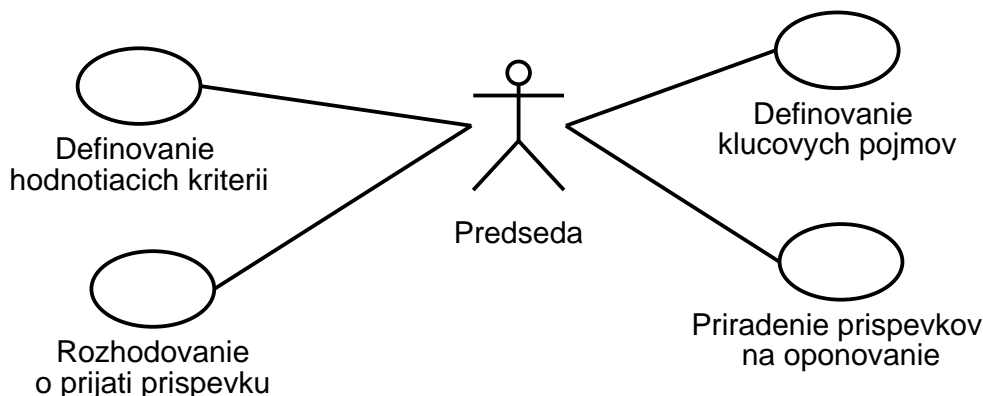
### 3.2.3 Funkcie systému z pohľadu účastníka



Obrázok 4: Akcie účastníka

Systém bude poskytovať možnosť podávania prihlášky na konferenciu prostredníctvom internetu. Toto bude umožnené vytvorením rozhrania pre vyplnenie registrácie účastníka konferencie.

### 3.2.4 Funkcie systému z pohľadu predsedu



Obrázok 5: Akcie predsedu

Funkcie systému pre predsedu konferencie budú umožňovať:

- Definovanie kľúčových pojmov konferencie. Pri tejto akcii bude definovať kľúčové slová, ktoré charakterizujú jednotlivé tématické okruhy konferencie.
- Definovanie hodnotiacich kritérií umožňuje definovať kľúčové slová, ktoré budú vhodné pomenovávať jednotlivé hodnotiace kritéria.
- Priradenie príspevkov na oponovanie. Pri tomto úkone predseda bude môcť priradiť prijaté články prihláseným oponentom, pričom mu môže pomôcť systém na základe definovaných tém článku a preferencií oponenta.
- Rozhodovanie o prijatí príspevku. V tejto akcii bude predseda rozhodovať o sporných článkoch, ktoré neboli automaticky prijaté, prípadne zamietnuté.

## 3.3 Model údajov

### 3.3.1 Popis jednotlivých dátových entít:

**Clovek** abstraktná dátová entita, ktorá bude slúžiť iba ako entita, ktorá definuje základné atribúty pre identifikáciu používateľa systému.

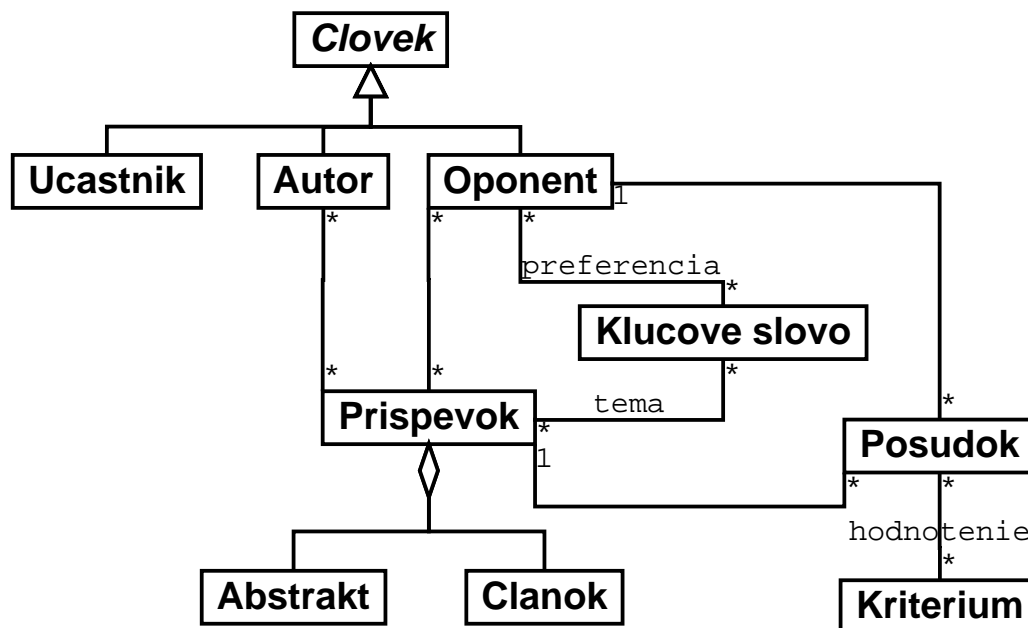
**Ucastnik** rozširujúca entita entity *Clovek*. Táto entita je iba obohatená o niektoré atribúty nutné pre evidenciu účastníkov konferencie.

**Autor** dátová entita rozširujúca entitu *Clovek*. Bude obsahovať údaje o autoroch článkov.

**Oponent** entita rozširujúca entitu *Clovek*, podobne ako *Autor*. Jej atribúty budú presnejšie definovať oponenta.

**Klucove slovo** ide o dátovú entitu, ktorá pomenováva tématické okruhy konferencie a zároveň preferencie oponentov. Tieto dve navzájom rôzne entity bolo nutné vsunúť do jednej, kvôli automatickému priradeniu príspevkov k oponentom.

**Prispevok** dátová entita, ktorá obsahuje príspevok autora, pre systém bude viditeľný aspoň abstrakt a samotný text článku.



Obrázok 6: Model údajov v systéme vedeckej konferencie

**Posudok** dátová entita obsahujúca text posudku na daný článok.

**Kriterium** dátová entita obsahujúca textový popis kritéria, ktoré by mal spĺňať príspevok.

### 3.3.2 Popis relácií:

**Autor – Prispevok** Definuje vzťah, že autor napísal niekoľko príspevkov, prípadne ešte ani jeden. V opačnom smere vyjadruje táto relácia možnosť napísania príspevku aspoň jedným autorom, prípadne kolektívom autorov.

**Prispevok – klucove slovo: tema** Definuje priradenie jednému príspevku niekoľko možných tématických okruhov – kľúčových slov. Pre príspevok nie je nutné definovať tému (priradiť mu kľúčové slová), no je to vhodné. Jedno kľúčové slovo môže byť použité na identifikáciu niekoľkých príspevkov, z čoho vyplýva možnosť mať v systéme niekoľko článkov na tú istú tému. Samozrejme nie je nutné priradiť príspevkom všetky kľúčové slová.

**Prispevok – oponent** Definuje vzťah medzi oponentom a zaslaným príspevkom. Vyjadruje priradenie príspevku aspoň jednému, prípadne viacerým oponentom. Oponent nemusí mať priradený ani jeden príspevok.

**Oponent – klucovne slovo: preferencie** Vyjadruje vzťah medzi oponentom a jeho preferenciami. Svoje preferencie si oponent zvolí podľa toho, aké tématické okruhy ovláda a je schopný ich oponovať.

**Oponent – posudok** Popisuje, že daný oponent napísal posudok. Je možné, aby oponent napísal niekoľko rôznych posudkov, prípadne žiaden. Každý posudok má definovaného iba jedného oponenta, nemôže existovať jeden a ten istý posudok od viacerých oponentov.



**Posudok – príspevok** Opisuje vzťah medzi posudkom a príspevkom. Jeden príspevok môže mať niekoľko rôznych posudkov, prípadne ani jeden. Posudok môže byť priradený nanajvýš jednému príspevku.

**Posudok – kritérium: hodnotenie** Táto relácia vyjadruje vzťah medzi posudkom a dátovou entitou kritéria – ide vlastne o hodnotenie – pričom táto relácia definuje aj výšku samotného hodnotenia. Nie je striktné povedané, že je nutné používať hodnotenie – vtedy posudok nebude mať priradené ani jedno kritérium. V opačnom prípade je možné priradiť všetky kritéria k posudku.

### 3.4 Používatelia systému

Systém Con4U budú cez webovské rozhranie používať používatelia, ktorý budú vystupovať v rôznych rolách. V ďalších častiach budeme opisovať možné roly používateľov tohto systému.

Každý používateľ (okrem anonymného používateľa) bude mať svoje prihlasovacie údaje a pre prácu so systémom sa bude pomocou nich autentifikovať. Jedna osoba môže disponovať viacerými rolami – ak bude nejaký používateľ súčasne administrátorom aj oponentom, bude mať priradené len jedno prihlasovacie meno a heslo.

#### 3.4.1 Anonymný používateľ

Anonymný používateľ má možnosť si prezerať len statické web-stránky o konferencii, ktoré v skutočnosti ani nebudú pod správou Con4U. Tieto môžu zahŕňať informácie o samotnej konferencii (program, niektorí účastníci, sponzori, organizátori a podobne) a aj informácie konkrétnejšieho charakteru, ktoré môžu mať podobný dizajn ako stránky spravované Con4U systémom a môžu s nimi byť aj hypertextovo zviazané, ale nie sú ním generované.

Okrem toho môže byť systém nastavený tak, že si anonymní používatelia môžu prezerať aj niektoré informácie o prijatých článkoch – napríklad mená autorov a názvy článkov, alebo aj abstrakty. Táto možnosť môže byť zapnutá pri nastavovaní konfiguračných parametrov systému administrátorom.

O anonymných používateľoch systém nebude evidovať žiadne informácie. Anonymní používatelia nemôžu meniť žiadne dáta v systéme a nemôžu si prezerať žiadne iné dáta v systéme ako tie, ktoré boli vopred povolené administrátorom.

#### 3.4.2 Účastník konferencie

Účastník konferencie zadáva osobné údaje prípadne iné údaje do formulára. Tieto údaje použijú organizátori pri plánovaní organizačných detailov konferencie. Niektoré položky (napríklad týkajúce sa platenia poplatkov za účasť) sa vzťahujú síce k účastníkovi konferencie, vyplňať ich však bude administrátor.

O účastníkoch konferencie bude systém evidovať údaje (ako je popísané v 3.3). Časť týchto údajov bude do systému vkladať sám účastník, časť bude zapisovať administrátor. Účastník si môže prezerať a modifikovať len údaje o svojej prihláške na konferenciu – žiadne iné.

#### 3.4.3 Autor príspevku

Autor príspevku do vedeckej konferencie bude pri podávaní článku zadávať svoje osobné údaje, údaje o článku a článok samotný. Tento článok bude posudzovaný oponentami a prijatý alebo neprijatý na konferenciu. Ako údaje o článku bude autor vyznačovať kategórie – kľúčové

slová, ktorých sa jeho článok týka. Zoznam týchto kľúčových slov bude konfigurovateľný a bude ich zadávať administrátor.

Autor si môže prezerat' a modifikovat' okrem svojich osobných údajov ešte údaje o svojich článkoch. Nebude môcť prezerat' žiadne iné údaje okrem tých, ktoré si bude prezerat' ako anonymný používateľ.

#### 3.4.4 Oponent – člen programovej komisie

Člen programovej komisie si na začiatku práce so systémom vyberie z preddefinovaných kategórií (kľúčových slov) svoje preferencie. Podľa týchto preferencií mu bude systém prideľovať príspevky na oponovanie. Toto prideľovanie príspevkov na oponovanie bude vykonávať administrátor, Con4U mu v tom však bude radiť vytváraním zoznamov možných oponentov podľa poradia najväčšej zhody oponentových preferencií a kľúčových slov článku.

Na samotné recenzovanie pridelených článkov bude mať možnosť si všetky tieto články zo systému stiahnuť. Po ich preštudovaní oponent vyplní formulár posudku s preddefinovanými otázkami, týkajúcimi sa rôznych kritérií kvality článku, napr. originalita, správnosť, forma, zhoda s predmetom konferencie a podobne. Nakoniec určí mieru svojho odporúčania/neodporúčania prijatia tohoto článku do programu konferencie.

Oponent nebude poznať ostatných oponentov priradených k článku a nebude vidieť ani hodnotenie článku ostatnými oponentami. Bude si môcť prezerat', prípadne modifikovat' len svoje vlastné posudky.

#### 3.4.5 Administrátor – predseda programovej komisie

Administrátor na začiatku nakonfiguruje systém pre danú konferenciu – určí možné kľúčové slová (oblasti) článkov, nastaví možnosti prezerania dát pre anonymných používateľov, priradí používateľov do rolí členov programovej komisie.

Počas etapy prijímania článkov a preferencií oponentov bude sledovať pridávané články, zjavne nevhodné príspevky (žarty, omyly, viacnásobne sa opakujúce články a podobne) bude rušiť.

Ďalšou úlohou predsedu programovej komisie je prideľovanie článkov jednotlivým oponentom na posudzovanie. Túto činnosť mu bude Con4U uľahčovať generovaním zoznamov vhodných oponentov k článkom (prípadne aj zoznamov vhodných článkov k oponentom) podľa miery zhody kľúčových slov článku a preferencií oponentov.

Počas bežiacej etapy posudzovania článkov bude sledovať progres jednotlivých oponentov pri posudzovaní vo forme tabuliek s počtom ukončených a chýbajúcich posudkov a jeho úlohou bude pripomenúť zaostávajúcim oponentom, že by mali dôslednejšie plniť svoju úlohu.

Po skončení etapy posudzovania bude mať predseda spolu s členmi programovej komisie k dispozícii generované poradia úspešností článkov podľa rôznych kritérií. Ich spoločnou úlohou bude určiť budú prijaté, a ktoré zamietnuté.

Po tom, ako sa programová komisia dohodne na zozname prijatých článkov, predseda tieto články vyznačí v systéme Con4U a systém to oznámi mailami autorom článkov. Generovaním zoznamov prijatých článkov sa úloha administrátora a aj celého systému Con4U končí.

## 4 Návrh

### 4.1 Hrubý návrh dátového modelu

Hrubý návrh dátového modelu je oproti modelu z kapitoly 3.3 obohatený o významné atribúty jednotlivých dátových entít. Zároveň boli rozbité niektoré asociačné väzby, nakoľko aj tie obsahujú isté atribúty potrebné pre presnejší opis vzťahu. Zároveň sa tým eliminovali nežiadúce vzťahy M ku N. Vzťahy M ku N, ktoré nenesú v asociácii žiadne atribúty okrem cudzích kľúčov, boli ponechané bez zmeny.

#### 4.1.1 Popis atribútov neväzobných dátových entít

Dátové entity ktoré boli opísané vo fáze analýzy a špecifikácie, sa v tejto časti už opisovať nebudú a dátové entity, ktoré vznikli až vo fáze hrubého návrhu, budú stručne opísané, aj keď budú mať rovnaký význam ako nimi nahradzované asociácie.

##### Clovek

**meno, priezvisko** skutočné meno osoby, ktorá chce používať systém ako účastník, autor, prípadne oponent;

**e-mail** kontaktnú e-mailovú adresu danej osoby;

**heslo** prístupové heslo do systému.

##### Prispevok

**abstrakt** abstrakt príspevku – ten je nutný vyplniť extra, aby mohol systém vytvárať program konferencie;

**clanok** celý článok autora;

**dátum prijatia** dátum skutočného prijatia príspevku;

**stav** atribút identifikujúci stav príspevku (prijatý na konferenciu alebo zamietnutý, pripravená alebo nepripravená prezentácia, odprezentovaný príspevok, atď...).

##### Klucove slovo

**meno** jednoduchý názov kľúčového slova;

**popis** presnejší opis významu kľúčového slova.

##### Posudok

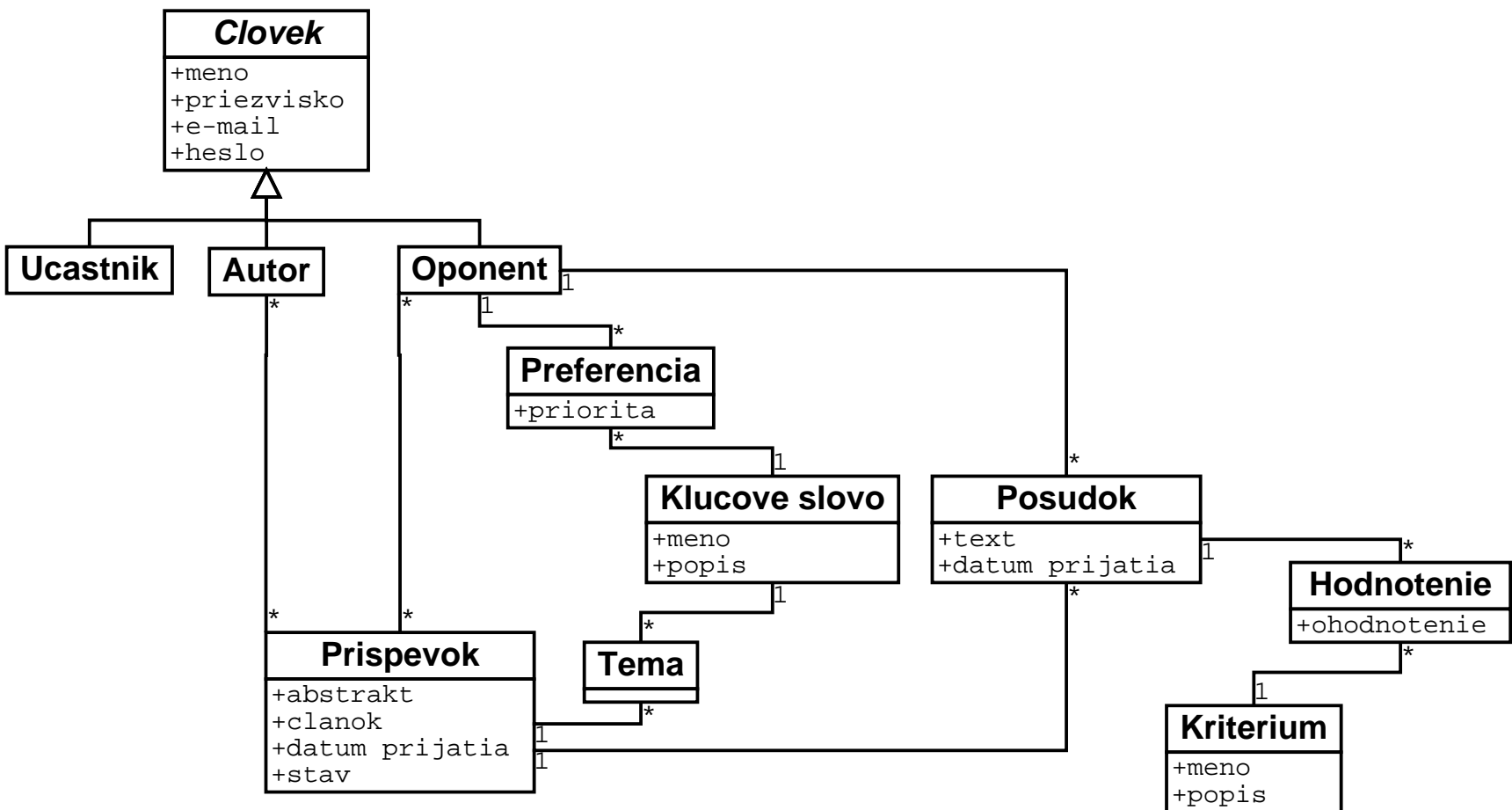
**text** obsah posudku, ktorý napísal oponent na článok;

**dátum prijatia** dátum skutočného prijatia posudku.

##### Kriterium

**meno** jednoduchý názov kritéria;

**popis** presnejší opis významu daného kritéria.



Obrázok 7: Hrubý návrh dátového modelu

### 4.1.2 Popis atribútov väzobných dátových entít

**Preferencia** dátová entita vyjadrujúca priradenia jednotlivých kľúčových slov – preferencií jednotlivým oponentom.

**priorita** určuje číselne vyjadrenie znalosti danej preferencie oponenta.<sup>9</sup>

**Tema** dátová entita určujúca vzťah medzi článkom a kľúčovým slovom – témou. Každý článok spadá pod jednu alebo niekoľko tém konferencie, autor nemusí túto väzobnú entitu nutne vytvárať (cez web-rozhranie aplikácie). Toto bude mať za následok, že programová logika nebude vedieť automaticky priradiť článok konkrétnemu oponentovi podľa preferencií.

**Hodnotenie** dátová entita určujúca vzťah medzi posudkom a jednotlivými hodnotiacimi kritériami, zadaných predsedom komisie.

**ohodnotenie** číselne vyjadrenie miery splnenia daného kritéria, na ktoré odkazuje hodnotenie.

## 4.2 Architektúra systému

Systém Con4U je vystavaný na architektúre klient-server, pričom väčšina funkčnosti je viazaná na jednotlivé HTTP požiadavky. Web-rozhranie bude ťažiskom celého systému – to spolu s použitím jazyka PHP určuje architektúru. Jednotlivé funkcie systému budú spúšťané požiadavkami od klientov, vykoná ich PHP modul bežiaci v adresnom priestore HTTP servera (v našom prípade je to Apache), pričom priamo z tohto procesu sa bude pristupovať na databázu. Je to technika vyskúšaná na množstve projektov obdobného typu – a nakoľko náš systém nepatrí do kategórie rozsiahlych projektov, je to dostatočné riešenie.

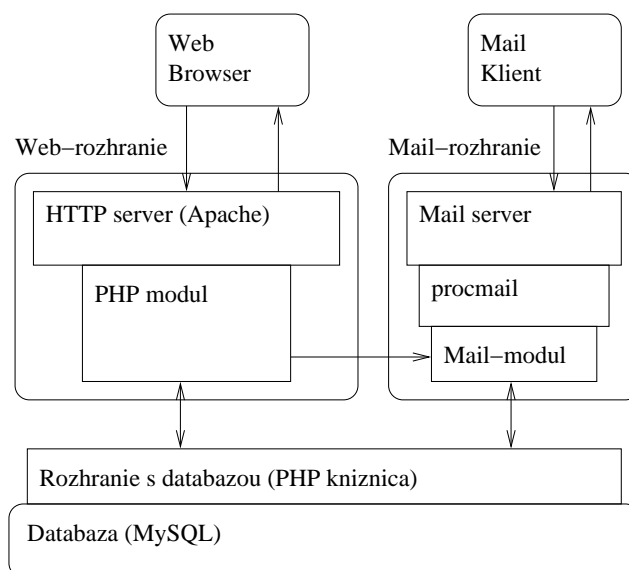
Systém obsahuje aj poštový modul, lebo pre mnohých používateľov môže byť mail prijateľnejšou formou pre odosielanie príspevkov, ale aj pre registráciu. Rovnako tak bude možné poslať e-mailom potvrdenie o zaplatení účastníkovi (ak si to bude želať), alebo informovať autora o prijatí (resp. neprijatí) jeho článku. Aj tu sú všetky činnosti riadené udalosťami.

Architektúra je znázornená na obrázku 8. Ako je vidieť, používateľ môže pristúpiť k systému buď pomocou prehliadača WWW, alebo využije mail klienta. Kým cez web-rozhranie sú prístupné všetky funkcie systému, funkcie mail-rozhrania budú zrejme obmedzenejšie. Hlavným obmedzením je, že funkcie administrácie sa cez toto rozhranie nedajú vykonávať. Toto je ale pochopiteľné obmedzenie – administrátor (predseda komisie, napr.) má obvykle systém prístupný po lokálnej sieti a nemal by mať žiadny technický problém pri využívaní web-rozhrania. Druhým dôvodom je potom aj bezpečnosť. Napokon – odozva mail-rozhrania nie je v intenciách toho, čomu sa hovorí komunikácia v reálnom čase – a tá je zrejme pri nastavovaní parametrov konferencie prakticky nutná. Administrátor potrebuje hneď vidieť, či je všetko nastavené tak, ako to vyžaduje.

Popíšme si teraz v krátkosti jednotlivé znázornené celky a komunikáciu medzi nimi.<sup>10</sup> Web browser (jeho používateľ) zasiela vo všeobecnosti žiadosť o zobrazenie nejakej stránky, na ktorej môže byť aj formulár. Po jeho vyplnení je tento spracovaný modulom web-rozhrania a údaje, ktoré majú byť perzistentné, sa uložia do databázy. Veľmi podobná je komunikácia

<sup>9</sup>V systéme sa bude používať automatické priraďovanie číselných hodnôt na základe zotriedeného vstupu preferencií od oponenta.

<sup>10</sup>Podrobnejší opis nasleduje v časti 4.3.



Obrázok 8: Architektúra systému

mailom – používateľ napríklad dostane predpripravený formulár, ktorý vyplní a odošle. Náš systém všetko spracuje a opäť uloží do databázy. Odpoveďami sú buď ďalšie web-stránky, alebo maily. Je možné si povšimnúť, že PHP modul v HTTP serveri komunikuje s Mail-modulom. K tomu dochádza, ak užívateľ cez web vykoná akciu, ktorá je potvrdzovaná mailom<sup>11</sup>, alebo napríklad cez web požiada o zaslanie mailového formulára. Opačná komunikácia nemá samozrejme zmysel.

PHP modul vo web-interface sa zaoberá len web-rozhraním, mail-modul zase iba mail-rozhraním. Spoločné funkcie pracujúce s databázou budú v jednej knižnici, ktorá bude využívaná oboma modulmi. V tejto by mali byť ukryté všetky potrebné funkcie pre databázu – knižnica by mala obaľovať všetky SQL príkazy.

### 4.3 Moduly systému

Ako bolo naznačené v časti o architektúre, systém má tri hlavné celky:

- modul **Web-rozhranie**;
- modul **Mail-rozhranie**;
- modul **Databázové rozhranie**.

#### 4.3.1 Web-rozhranie

Web-rozhranie zabezpečuje komunikáciu používateľa so systémom pomocou prehliadača WWW. Technológie, ktoré použijeme v tomto module, už boli spomenuté: HTTP server **Apache** a modul **PHP**. PHP stránky budú zabezpečovať interaktívnu komunikáciu s používateľom. Implementácia bude realizovaná ako vo forme PHP stránok, tak aj vo forme knižnice komponentov.

<sup>11</sup>Nemusí ísť nutne o toho istého používateľa. Napríklad, keď predseda v role administrátora prijme príspevok, autorovi príspevku sa pošle e-mailom potvrdenie (ak si tak zvolil).

Knižnica komponentov bude vo forme jednoduchých funkcií implementovať väčšinu vizuálnych prvkov stránky – ide najmä o prvky formulára. Identifikovali sme napríklad potrebu častého pohľadu na obsah tabuľky z databázy vo forme tabuľky v HTML. Tento pohľad je vo väčšine prípadov veľmi podobný, takže vytvoríme komponentu, ktorá na požiadanie vráti hotový kód HTML – ten potom už len vložíme do stránky.<sup>12</sup>

Bezpečnosť bude realizovaná technikou cookies, pričom mechanizmy pre prácu s reláciou si zrejme vytvoríme všeobecne tak, aby sme nemuseli na každej PHP stránke explicitne pracovať s cookies (opäť to budú funkcie v knižnici).

Komunikácia s databázou je realizovaná výhradne cez modul Databázové rozhranie.

### 4.3.2 Mail-rozhranie

Pre tento modul je potrebné zvoliť vhodnú technológiu umožňujúcu automatické spracovanie mailov. Jednoduchým a súčasne dostatočne vhodným nástrojom je **procmail**, ktorý umožňuje jednoduchým spôsobom poslať obsah mailu zvolenému skriptu. Ak sa ukáže, že PHP je vhodný na úlohy, ktoré budú mať skripty vykonávať, zvolíme tento jazyk, aby sme použili jednotný jazyk vo všetkých moduloch. Ako prakticky istou alternatívou je potom Perl. Rozhodnutia o implementačnom prostredí pre mail-modul však vykonáme neskôr.

Spomínané skripty zabezpečujú odosielanie mailov na požiadanie web-rozhrania, spracovanie príšlých mailov, ukladanie všetkých potrebných informácií do databázy, ako aj ich vyberanie pre prípad generovaných a odosielaných mailov. Komunikácia s databázou je samozrejme výhradne cez modul rozhrania s databázou.

Tento modul vyžaduje SMTP server a procmail – oboje nainštalované na cieľovom prostredí.

### 4.3.3 Databázové rozhranie

Myšlienka databázového rozhrania je veľmi jednoduchá – zabaliť všetky príkazy v jazyku SQL do funkcií. Volanie funkcií bude v kóde, ktorý toto rozhranie využíva, vyzeráť lepšie, bude kratšie – a potenciálne umožňuje zmeniť databázu bez zmeny vyšších modulov.<sup>13</sup> Všetky prvky jazyka SQL sa týmto izolujú do jedného kompaktného modulu. Môžeme vytvoriť elegantné funkcie na prácu s asociačnými tabuľkami, ktoré popisujú vzťahy M ku N. Využívanie modulu by malo výrazne zprehľadniť kód v moduloch rozhraní.

Modul neslúži na ukrývanie tabuliek, atribútov, a podobne – súvisí to s tým, že mapovanie dátového modelu do tabuliek je veľmi priamočiare, takže ho netreba zapúzdrovať do iného modelu.

## 4.4 Návrh formulárov pre web-rozhranie

Na web stránke konferencie je možné rozlíšiť niekoľko typov informácií. Ide o statické stránky, ktoré informujú o konferencii, pričom ich paleta sa mení konferencia od konferencie.

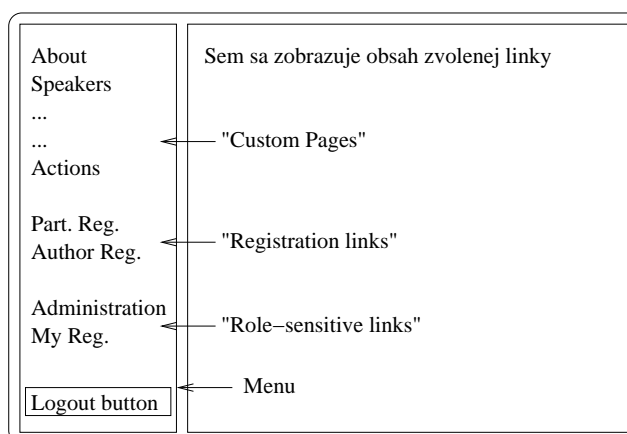
---

<sup>12</sup>Ak ostaneme u tohto príkladu, komponenta *tableView* by ako parametre prijímala *meno tabuľky*, ďalej *zoznam stĺpcov*, ktoré budú (v zvolenom poradí) zobrazené, a napokon informáciu o *štýle* vykreslenia tabuľky. Výsledkom bude spomínaný HTML kód tabuľky.

Štýl vykresľovania môže napríklad hovoriť o tom, či bude tabuľka len na prezeranie, alebo bude možné s položkami vykonávať isté operácie – napríklad v každom riadku sa vykreslí checkbox a dole tlačítko so zvoleným textom. Čo sa vykoná, na tom tejto komponente nemusí záležať, ona len vykreslí tabuľku a doplní ju prvkami formulára, ak to bude požadované.

<sup>13</sup>Je známe, že hoci je SQL do istej miery štandard, rôzne databázy majú rôzne pomenovania typov napríklad pre binárne dáta – týmto sa tieto problémy elegantne obchádzajú.

Ďalej ide o linky k niekoľkým štandardným formulárom – *Registrácia účastníka* (**Participant Registration**), *Registrácia autora* (**Author Registration**). Nasledujú linky, ktoré by mali byť prístupné podľa jednotlivých rolí: pre administrátora linka *Administrácia* (**Administration**), pre oponenta *Moje posudky* (**My Reviews**), pre autora *Moje príspevky* (**My Papers**), a napokon každý bude mať linku *Moje registračné informácie* (**My Registration**). Všetky linky budú v konečnej verzii v anglickej podobe, a naďalej budeme používať len anglický tvar. Samozrejme je možné samotný text linky zmeniť. Všetky dôležité linky budú vložené do ľavého menu, ako je znázornené na obrázku 9.



Obrázok 9: Návrh obrazovky web-rozhrania

Podobne ako iné konferenčné systémy, aj my sme zvolili model, kedy je menu vľavo a obsah sa premieta do pravej hlavnej časti obrazovky. Ako je vidieť na obrázku, menu obsahuje niekoľko liniek, pričom sa začína stránkami, ktorých povaha je informačná, potom sú linky pre registráciu, a nakoniec nasledujú linky závislé na roli prihláseného používateľa. Tlačítkom *Logout* sa používateľ odhlasuje. V prípade, že používateľ nie je prihlásený, miesto poslednej skupiny je na obrazovke ponúknutá možnosť prihlásenia (vstupné riadky *Login name* a *Password* a tlačítko *Login*).

V systéme Con4U je možné konfigurovať všetky linky na informačné stránky. Tieto stránky sú nazvané termínom *Custom pages*. V administrácii je možnosť nastaviť mená liniek, ich poradie, ako aj natiahnúť obsah stránky do systému. Je dokonca možné nastaviť viditeľnosť stránky pre jednotlivé role, pričom toto nastavenie nie je možné jednoducho obísť, lebo všetky custom-stránky sú prístupné cez jednu PHP stránku, ktorá kontroluje prístup ku custom-stránkam.

V nasledujúcich kapitolách si popíšeme funkcie formulárov prístupných cez jednotlivé uvedené linky. Každý formulár (označený ako typ FORM) bude krátko opísaný a budú vymenované prvky, ktoré sa na ňom nachádzajú. Pre každý prvok bude uvedený typ komponenty, ktorou bude na formulári predstavovať daný prvok. Možné komponenty sú: TEXT, CHECKBOX, SELECT, RADIO, PASSWORD, TEXTAREA – teda typické prvky formulárov jazyka HTML. BUTTON je tlačítko, ktorého akcia bude popísaná priamo na mieste. Komponenta LIST je zoznam hodnôt, z ktorých je možné jednu vybrať.

Navyše k týmto budeme používať komponentu TABLE, ktorá bude veľmi podobná PHP komponente *tableView* spomínanej ako typický príklad PHP komponenty z modulu web-rozhrania. Pre tabuľku budú vymenované jej stĺpce, ktoré budú označené typom ATTRIBUTE. Okrem toho využijeme aj atribút *Select* (typ ATTRIBUTE-CHECKBOX), ktorý slúži na označovanie údajov tabuľky, aby s nimi bolo možné realizovať akcie (napríklad mazať označených



používateľov).

LINK je linka na iný udaný formulár, pričom môže byť podľa potreby kombinovaná s iným atribútom (BUTTON-LINK je tlačítko na iný formulár, ATTRIBUTE-LINK je atribút tabuľky, ktorý je súčasne linkou, ...).

#### 4.4.1 Štandardné formuláre

**Confirmation (FORM)** Formulár určený na potvrdzovanie akcií, ktoré si potvrdenie vyžadujú (napríklad mazanie používateľov). Obe tlačítka na formulári slúžia ako LINK na formulár, z ktorého bola potvrdzovaná akcia vyvolaná – ak nebude uvedené inak.

**Message (TEXT)** Popis potvrdzovanej udalosti.

**OK (BUTTON)** Potvrdenie akcie.

**Cancel (BUTTON)** Odmietnutie akcie.

#### 4.4.2 Registration of Authors

Po zvolení tejto linky na hlavnej stránke sa v pravej časti obrazovky sprístupní formulár Register.

**RegisterAuthor** Tento formulár slúži na zadanie registračných informácií o používateľovi, ktorý ich potvrdením vstupuje do role *Author*. Z tohto formulára je možné zadať údaje potrebné na vyplnenie atribútov v tabuľke *tblPerson* – *Login*, *Password*, *FirstName*, *MiddleName*, *Surname*, *Titles*, *Email*, *Organization*.

**Login (TEXT)** Textové pole umožňujúce zaregistrovanie prihlasovacieho mena.

**Password (TEXT)** Textové pole umožňujúce zadať heslo.

**Retype Password (TEXT)** Textové pole na potvrdenie zadaného hesla.

**First Name (TEXT)** Textové pole umožňujúce zadanie mena.

**Middle Name (TEXT)** Textové pole umožňujúce vstup mena.

**Surname (TEXT)** Textové pole umožňujúce zadanie priezviska.

**Titles (TEXT)** Textové pole umožňujúce zadať tituly používateľa.

**Email (TEXT)** Textové pole umožňujúce vstup emailovej adresy.

**Organization (TEXT)** Textové pole umožňujúce evidovať príslušnosť k nejakej organizácii.

**Notify (CHECKBOX)** Ide o prepínač, ktorým používateľ požiadava od systému správu v podobe mailu v prípade prijatia článku.

**OK (BUTTON-LINK: *Login*)** Tlačítko na zapísanie zmien do systému.

#### 4.4.3 Registration of Participant

Po zvolení tejto linky na hlavnej stránke sa v pravej časti obrazovky sprístupní formulár Register, ktorý je rozšírením formuláru pre autorou o položky, ktoré sa týkajú platieb za konferenciu, apod.

**RegisterParticipant** Tento formulár slúži na zadanie registračných informácií o používateľovi, ktorý ich potvrdením vstupuje do role *Participant*. Z tohto formulára je možné zadať najdôležitejšie údaje potrebné na vyplnenie atribútov z tabuľky *tblPerson* – *Login*, *Password*, *FirstName*, *MiddleName*, *Surname*, *Titles*, *Email*, *Organization* – ako aj údaje, ktoré sa týkajú len účastníka (tabuľka *tblParticipant*) – *Address*, *Nationality*, *Fee*, *FeePaid*, *Phone* a *RegState*.

**Login** (TEXT) Textové pole umožňujúce zaregistrovanie prihlasovacieho mena.

**Password** (TEXT) Textové pole umožňujúce zadať heslo.

**Retype Password** (TEXT) Textové pole na potvrdenie zadaného hesla.

**First Name** (TEXT) Textové pole umožňujúce zadanie mena.

**Middle Name** (TEXT) Textové pole umožňujúce vstup mena.

**Surname** (TEXT) Textové pole umožňujúce zadanie priezviska.

**Titles** (TEXT) Textové pole umožňujúce zadať tituly používateľa.

**Email** (TEXT) Textové pole umožňujúce vstup emailovej adresy.

**Organization** (TEXT) Textové pole umožňujúce evidovať príslušnosť k nejakej organizácii.

**Notify** (CHECKBOX) Ide o prepínač, ktorým používateľ požiada od systému správu v podobe mailu, ktorá potvrdzuje zaplatenie poplatku za konferenciu.

**Address** (TEXTAREA) Adresa účastníka.

**Nationality** (TEXT) Národnosť účastníka.

**Fee** (SELECT) Výber ceny z ponúkaných možností (pre rôzne organizácie môžu byť rôzne cenové ponuky).

**Phone** (TEXT) Telefónne číslo na účastníka.

**State** (SELECT) Výber štátu účastníka.

**OK** (BUTTON-LINK: *Login*) Tlačítko na zapísanie zmien do systému.

#### 4.4.4 My Registration

Po zvolení tejto linky sa v pravej časti obrazovky sprístupní formulár *MyRegistration*. Tento obsahuje všetky potrebné údaje z *tblParticipant* a ak je používateľ účastníkom, tak potom aj údaje z tabuľky *tblParticipant*.

**MyRegistration** Tento formulár slúži na zobrazenie a zmenu registračných informácií o používateľovi (role *Author*, *Reviewer*). Jeho obsah je závislý od uvedených rolí, ktoré používateľ v systéme má. Z tohto formulára je možné meniť najdôležitejšie atribúty z tabuľky *tblPerson* – *Login*, *Password*, *FirstName*, *MiddleName*, *Surname*, *Titles*, *Email*, *Organization* (a pre účastníka aj atribúty z tabuľky *tblParticipant*). Ak sa niektoré pole zobrazí len pre niektoré role, sú tieto uvedené za popisom v zátvorkách.

**Login** (TEXT) Textové pole umožňujúce zmenu prihlasovacieho mena.

**Password** (TEXT) Textové pole umožňujúce zadať nové heslo.

**Retype Password** (TEXT) Textové pole na potvrdenie zadaného hesla.

- First Name** (TEXT) Textové pole umožňujúce zmenu (opravu) mena.
- Middle Name** (TEXT) Textové pole umožňujúce zmenu (opravu) mena.
- Surname** (TEXT) Textové pole umožňujúce zmenu (opravu) priezviska.
- Titles** (TEXT) Textové pole umožňujúce zmeniť tituly používateľa.
- Email** (TEXT) Textové pole umožňujúce zmenu (opravu alebo aktualizáciu) emailovej adresy.
- Organization** (TEXT) Textové pole umožňujúce evidovať a meniť príslušnosť k nejakej organizácii.
- Notify** (CHECKBOX) Ide o prepínač, ktorým používateľ v roli *Author* požaduje, aby mu systém zaslal správu v podobe mailu v prípade prijatia článku. Pri roli *Participant* sa notifikácia týka zaslania potvrdenia o zaplatení – tj. pozvánka na konferenciu. (*Author, Participant*)
- All Topics** (LIST) Zoznam tém v systéme, z ktorých si *Reviewer* môže vybrať ním preferované. (*Reviewer*)
- Selected Topics** (LIST) Zoznam tém, ktorých týkajúce sa články *Reviewer* uprednostňuje. Medzi týmito dvomi zoznamami je možné prehadzovať položky pomocou dvoch tlačítok. (*Reviewer*)
- Address** (TEXTAREA) Adresa účastníka. (*Participant*)
- Nationality** (TEXT) Národnosť účastníka. (*Participant*)
- Fee** (SELECT) Výber ceny z ponúkaných možností (pre rôzne organizácie môžu byť rôzne cenové ponuky). (*Participant*)
- Fee Paid** (výpis dátumu) Informácia o dátume zaplatenia poplatku – nie je možné editovať. (*Participant*)
- Phone** (TEXT) Telefónne číslo na účastníka. (*Participant*)
- State** (SELECT) Výber štátu účastníka. (*Participant*)
- Change** (BUTTON-LINK: *MyRegistration*) Tlačítko na zapísanie zmien do systému.

#### 4.4.5 My Papers

Po zvolení linky My Papers sa v pravej časti obrazovky sprístupní formulár MyPapers.

**MyPapers** Tento formulár umožňuje používateľovi v roli *Author* prezerať zoznam článkov, ktoré zaslal do konferencie, ich stav (prijatý, neprijatý) a jednotlivé pripomienky, ktoré k článkom majú oponenti. Taktiež obsahuje odkaz na formulár pre pridanie článku do konferencie.

**Papers** (TABLE) Tabuľka obsahuje najdôležitejšie údaje o článku a priebehu jeho akceptácie na konferencii.

**Name** (ATTRIBUTE-LINK: *EditPaper*) Názov článku.

**State** (ATTRIBUTE) Informácia o stave, ktorá môže zahŕňať aj počet pridelených oponentov.

**Add** (BUTTON-LINK: *EditPaper*) Tlačítko na pridanie ďalšieho článku do konferencie.

**EditPaper** Tento formulár umožňuje používateľovi v roli *Author* pridávať nové články do konferencie, alebo editovať údaje v už zaslaných článkoch. Zadávané údaje zodpovedajú atribútom tabuľky `tblPerson`. Obsahuje tiež komponent na priradenie tém, ktoré súvisia s článkom.

**Name** (TEXT) Textové pole umožňujúce zadať názov článku, ktorým sa článok bude identifikovať pre používateľov konferencie.

**Authors** (TEXT) Textové pole umožňujúce zadať zoznam autorov článku.

**Body** (TEXT) Textové pole umožňujúce zadať názov súboru, ktorý obsahuje samotný článok a bude prenesený na stranu servera.

**Abstract** (TEXTAREA) Viacriadkové textové pole umožňujúce autorovi priamo zadať abstrakt článku.

**KeyWords** (TEXT) Textové pole umožňujúce zadať kľúčové slová, ktorých sa článok dotýka.

**All Topics** (LIST) Zoznam tém v systéme, z ktorých si *Author* môže vybrať tie, ktoré sa dotýkajú ním zasielaného článku.

**Selected Topics** (LIST) Zoznam tém, ktoré *Author* vyberie ako súvisiace s článkom.

**Comments** (odstavec textu pre každý komentár oponenta) Tu sa nachádza zoznam pripomienok od oponentov.

**OK** (BUTTON-LINK: *MyPapers*) Tlačítko na potvrdenie zmenených údajov a ich zápis do systému.

#### 4.4.6 My Reviews

Po zvolení linky *My Reviews* sa do pravej časti obrazovky zobrazí formulár *MyReviews*.

**MyReviews** (FORM) V tomto formulári je centrálnym prvkom tabuľka článkov pridelených oponentovi na posudzovanie. Je možné dať si zobrazíť len články, pre ktoré autor ešte posudok nezaslal, články, ktoré už posúdil alebo všetky články. Rovnako tak tu bude výrazne vypísaná informácia o termíne skončenia procesu posudzovania.

**To Review** (CHECKBOX) Budú zobrazené príspevky, ktoré má oponent ešte posúdiť. Na začiatku bude tento zapnutý.

**Reviewed** (CHECKBOX) Budú zobrazené príspevky, ktoré oponent už posudzoval. Na začiatku bude tento zapnutý.

**Papers** (TABLE) Tabuľka bude obsahovať v riadkoch záznamy o jednotlivých príspevkoch priradených oponentovi na posudzovanie:

**Name** (ATTRIBUTE) Meno príspevku.

**Review Date** (ATTRIBUTE) Ak ide o článok, ktorý už oponent posúdil, zobrazí sa dátum posudku.

**Recommandation** Hodnota položky Odporúčanie na prijatie.

**ReviewPaper** (FORM) Základom tohto formulára sú položky hodnotenia príspevku podľa kritérií zaradené v tabuľke. Na úvod formulára sú vypísané vybrané informácie o príspevku.

**Ranking** (TABLE) Tabuľka obsahuje hodnotiace atribúty postupne pre všetky kritériá.

**Criteria** (ATTRIBUTE) Meno kritéria.

**Value** (ATTRIBUTE-SELECT) Tu oponent vyberie hodnotu pre dané kritérium.

**Comment** (TEXTAREA) Pole pre komentár k tomuto hodnoteniu.

**Description** (ATTRIBUTE) Bližší opis kritéria.

**Recommendation** (SELECT) Z týchto možností oponent vyberie svoje celkové odporúčanie ohľadom prijatia či neprijatia tohto príspevku na konferenciu.

**Comment to author** (TEXT) do tohto textového poľa oponent napíše svoje odporúčania či bližšie poznámky pre autora posudzovaného príspevku.

**Comment to program comitee** (TEXT) toto textové pole slúži na komentáre oponenta pre členov programovej komisie týkajúce sa posudzovaného článku.

**Ok** (BUTTON) Potvrdenie zmien a uloženie formulára do databázy.

#### 4.4.7 Administration

Po zvolení linky *Administration* sa celé ľavé menu zmení a objavia sa v ňom nasledovné položky:

**Users** (LINK: *AdminUsers*) Administrácia používateľov.

**Review Assignment** (LINK: *AdminReviewAssignment*) Priradovanie oponentov k článkom.

**Custom Pages** (LINK: *AdminCustomPages*) Administrácia custom-stránok.

**Con4U Setup** (LINK: *AdminProperties*) Nastavovanie globálnych vlastností systému.

**Ranking Criterias** (LINK: *AdminRankingCriteria*) Administrácia hodnotiacich kritérií.

**Topics** (LINK: *AdminTopics*) Zadávanie tém konferencie.

**AdminUsers** (FORM) Tento formulár slúži na administráciu používateľov. Je možné vybrať si role, ktorých používatelia sa zobrazia. Ďalej je možné priamo z tohto formulára mazať viacerých používateľov. Pre tento účel bola použitá komponenta TABLE s checkboxom pre každú položku, pričom po stlačení tlačítka *Delete* sa označení používateľa odstráni zo systému (vyžaduje sa potvrdenie operácie).

**Roles** (CHECKBOX skupina) – skupina štyroch checkboxov pre roli *Administrator*, *Author*, *Reviewer* a *Participant*. V tabuľke sa zobrazujú len tí používatelia, ktorý patria aspoň do jednej zo zaškrtnutých rolí.

**Users** (TABLE) Tabuľka bude obsahovať vybrané najdôležitejšie atribúty z tabuľky tblPerson:

**Surname** (ATTRIBUTE)

**FirstName** (ATTRIBUTE)

**MiddleName** (ATTRIBUTE)

**Titles** (ATTRIBUTE)

**Organization** (ATTRIBUTE)

**Email** (ATTRIBUTE)

**Role** (ATTRIBUTE) Obsahuje vypísané role, ktorými používateľ disponuje.

**Select** (ATTRIBUTE-CHECKBOX)

**Login** (ATTRIBUTE-LINK: *AdminUserEdit*)

**Add** (BUTTON-LINK: *AdminUserEdit*) Tlačítko pre pridanie nového používateľa.

**Delete** (BUTTON-LINK: *Confirmation*) Vymazanie vybraných používateľov pomocou komponenty – na túto operáciu je požadované potvrdenie, ktoré nasleduje na samostatnom formulári. Po potvrdení sa obnoví formulár *AdminUsers*.

**AdminUserEdit** (FORM) Formulár AdminUserEdit slúži na editáciu už existujúceho záznamu, prípadne naplnenie vytvoreného záznamu (pridanie používateľa). Bude obsahovať jednotlivé položky záznamu z tabuľky tblParticipant:

**Login** (TEXT)

**Password** (TEXT)

**First Name** (TEXT)

**Middle Name** (TEXT)

**Surname** (TEXT)

**Titles** (TEXT)

**Email** (TEXT)

**Organization** (TEXT)

**Role** (CHECKBOX) Štvorica checkboxov pre každú rolu: *Administrator*, *Participant*, *Reviewer*, *Author*.

**Registration Date** (TEXT) Dátum registrácie.

**Add/Edit** (BUTTON-LINK: *AdminUsers*) Potvrdenie záznamu a znovu načítanie formuláru *AdminUsers*. Popis bude buď *Add* alebo *Edit*, čo závisí na akcii, ktorú formulár momentálne realizuje.

**Cancel** (BUTTON-LINK: *AdminUsers*) Znovu načítanie formuláru *AdminUsers* bez uloženia záznamu.

**AdminReviewAssignment** (FORM) Toto je úvodný formulár pre priradovanie oponentov na jednotlivé príspevky. Obsahuje prehľad príspevkov a počty oponentov pri každom príspevku. Na tomto formulári neprebíha samotné priradovanie – na to slúžia nasledovné dva formuláre (*AdminArticleReviews* a *AdminReviewArticles*).

**Papers** (TABLE) Tabuľka všetkých článkov evidovaných systémom bude obsahovať tieto atribúty:

**Article name** (ATTRIBUTE-LINK: *AdminReviewArticle*) Názov článku. Po kliknutí na článok sa dostaneme na formulár, kde priradíme oponentov k tomuto (jedinému) príspevku.

**Date of receiving** (ATTRIBUTE) Dátum prijatia článku do systému.

**Number assignment reviewers** (ATTRIBUTE) Počet priradených oponentov.

**Select** (ATTRIBUTE-CHECKBOX)

**Delete** (BUTTON-LINK: *Confirm*) Vymaže označené články.

**AdminArticleReviews** (FORM) Formulár slúžiaci na priradenie oponentov jednému príspevku.

**Reviews** (TABLE) Tabuľka bude obsahovať zoznam všetkých oponentov. Tabuľka bude zoradená podľa zhody preferencií oponenta s témami príspevku.

**Name** (ATTRIBUTE-LINK: *AdminReviewArticles*) Meno oponenta.

**Numbers of articles to be done** (ATTRIBUTE) Počet posudkov, ktoré má daný oponent priradené.

**Selected** (ATTRIBUTE) Obsahuje informáciu, či oponent už oponuje tento článok. Na začiatku práce s formulárom sa stav zhoduje s atribútom *Select*. Táto informácia je vhodná najmä po použití automatického návrhu (*Suggest Selection*), aby mal administrátor kontrolu aj nad pôvodným stavom priradenia.

**Select** (ATTRIBUTE-CHECKBOX) Na začiatku sú označení oponenti, ktorí oponujú daný príspevok. Tento stav je možné zmeniť (ako ručne, tak aj tlačítkom *Suggest Selection*).

**Suggest Selection** (BUTTON) Navrhne priradenie oponentov pre príspevok tak, aby ich počet sedel s nakonfigurovaným počtom oponentov pre článok. Pri návrhu sa postupuje podľa poradia v zozname, ale prihliada sa na množstvo príspevkov, ktoré už oponent má pridelené.

**Submit** (BUTTON-LINK: *Confirmation*) Potvrdenie zmien priradení.

**Reset** (BUTTON) Obnovenie pôvodného stavu formulára.

**AdminReviewArticles** (FORM) Úlohov tohto formulára je uľahčiť priradenie jednotlivých príspevkov oponentom.

**Articles** (TABLE)

**Name** (ATTRIBUTE-LINK: *AdminArticleReviews*) Názov príspevku.

**Autor** (ATTRIBUTE) Meno autora.

**Numbers of reviewers** (ATTRIBUTE) Počet oponentov priradených k tomuto príspevku.

**Selected** (ATTRIBUTE) Obsahuje informáciu, či je článok oponovaný týmto oponentom. Na začiatku práce s formulárom sa stav zhoduje s atribútom *Select*. Táto informácia je vhodná najmä po použití automatického návrhu (*Suggest Selection*), aby mal administrátor kontrolu aj nad pôvodným stavom priradenia.

**Select** (ATTRIBUTE-CHECKBOX) Na začiatku sú označené tie príspevky, ktoré daný oponent oponuje. Tento stav je možné zmeniť (ako ručne, tak aj tlačítkom *Suggest Selection*).

**Suggest Selection** (BUTTON) Navrhne priradenie oponentov pre príspevok tak, aby ich počet sedel s nakonfigurovaným počtom oponentov pre článok. Pri návrhu sa postupuje podľa poradia v zozname, ale prihliada sa na množstvo príspevkov, ktoré už oponent má pridelené.

**Submit** (BUTTON-LINK: *Confirmation*) Potvrdenie zmien priradení.

**Reset** (BUTTON) Obnovenie pôvodného stavu formulára.

**AdminCustomPages** (FORM) V tomto formulári je možné pridávať a odoberať statické stránky do/zo systému. Ďalej sa tu určuje viditeľnosť týchto stránok pre jednotlivé roly.

**Pages** (TABLE) Tabuľka bude obsahovať atribúty:

**Name** (ATTRIBUTE-LINK: *AdminCustomPageEdit*)

**MimeType** (ATTRIBUTE)

**Protection** (ATTRIBUTE)

**Roles** (ATTRIBUTE) Textový výpis rolí, ktoré môžu pristúpiť na túto stránku.

**Select** (ATTRIBUTE-CHECKBOX)

**Add** (BUTTON-LINK: *AdminCustomPageEdit*) Pridanie ďalšej statickej stránky.

**Delete** (BUTTON-LINK: *Confirmation*) Zmazanie vybraných stránok.

**AdminCustomPageEdit** (FORM)

**Name** (TEXT) Názov stránky (text linky, titul).

**MimeType** (COMBOBOX) – Mime typu objektu.

**Body** (BUTTON-LINK: upload stránky)

**Role Author** (ATTRIBUTE-CHECKBOX) Určuje, či stránka je prístupná autorom (prihláseným do systému).

**Role Participant** (ATTRIBUTE-CHECKBOX)

**Role Reviewer** (ATTRIBUTE-CHECKBOX)

**Submit** (BUTTON-LINK: *AdminCustomPages*) Uloženie zmien do databázy.

**Cancel** (BUTTON-LINK: *AdminCustomPages*) Znovu načítanie formuláru *AdminCustomPages* bez uloženia zmien.

**AdminProperties** (FORM) Tento formulár je určený na nastavenie všeobecných parametrov konferencie. Na formulári bude umiestnená tabuľka, kde prvý stĺpec bude označovať názov parametra a druhý stĺpec jeho hodnotu.

**Parameters** (TABLE) Tabuľka bude obsahovať atribúty z tabuľky *tblParameters*:

**ParameterName** (ATTRIBUTE) - Statický text.

**ParameterValue** (ATTRIBUTE-TEXT) - Editovateľný text.

Jednotlivé záznamy tabuľky nebude možné mazať a ani pridávať. Pojde o tabuľku kde bude možné editovať iba stĺpec s atribútom *parameterValue* (ATTRIBUTE). Jednotlivé parametre sú do systému pevne zavedené.

Ako príklad záznamov pre túto tabuľku sú:

*DeadlineForAutors* – dátum posledného možného poslania článku do konferencie.

*DeadlineForReviewers* – dátum posledného možného poslania posudku na článok do konferencie.

*NumberReviewersForPaper* – počet oponentov na článok.

*MaximumPapersForReviewer* – maximálny počet článkov na oponenta.



**Submit** (BUTTON-LINK: *Confirmation* (FORM)) Potvrdenie všetkých zmien hodnôt atribútov parameterValue v tabuľke *Parameters*.

**Reset** (BUTTON) Znovunačítanie formuláru.

**AdminRankingCriteria** (FORM) Tento formulár je určený pre manažovanie jednotlivých hodnotiacich kritérií, ich bližší popis a definíciu.

**AllCriteria** (TABLE) tabuľka obsahujúca všetky mená hodnotiacich kritérií.

**Name** (ATTRIBUT-LINK: *AdminRankingEdit*) Meno hodnotiaceho kritéria.

**Description** (ATTRIBUT) Bližší popis hodnotiaceho kritéria.

**Select** (ATTRIBUT-CHECKBOX)

**Add** (BUTTON-LINK: *AdminRankingEdit*)

**Delete** (BUTTON-LINK) Vymazanie vybraných hodnotiacich kritérií.

**AdminRankingEdit** (FORM)

**Name** (TEXT) Meno kritéria.

**Description** (TEXTAREA) Text, ktorý bližšie popisuje kritérium.

**All Values** (LIST) Zoznam všetkých hodnôt všetkých kritérií.

**Add Value** (BUTTON-LINK) Tlačidlo na pridanie hodnoty do množiny hodnôt tohto kritéria (*Criteria Values*).

**Remove Value** (BUTTON-LINK) Odstránenie hodnoty z množiny pre editované kritérium.

**Criteria Values** (LIST) Vybrané hodnoty pre editované kritérium.

**New Value** (TEXT) Ak chceme pridať novú hodnotu pre kritérium, sem napíšeme jej označenie.

**Add New Value** (BUTTON-LINK) Pridá novú hodnotu do *Criteria Values*.

**Ok** (BUTTON-LINK: *AdminRankingCriteria*) Potvrdenie zmien vo formulári.

**Cancel** (BUTTON-LINK: *AdminRankingCriteria*)

**AdminTopics** (FORM) Administrátor v tomto formulári bude môcť definovať jednotlivé témy konferencie, tj. pridávať, mazať, meniť názov a popis témy.

**Topics** (TABLE) Bude obsahovať atribúty z tabuľky tblTopics:

**Name** (ATTRIBUTE-LINK: *AdminTopicEdit*) Meno témy.

**Description** (ATTRIBUTE) Bližšie špecifikuje zvolenú tému.

**Select** (ATTRIBUTE-CHECKBOX)

**Add** (BUTTON-LINK: *AdminTopicEdit*) Pridanie novej témy.

**Delete** (BUTTON-LINK: *Confirmation*) Zmaže vybrané záznamy.

**AdminTopicEdit** (FORM) Formulár slúži na editáciu, už existujúceho záznamu, prípadne naplnenie novo vytvoreného záznamu. Bude obsahovať jednotlivé položky záznamu z tabuľky tblTopic:

**Name** (TEXT) Meno témy.

**Description** (TEXTAREA) Popis témy.

**Submit** (BUTTON-LINK) Potvrdenie záznamu a načítanie *AdminTopics*.

**Cancel** (BUTTON-LINK: *AdminTopics*) Načítanie *AdminTopics* bez uplatnenia zmien.

## 4.5 Návrh dátového modelu – verzia 2

Počas fázy návrhu a prototypovania sme postupne upravili, doplnili a zjemnili dátový model. Hrubý návrh dátového modelu uvedený v kapitole 4.1 je tu doplnený o atribúty, upravený podľa reakcií zákazníka na prototyp a rozvinutý do fyzického modelu dát.

Na rozdiel od modelu uvedeného v hrubom návrhu sme prešli k anglickým názvom entít a atribútov. Je to tak preto, lebo sme dátový model priamo transformovali do tabuliek databázového systému v implementácii a chceli sme zachovať korešpondujúce názvy tabuliek s názvami entít v modeli.

### 4.5.1 Logický model údajov

V tejto časti je uvedený logický model dát, ktorý vychádza z modelu uvedeného v kapitole 3.3, ale je pozmenený a doplnený, ako bolo uvedené v úvode tejto podkapitoly. Nasleduje popis entít, niektorých atribútov a väzieb medzi entitami s dôrazom na zmeny oproti pôvodnej verzii modelu. Samotný model je znázornený na obrázku 10.

**Person** Entita označujúca používateľa systému, ktorý je v systéme evidovaný a jednoznačne identifikovateľný. Zahŕňa účastníkov konferencie (pozri aj entitu *Participant*), autorov príspevkov, členov programovej komisie a administrátorov systému.

**Login, Password** Atribúty slúžiace pre zabezpečenie prístupu do systému.

**First name, Middle name, Surname, Titles** Meno a akademické tituly osoby.

**Email, Organization** Elektronická adresa osoby a jej pôsobisko.

**IsAuthor, IsReviewer, IsAdmin** Položky určujúce roly osoby v systéme. O jednotlivých rolách v systéme hovorí kapitola 3.4. Jedna osoba môže patriť do viacerých rolí a o priradení osôb do rolí oponenta a administrátora rozhoduje administrátor systému. Položka *IsAuthor* sa nastaví automaticky, ak osoba podá do systému príspevok do konferencie.

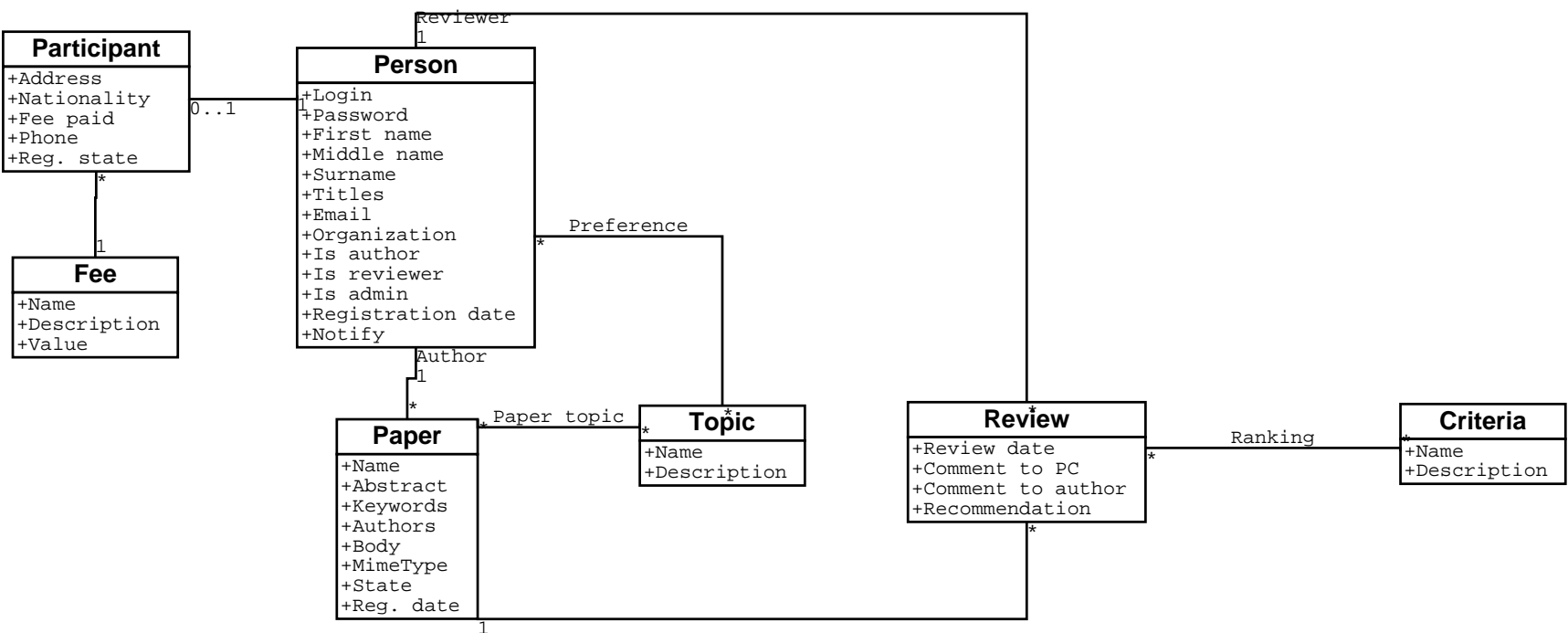
**RegDate** Dátum registrácie osoby do systému.

**Notify** Určuje, či si používateľ želá byť notifikovaný mejlom o prípadných zmenách, o prijatí či neprijatí článku a podobne.

**Participant** Entita zahrňujúca údaje o účastíkovi konferencie. Inštancia tejto entity sa vytvorí pre inštanciu entity *Person*, keď sa daná osoba zaregistruje ako účastník konferencie.

**Address, Nationality, Phone** Ďalšie doplňujúce atribúty pre účastníka konferencie pridané k tým, ktoré sa o ňom evidujú ako o osobe v inštancii entity *Person*.

**FeePaid** Dátum zaplatenia registračného poplatku. Ak poplatok ešte nebol uhradený, bude tu uložená nejaká definovaná hodnota (napríklad nula).



Obrázok 10: Logický model údajov

**RegState** Stav registrácie. Registrácia účastníka sa môže nachádzať v rôznych stavoch, ktoré závisia napríklad od zaplatenia poplatku, predbežnej registrácie a podobne.

**Fee** Entita poplatkov za účasť na konferencii. Je možných viacero rôznych typov účastníckych poplatkov, s rôznou cenou, napríklad v závislosti od členstva účastníka v nejakej organizácii podporujúcej konferenciu, od toho, či sa jedná o študenta alebo napríklad telesne postihnutého.

**Name, Description** Krátky názov poplatku a jeho bližší textový opis. Tu budú uvedené podmienky, za ktorých je možné použiť tento typ poplatku.

**Value** Peňažná hodnota poplatku v mene definovanej globálne pre celý systém.

**Paper** Entita príspevok do konferencie (článok). Toto je jedna z fundamentálnych entít v celom systéme Con4U, pretože k príspevku patrí jeho autor, oponenti, posudky.

**Name, Authors** Názov článku a jeho autori. Je potrebné rozlíšiť atribút *Authors*, ktorý je textovou položkou, do ktorej sú zapísaní všetci autori článku tak, ako sú uvedení na titulnej strane článku od atribútu *Author*, ktorý je väzbou na inštanciu entity *Person* a označuje autora, ktorý pridal tento článok do systému.

**Abstract, Keywords** Textové položky, ktoré obsahujú abstrakt a kľúčové slová príspevku. Znovu je potrebné rozlíšiť kľúčové slová (*Keywords*) od tém (*Topic*). Množina tém je globálna pre všetky príspevky a autor príspevku pre každý príspevok vyberie z tohto zoznamu tie, ktorých sa tento príspevok týka. Kľúčové slová sú voľným textom, ktoré zadáva autor a niekedy sa tiež uvádzajú v samotnom príspevku.

**Body** Samotné telo článku vo forme súboru.

**MimeType** Typ (formát) tela článku – súboru.

**State** Stav príspevku môže nadobudnúť jednu z hodnôt – prijatý, zamietnutý, ešte nerozhodnuté atď.

**RegDate** Dátum zaslania príspevku do systému.

**Topic** Téma konferencie. Tieto témy sa používajú pri bližšom určení článku (každému článku autor priradí, ktorých tém sa článok týka) a slúžia aj na vyjadrenie preferencií oponentov (každý oponent si vyberie, ktoré témy preferuje a podľa toho mu budú pridelované príspevky na oponovanie).

**Name, Description** Názov a bližší popis témy.

**Review** Entita posudok na príspevok. Táto entita je vo viacerých vzťahoch s niekoľkými inými entitami a preto je vhodné bližšie vysvetliť niektoré vzťahy aj na tomto mieste. Jeden oponent je autorom niekoľkých posudkov (na rôzne príspevky). Na jeden príspevok píše posudky niekoľkí oponenti. V každom posudku sa príspevok hodnotí podľa všetkých kritérií.

**ReviewDate** Dátum podania posudku do systému.

**CommentToAuthor** Poznámky, odporúčania, komentáre oponenta pre autora príspevku.

**CommentToPC** Poznámky, odporúčania, komentáre oponenta pre ostatných členov programovej komisie. Tieto môžu slúžiť ako podpora pri finálnom rozhodovaní o prijatí či neprijatí článku na konferenciu.

**Recommendation** Celkové odporúčanie oponenta ohľadom prijatia alebo neprijatia článku na konferenciu.

**Criteria** Entita kritérium hodnotenia príspevku. Príklady takýchto kritérií sú napr. originalita, vhodnosť, prínos a podobne. Príspevok sa posudkom hodnotí podľa všetkých týchto kritérií a pri hodnotení podľa každého kritéria sa príspevku priraduje hodnota, určujúca mieru naplnenia daného kritéria, napríklad výborná originalita, menšia vhodnosť na konferenciu, veľmi veľký prínos. . .

**Name, Description** Krátke meno kritéria a jeho obsiahlejší opis, ktorý môže zahŕňať aj návod na spôsob hodnotenia podľa tohto kritéria.

**Preference** Väzba určujúca, že daná téma (*topic*) je preferovaná daným oponentom (*reviewer*). Každý oponent môže mať niekoľko preferencií.

**PaperTopic** Väzba určujúca, že príspevok (*paper*) sa týka danej témy (*topic*). Jeden príspevok sa môže týkať viacerých tém, jednej témy sa môže týkať viac príspevkov.

**Ranking** Väzba opisujúca hodnotenie príspevku v posudku (*review*) podľa kritéria (*criteria*). Každý posudok hodnotí príspevok podľa viacerých (všetkých) kritérií a podľa jedného kritéria sa hodnotí vo viacerých (všetkých) posudkoch.

#### 4.5.2 Fyzický model údajov

Fyzický model dát vznikol transformáciou z logického modelu dát (jeho normalizáciou) a pridaním niektorých ďalších entít, väzieb a atribútov, ktoré riešia fyzickú realizáciu dát potrebných na zabezpečenie niektorých funkcií systému.

Vo fyzickom modeli sú už bližšie špecifikované typy atribútov entít.

V slovnom opise uvádzame len tie atribúty a väzby, ktoré boli pridané oproti logickému modelu.

**Person** Entita.

**Participant** Entita.

**Fee** Entita.

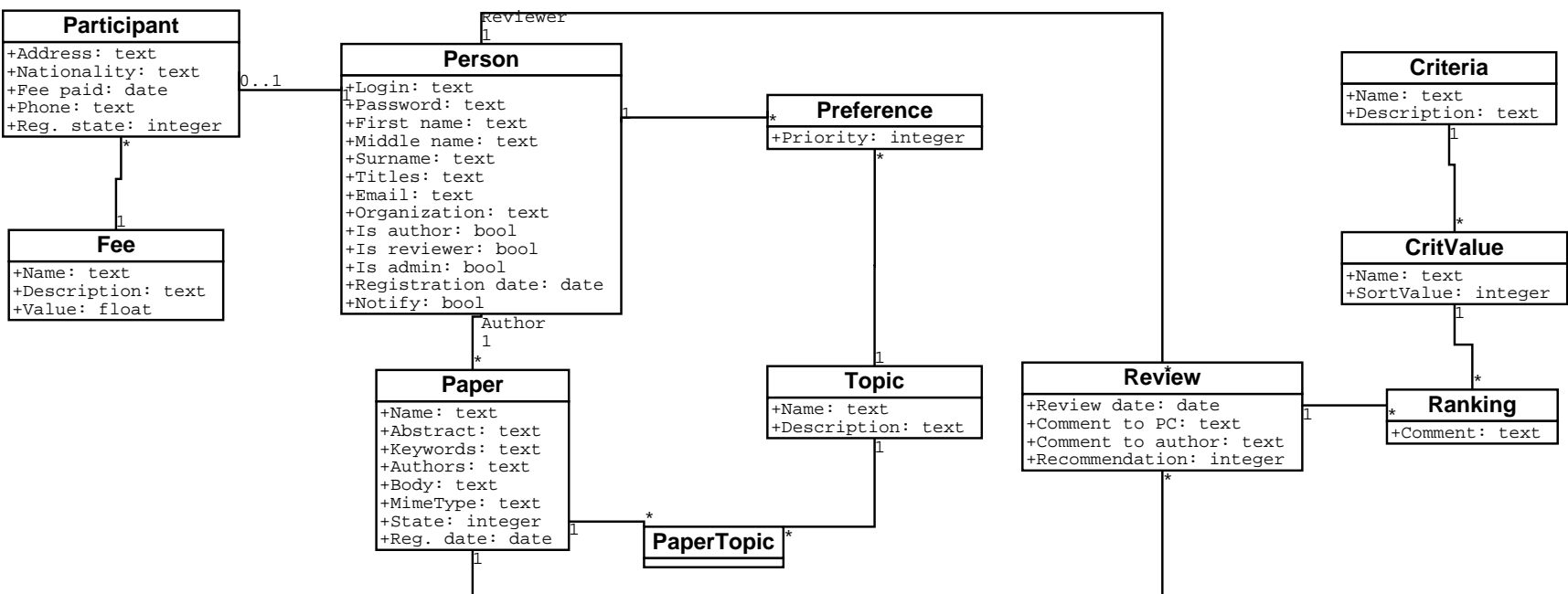
**Paper** Entita.

**Topic** Entita.

**Review** Entita.

**PaperTopic** Entita vzniknutá transformáciou väzby *PaperTopic* z logického modelu dát. Určuje, že príspevok sa týka témy.

**Preference** Entita vzniknutá transformáciou väzby *Preference* z logického modelu dát. Určuje, že oponent preferuje tému a mieru preferencie vyjadrenú prioritou.



Obrázok 11: Fyzický model údajov

**Priority** Priorita preferencie oponenta k téme.

**Ranking** Entita vzniknutá z logického modelu transformáciou väzby *Ranking*. Určuje hodnotenie posudzovaného príspevku podľa kritéria. Hodnotenie je vyjadrené priradením inštancie entity *CritValue* k inštancii entity *Ranking*.

**Criteria** Entita. Oproti dátovému modelu je entita *Criteria* s entitou *Ranking* zviazaná cez *CritValue*, čo je akýsi zoznam hodnôt pre jednotlivé kritéria.

**CritValue** Entita označujúca možnú hodnotu hodnotenia príspevku v posudku. Niekoľko položiek entity má vždy spoločný cudzí kľúč do tabuľky *Criterion*. Jedna inštancia entity teda určuje ako hodnotenie, tak aj kritérium, ktorého sa hodnotenie týka.

**Name** Textové vyjadrenie hodnotenia.

**SortValue** Aby bolo možné radiť textové označenia hodnotení, každé hodnotenie má priradenú túto pomocnú hodnotu.

**StaticPageObject** Entita pre stránky a iné objekty webovej prezentácie konferencie. Tieto objekty slúžia na dynamické, aktuálne informácie, pokyny, novinky a podobne, pridávané do systému dynamicky – za behu konferencie. Systém potom prezentuje linky na tieto objekty a na požiadanie tieto objekty prezentuje používateľom.

**Name** Názov objektu, ktorý bude použitý aj ako hypertextová linka na tento objekt.

**MimeType** Typ (formát) dokumentu – objektu.

**Body** Samotné telo objektu vo forme súboru.

**Protection** Atribút, ktorý určuje, či objekt bude prezentovateľný anonymným používateľom, autorom príspevkov, registrovaným účastníkom konferencie alebo oponentom. Možné sú aj kombinácie implementované bitovým súčtom hodnôt.

## 4.6 Revízia návrhu

Táto kapitola popisuje rámcové zmeny, o ktorých bolo rozhodnuté v prvých týždňoch letného semestra (prípadne ešte počas skúškového obdobia), ako aj opisuje navrhované časti, ktoré sme zabudli uviesť v predchádzajúcej fáze.

Pokým pôvodný návrh uvažoval o troch subsystémoch (ktoré sme nie celkom presne nazývali moduly), nový návrh vychádza zo zjednodušenej štruktúry, z ktorej úplne vypadol mail-subsystém (po dohode tímu so zákazníkom), aj keď na konci projektu sa ukázalo, že by možno našiel využitie.

Podstatné zmeny:

- V systéme sa nachádza tabuľka *Property*, v ktorej sa uchováajú dôležité nastavenia vo forme páru (*name*, *value*). V poslednom fyzickom modeli sa na ňu zabudlo, lebo nemá logický vzťah k jadru dátového modelu.
- Odpadá mailový subsystém.

- Používanie pojmu *modul* zodpovedalo skôr pojmu *subsystém*. Pre mnohé zjednodušenia sme prestali pojem *subsystém* napokon ale úplne používať. Pre istú množinu formulárov, ktoré sa starajú o istú časť dátového modelu, sme zvolili pomenovanie *modul*. Jeden súbor PHP bude implementovať typicky jeden formulár, preto väčšinou PHP súbor jednoducho voláme *formulár*.
- Pre implementáciu sme sa rozhodli využiť existujúcu knižnicu *Phormation* na reprezentáciu údajov z databázy na web-rozhraní.

## 4.7 Návrh modulov systému

V nasledujúcom popise *modul* označuje množinu príbuzných *formulárov*, kde jeden formulár je typicky implementovaný jedným PHP súborom.<sup>14</sup> V rámci modulu si formuláre predávajú isté informácie. Typicky v jednom module môže existovať jeden formulár pre prehľadovú tabuľku (tzv. *index* formulár) a jeden formulár pre detailný náhľad na konkrétnu položku (tzv. *detail* formulár). Za bežných okolností je detail volaný z indexu a cez metódu GET sa posielajú informácie o kľúčoch požadovaného záznamu.

### 4.7.1 Modul Registration

Ide o modul implementujúci formuláre:

**MyRegistration a AdminUserEdit (user\_detail.php)** Oba formuláre zabezpečujú rovnaké funkcie.

#### Funkcie:

1. Editáciu záznamu tabuľky Person (pre všetky role).
2. Editáciu záznamu tabuľky Participant (pre rolu Participant).
3. Vytváranie väzieb na tabuľku Topic – tzv. Preferencie (pre rolu Reviewer).

#### Použitie:

1. Po zvolení linky My Registration (role Participant, Reviewer, Author).
2. Po zvolení konkrétneho používateľa v administrácii používateľov (rola Administration).

### 4.7.2 Modul Papers (Author)

Ide o modul implementujúci formuláre:

**MyPapers (paper\_index.php)**

#### Funkcie:

1. Prehľad článkov – sú zobrazené atribúty Name a State z tabuľky Paper.
2. Linka na pridanie článku.

---

<sup>14</sup>Z tohto pravidla sa samozrejme našli aj výnimky.



**Použitie:**

1. Po zvolení linky My Papers (rola Author).

**EditPaper (paper\_detail.php)** Formulár je určený ako na editáciu existujúceho záznamu o príspevku, tak aj na vytvorenie nového záznamu.

**Funkcie:**

1. Editácia záznamu tabuľky Paper.

**Použitie:**

1. Po zvolení funkcie Add vo formulári MyPapers.
2. Po zvolení konkrétneho príspevku zo zoznamu vo formulári MyPapers.

#### 4.7.3 Modul Reviews (Reviewer)

Ide o modul implementujúci formuláre:

##### MyReviews (review\_index.php)

**Funkcie:**

1. Prehľad príspevkov na oponovanie.

**Použitie:**

1. Po zvolení linky My Reviews (rola Reviewer).

##### ReviewPaper (review\_detail.php)

**Funkcie:**

1. Editácia záznamu tabuľky Review.
2. Vytváranie väzby medzi Review a CritValue (tabuľky Ranking) – ide o samotné hodnotenie príspevku.

**Použitie:**

1. Po zvolení funkcie Add vo formulári MyReviews.
2. Po zvolení konkrétneho review zo zoznamu vo formulári MyReviews.

#### 4.7.4 Modul Administration (Administrator)

tabuľky,

Ide o modul implementujúci nasledovné formuláre:

**SystemInformation (system\_info.php)****Funkcie:**

1. Prehľad o stave systému a priebehu konferencie (počet používateľov, počet príspevkov, počet zoponovaných príspevkov, termínov).

**Použitie:**

1. Po zvolení linky System Information.
2. Automaticky po prihlásení administrátora.

**AdminUsers (user\_index.php)****Funkcie:**

1. Prehľad používateľov registrovaných v systéme – prehľad tabuľky Person.
2. Linka na pridanie používateľa (viď modul Registration).
3. Mazanie používateľov (aj hromadne).

**Použitie:**

1. Po zvolení linky Users z administrátorského menu.

**AdminReviewAssignment (review\_assignment.php)****Funkcie:**

1. Prehľad príspevkov s informáciami o procese ich oponovania.
2. Mazanie príspevkov z konferencie (aj hromadne).

**Použitie:**

1. Po zvolení linky Review Assignment z administrátorského menu.

**AdminArticleReviews (article\_reviews.php)****Funkcie:**

1. Prehľad najdôležitejších informácií o článku.
2. Priradovanie oponentov pre jeden vybraný príspevok.
3. Automatický návrh priradenia (na základe preferencií oponentov).

**Použitie:**

1. Po zvolení príspevku zo zoznamu z formulára AdminReviewAssignment.
2. Po zvolení príspevku zo zoznamu z formulára AdminReviewArticles.

**AdminReviewArticles (review\_articles.php)****Funkcie:**

1. Prehľad tých informácií o oponentovi, ktoré sú rozhodujúce pre priradenie príspevku.
2. Priradovanie príspevkov jednému oponentovi.
3. Automatický návrh priradenia (na základe preferencií oponenta).

**Použitie:**

1. Po zvolení oponenta zo zoznamu z formulára AdminArticleReviews.

**AdminProperties (properties.php)****Funkcie:**

1. Editácia položiek tabuľky Properties.

**Použitie:**

1. Po zvolení linky Con4U Setup z administrátorského menu.

**AdminRankingCriteria (ranking\_index.php)****Funkcie:**

1. Prehľad informácií z tabuľky Criteria.
2. Možnosť mazať kritériá (aj hromadne).
3. Linka na pridanie nového kritéria.

**Použitie:**

1. Po zvolení linky Ranking Criteria z administrátorského menu.

**AdminRankingEdit (ranking\_detail.php)****Funkcie:**

1. Pridávanie a odoberanie hodnôt pre kritérium.
2. Zoradenie hodnôt pre kritérium.
3. Definovanie váhy pre kritérium.

**Použitie:**

1. Po zvolení konkrétneho kritéria zo zoznamu z formulára AdminRankingCriteria.
2. Po zvolení linky na pridanie nového kritéria.

**AdminTopics (topic\_index.php)****Funkcie:**

1. Prehľad tabuľky Topics.
2. Možnosť mazania kritérií (aj hromadne).
3. Linka na pridanie novej témy konferencie.

**Použitie:**

1. Po zvolení linky Topics z administrátorského menu.

**AdminTopicsEdit (topic\_detail.php)****Funkcie:**

1. Editovanie záznamu tabuľky Topics.

**Použitie:**

1. Po zvolení konkrétnej témy zo zoznamu z formulára AdminTopics.

**4.7.5 Modul CustomPages****AdminCustomPages (custom\_index.php)****Funkcie:**

1. Prehľad tabuľky StaticPageObject.
2. Možnosť mazania voliteľných stránok (aj hromadne).
3. Linka na pridanie novej stránky.

**Použitie:**

1. Po zvolení linky Custom Pages z administrátorského menu.

**AdminCustomPageEdit (custom\_detail.php)**

**Funkcie:**

1. Editovanie záznamu tabuľky StaticPageObject.
2. Možnosť priamej editácie textu stránky.
3. Linka na preview stránky.
4. Tlačítko pre upload stránky.

**Použitie:**

1. Po zvolení konkrétnej stránky zo zoznamu z formulára AdminCustomPages.

**4.7.6 Modul Phormation**

Tento modul bude použitý pre jednoduché generovanie *index* a *detail* formulárov. Pre zložitejšie úlohy sa Phormation upraví podľa našich požiadaviek, alebo sa formulár implementuje bez využitia tohto modulu. Zmeny do phormation boli navrhnuté len veľmi zbežne, nakoľko sa aj tak počítalo s istými požiadavkami zákazníka na vzhľad. Isté bolo len, že sa tlačítka na mazanie nahradia selectami a tlačítkom pre možnosť hromadného mazania a prechod na detail sa vyrieši linkou miesto tlačítka.

**4.7.7 Modul spoločných funkcií (include.php)**

V module spoločných funkcií budú spoločné funkcie, ktoré sa vkladajú ako jeden súbor *include.php* do prakticky všetkých nami používaných formulárov (okrem phormation knižnice samozrejme). Modul implementuje aj autorizačné funkcie. Modul bude zrejme podliehať ešte mnohým zmenám, takže neboli špecifikované jeho funkcie.

## 5 Špecifikácia požiadaviek 2

Na začiatku letného semestra došlo k zmene špecifikácie projektu, nakoľko mal byť nasadený v školskom prostredí na odovzdávanie projektov (záverečných, diplomových). V ďalšom postupe projektu bola snaha, aby bolo stále možné Con4U nasadiť do konferenčného prostredia s minimálnymi implementačnými zmenami, čo sa odrazilo v návrhu systému konfigurácie pomocou tabuľky vlastností. Akékoľvek ďalšie snaženie tímu ale viedlo projekt do cieľového (tj. do školského) prostredia.

V tejto kapitole je uvedený kompletný zoznam požiadaviek zákazníka, pričom pôvodný návrh systému ostáva naďalej v platnosti. Rozdiely medzi návrhom a samotnou implementáciou (popísanou v kapitole 6) nie sú príliš podstatné – implementovaných však bolo viac formulárov, než je v pôvodnom návrhu, nakoľko sa implementovalo podľa požiadaviek zákazníka. Bližšie k tomuto problému v podkapitole 5.4.

Nasleduje zoznam zmenených termínov, po nich podrobný zoznam požiadaviek a potom tabuľka, v ktorej je uvedený dosiahnutý stav (a teda aj to, čo ešte nie je spravené).

### 5.1 Zmeny v terminológii

**študent** tento pojem nahrádza označenie role *author*;

**učiteľ (pedagóg)** je náhradou za označenie role *reviewer*;

**práca (projekt)** nahrádza pojem *príspevok, článok (paper)*;

**oponent (recenzent)** učiteľ, ktorý vyplňa k práci oponentský posudok;

**vedúci učiteľ**, ktorý vyplňa k práci posudok vedúceho;

**administrátor** bez zmien.

### 5.2 Zoznam požiadaviek

Pôvodné požiadavky boli zoskupené podľa rolí, toto je zachované, sú však pridané aj ďalšie požiadavky v iných skupinách. Z tohto dôvodu je za menom požiadavky v zátvorke uvedené, kto ju zadal (nakoľko niektoré požiadavky sme si zadali sami, keď nám vyplynuli ako nevyhnutnosť) a ďalšie prípadné poznámky.

#### Požiadavky pre rolu študent

##### 1.1 prihlásenie (zákazník)

Študent sa musí vedieť prihlásiť do systému.

##### 1.1.1 po prihlásení zobrazíť priamo prácu (zákazník)

Došpecifikované tak, že ak má študent jedinú prácu, nezobrazuje sa navrhnutý formulár `paper_index.php` (čo je stránka, navrhnutá pre zobrazenie po prihlásení), ale rovno `paper_detail.php`. Funkcia je konfigurovateľná.

##### 1.1.2 miesto indexovej stránky zobrazovať hneď prihlasovací formulár (zákazník)

Funkcia má byť konfigurovateľná.

### 1.2 vložiť nový abstrakt (zákazník)

Abstrakt sa vkladá v dvoch jazykoch (v našom prípade v slovenskom a anglickom).

### 1.3 vložiť text práce (zákazník)

Bez zmien v návrhu – práca sa bude posielat' na server cez formulár metódou *post* (tzv. *upload* súboru). Zákazník upozornil na fakt, že práca môže mať niekoľko MB. V dobe zadania požiadavky sme nevedeli určiť limitujúce faktory.

### 1.4 modifikovať abstrakt, text (zákazník)

Modifikácia je rovnaký proces ako vkladanie abstraktu a textu do práce pre jestvujúcu položku v databáze.

### 1.5 zadať údaje o projekte a autorovi (zákazník)

Jedná sa o nasledujúce údaje:

1. Meno študenta;
2. jeho identifikačné číslo;
3. názov projektu v slovenskom jazyku;
4. názov projektu v anglickom jazyku;
5. vedúci projektu (výber zo zoznamu vedúcich);
6. e-mailová adresa.

Tieto požiadavky boli neskôr upresnené a so zákazníkom sme sa dohodli na tom, že údaje 1, 2, 3 a 5 sa budú importovať (tým pribudol modul importu do implementácie). Zvyšné údaje doplní študent v svojom profile (navrhnutý formulár `user_detail.php`, údaj 6), resp. vo svojej práci (navrhnutý formulár `paper_detail.php`, údaj 4).

## Požiadavky pre rolu učiteľ

### 2.1 prihlásenie (zákazník)

Návrh systému je založený na rolách používateľov, prihlasovanie je tak zabezpečené pre všetky role.

### 2.2 prezerať abstrakty všetkých prác (zákazník)

### 2.3 vidieť práce pridelené na posudzovanie (zákazník)

Je nutné odlišiť práce, pre ktoré je učiteľ vedúcim, od tých, ktoré oponuje. Táto požiadavka si vynútila zavedenie pojmu *typ posudku*.

### 2.4 vyjadriť preferencie pri posudzovaní (zákazník)

Toto sa týka oponovania prác. Problémom je, že škála prác pokrýva veľa tém, takže využitie *tém* (pôvodne *témy konferencie*) je menej prínosné ako pri konferencii, ktorá je zameraná na istý okruh tém. Bola tu ešte možnosť rozdeliť pomocou tém práce aspoň podľa odborov (napr. SI, PSS pre odbor Informatika). Táto požiadavka napokon nebola doriešená. Výskyt pojmu *téma* v systéme je aj z tohto dôvodu konfigurovateľný a v prvej prevádzke Con4U boli témy vypnuté (viď popis vlastností systému v kapitole 6.3). Vyjadrenie preferencií tak ostáva procesom prebiehajúcim mimo systém, pričom administrátor potom ručne priraduje práce k oponentom (požiadavka 3.2).

## 2.5 vyplniť posudok (zákazník)

Posudky sú rôzne pre oponenta a vedúceho. Rovnako tak sa líšia posudky pre rôzne typy posudkov a rôzne typy projektov. Problém rôznych projektov nebol riešený, nakoľko už pri pôvodnom návrhu konferenčného systému sa uvažovalo o rôznych inštanciách pre rôzne konferencie – podobne teda pre každý projekt musí byť nainštalovaná jedna inštancia systému.

### 2.5.1 rôzne typy posudkov (vyplývalo z 2.5, doplnené tímom)

Pridanie pojmu *Typ posudku* je najväčšou návrhovou zmenou ako reakcia na nové požiadavky. Na typ posudku sa viažu hodnotiace kritéria, na informáciu o type sa odkazuje samotný posudok, čo určuje vzťah učiteľa k danej práci (viď implementovaný dátový model 6.1). Nebolo žiaduce, aby systém napevno obsahoval dva typy posudkov, preto sa typy stali súčasťou dátového modelu a bola pre ne vytvorená tabuľka.

### 2.5.2 celkové hodnotenie (doplnené tímom ako súčasť dekompozície požiadavky 2.5)

Celkové hodnotenie je kritérium, tj. pri vyplňaní posudku sa preň volí hodnota. Pre svoje špecifické vlastnosti ale bolo nutné sa ním zaoberať osobitne. Pri vyplňaní posudku mala byť k dispozícii možnosť výpočtu navrhovaného celkového hodnotenia zo všetkých kritérií, samozrejme okrem samotného celkového hodnotenia.

## 2.6 vytlačiť posudok (zákazník)

Vytvoríť výstupný formulár pre vyplnený posudok tak, aby bol vhodný priamo na tlač z WWW prehliadača.

## Požiadavky pre rolu administrátor

### 3.1 všetky funkcie čo učiteľ (zákazník)

### 3.2 pridelíť práce na posudzovanie (zákazník)

Oponent, vedúci, vedúci sa prípadne pridelí automaticky. Počas neskoršieho jednania bolo dohodnuté, že vedúci sa pridelí v dobe importu študentov, prác a učiteľov do systému. Oponent sa bude prideľovať ručne.

#### 3.2.1 rôzne typy posudkov (došpecifikované tímom)

Formuláre pre pridelenie posudkov musia obsahovať možnosti pre voľbu typu posudku.

### 3.3 prezeráť všetky práce (zákazník)

Pomocou jedného formulára bude dostupný zoznam prác, formulár pre jednu prácu bude zhodný s formulárom, pomocou ktorého študent zadáva údaje o práci.

### 3.4 prezeráť výsledky posudzovania (zákazník)

#### 3.4.1 zobrazenie práca-autor-posudzovateľ (došpecifikované so zákazníkom)

Na základe 3.4 boli navrhnuté prehľadové zoznamy posudkov.

### 3.5 vytlačiť posudky (zákazník)

Funkčnosť ako 2.6.



### 3.6 archivácia abstraktov a prác (zákazník)

Možná (zákazníkom navrhovaná forma) bolo vytvorenie archívu, kde by každý študent mal svoj adresár, v ktorom by sa nachádzali všetky relevantné súbory.

### 3.7 založenie nového projektu (zákazník)

Došpecifikované ako inštanciovanie celého systému – systém teda nepodporuje v jednej inštalácii viacero projektov.

### 3.8 definovanie učiteľov do systému (zákazník)

Bola požadovaná aj možnosť importu. Došpecifikované na import súboru typu CSV, ktorý je možné ľahko vyprodukovať z MS Excel, kde typicky zoznamy vznikajú.

### 3.9 definovanie študentov do systému (zákazník)

Funkčnosť ako pri 3.8. Druhou alternatívou je nechať študentov registrovať sa v systéme. Obe funkčnosti boli špecifikované, hoci preferovaný bol import. Druhá funkčnosť je obdobou registrácie autorov na konferenciu.

### 3.10 ad-hoc SQL dotaz (zákazník)

Vypíše sa zoznam tabuliek a pole pre zadanie SQL dotazu. Odpoveď sa zobrazí vhodne v tabuľke.

### 3.11 vytvoríť dokument s abstraktami (zákazník)

Došpecifikované ako export abstraktov vo formáte RTF podľa zákazníkom dodanej šablóny. Dodatočne boli špecifikované isté požiadavky na použitie štýlov vo výslednom RTF dokumente (štýl Head pre hlavičku a abstract pre text abstraktu).

### 3.12 konfigurácia systému (zákazník)

Požiadavky na rôzne konfigurovateľné vlastnosti vznikali potom priebežne, jasne špecifikovaných bolo 5, mnohé sa implementovali bez špecifikácie.

#### 3.12.1 readonly formuláre (zákazník, tím)

Pre situácie obdobné stavu, kedy sa na konferencii prestanú prijímať príspevky, je vhodné, aby doteraz modifikovateľné formuláre boli naďalej len pre čítanie. Túto požiadavku sme došpecifikovali do takej miery, že každý formulár, ktorý je realizovaný pomocou Phormation môže mať túto vlastnosť nastavovateľnú. Zákazník túto potrebu prejavil najmä pri formulári *paper\_detail.php*, aby študent nemohol modifikovať údaje o práci, ak tak zvolí administrátor.

#### 3.12.2 voliteľný pojem topics (zákazník, tím)

Nakoľko nie vždy sú témy potrebné, je možnosť ich vôbec nepoužívať. Nezobrazujú sa vo formulároch, nekonfigurujú sa.

#### 3.12.3 voliteľný pojem keywords (tím)

Kľúčové slová sa týkajú len práce a majú len informatívny význam. Nakoľko bolo zrejmé, že nebudú potrebné napríklad pri odovzdávaní záverečných projektov, bolo umožnené ich výskyt v systéme potlačiť.

#### 3.12.4 kritériá hodnotenia (zákazník, tím)

V súvislosti so zavedením novej entity *typ\_posudku* bolo nutné prešpecifikovať konfiguráciu kritérií hodnotenia – kritérium sa vzťahuje ku konkrétnemu typu posudku.

**3.12.5 prednastavené hodnoty pre kritériá (zákazník, tím)**

Mnohé kritériá používajú tú istú množinu hodnôt. Miesto konfigurácie týchto množín bol napokon zvolený model použitia funkcie na skopírovanie hodnôt z iného kritéria.

**3.12.6 voliteľný pojem authors (tím)**

Veľmi podobné ako 3.12.3 – ide o informatívne pole pre prácu. Na konferenciách je možné, že prácu realizujú viacerí autori, čo je dôvod pre existenciu poľa. Jeho výskyt je možné potlačiť.

**Iné požiadavky****4.1 lokalizácia (zákazník, tím)**

Spôsob zmeny textov by mal byť čo najjednoduchší. Došpecifikované tak, že ide síce o implementačnú činnosť, ale len v súboroch deklaratívnej povahy – všetky texty sa budú nachádzať vo vyhradenom súbore.

**4.2 mail notifikácia (zákazník, 17. 4. 2002)**

Po kompletnom odovzdaní práce študentom treba poslať mail administrátorovi (alebo napr. tajomníčke katedry).

**5.3 Stav požiadaviek na konci projektu**

Požiadavka	Realizované	Poznámka
1.1	áno	
1.1.1	áno	
1.2	áno	limit na veľkosť abstraktu je 1024 znakov
1.3	áno	limit na veľkosť súboru je 2 MB
1.4	áno	viď limity pre 1.2 a 1.3
1.5	áno	riešené importom údajov
2.1	áno	
2.2	áno	
2.3	áno	
2.4	<b>nie</b>	nebolo dohodnuté využitie tém
2.5	áno	
2.5.1	áno	
2.5.2	áno	
2.6	áno	
3.1	áno	
3.2	áno	
3.2.1	áno	
3.3	áno	
3.4	áno	
3.4.1	áno	
3.5	áno	obdoba 2.5
3.6	áno	
3.7	nie	realizované ďalšou inštaláciou systému
3.8	áno	import

Požiadavka	Realizované	Poznámka
3.9	áno	import
3.10	áno	
3.11	áno	
3.12	áno	
3.12.1	áno	
3.12.2	áno	property <i>bool_topics</i>
3.12.3	áno	property <i>bool_keywords</i>
3.12.4	áno	tabuľky <i>criteria</i> , <i>criteria_value</i>
3.12.5	áno	funkcia typu <i>create-like</i>
3.12.6	áno	property <i>bool_authors</i>
4.1	áno	vydelenie textov do špeciálnych súborov
4.2	<b>nie</b>	neskoré zadanie, špecifikácia do budúcnosti

## 5.4 Dôsledky pre návrh

Žiadny z nasledujúcich bodov nebol formálne navrhnutý – implementovalo sa prakticky okamžite, ako bol na danú požiadavku čas.

- zavedenie typu posudku (zmena dátového modelu);
- návrh importu z CSV (comma separated value – tj. nepriamo z Excelu);
- návrh generovania RTF dokumentov z PHP;
- implementácia mnohých ďalších formulárov spĺňajúcich používateľské požiadavky.

## 6 Implementácia

Táto kapitola popisuje implementáciu systému, pričom sa zameriava na zdôraznenie menej štandardných riešení. Je určená pre vývojárov, ktorý by chceli pokračovať vo vývoji systému alebo využiť niektoré jeho časti. Preto zdôrazňuje najmä netypické riešenia, ktoré sme použili, ako aj uvádza dôvody, prečo sme dané riešenie použili.

### 6.1 Implementovaný dátový model

Finálny dátový model, ktorý je možné v podobe SQL popisu tabuliek nájsť v adresári **SQL**, je uvedený na obrázku 12.

### 6.2 Popis modulov a formulárov

Implementácia systému je realizovaná sadou PHP skriptov, pričom v hlavnom adresári projektu sa nachádzajú všetky naše formuláre a ďalšie využívané súbory. V podadresári **SQL** sú skripty pre SQL databázu – v našom prípade pre MySQL. V podadresári **phormation** je nami upravená verzia knižnice **phormation**. Pri opise je použitý odsek **TODO** pre veci, ktoré treba dorobiť, a **Hacks** označuje nečisté riešenia (je to dosť subjektívny pohľad) alebo proste veci, ktoré nie je ľahké zistiť čítaním kódu.

#### 6.2.1 Knižnica **phormation**<sup>15</sup>

Knižnicu **phormation** sme využili na jednoduché generovanie formulárov dvoch typov:

- prehľad celej tabuľky (nemuselo pritom ísť len o jednu tabuľku v databáze, zobrazíť je možné ľubovoľný `select`) – jednotky zdrojového kódu implementujúce tento typ pohľadu majú meno ukončené typicky `_index.php`;
- pohľad na jeden záznam v tabuľke (tu je jednoduché prezerať záznam z jednej databázovej tabuľky, akékoľvek spojenia sú zložitejšie) – zdrojové súbory majú typicky koncovku `_detail.php`.

Samotná knižnica je dobre zdokumentovaná z pohľadu klientskeho programátora, rovnako tak nie je problém knižnicu modifikovať. Pred ďalším popisom zmien, ktoré sme vykonali v jednotlivých PHP súboroch knižnice je dobré si preštudovať túto dokumentáciu. Nasleduje len popis zmien:

**data\_fcns.php** bez zmien

**dbi.php** bez zmien

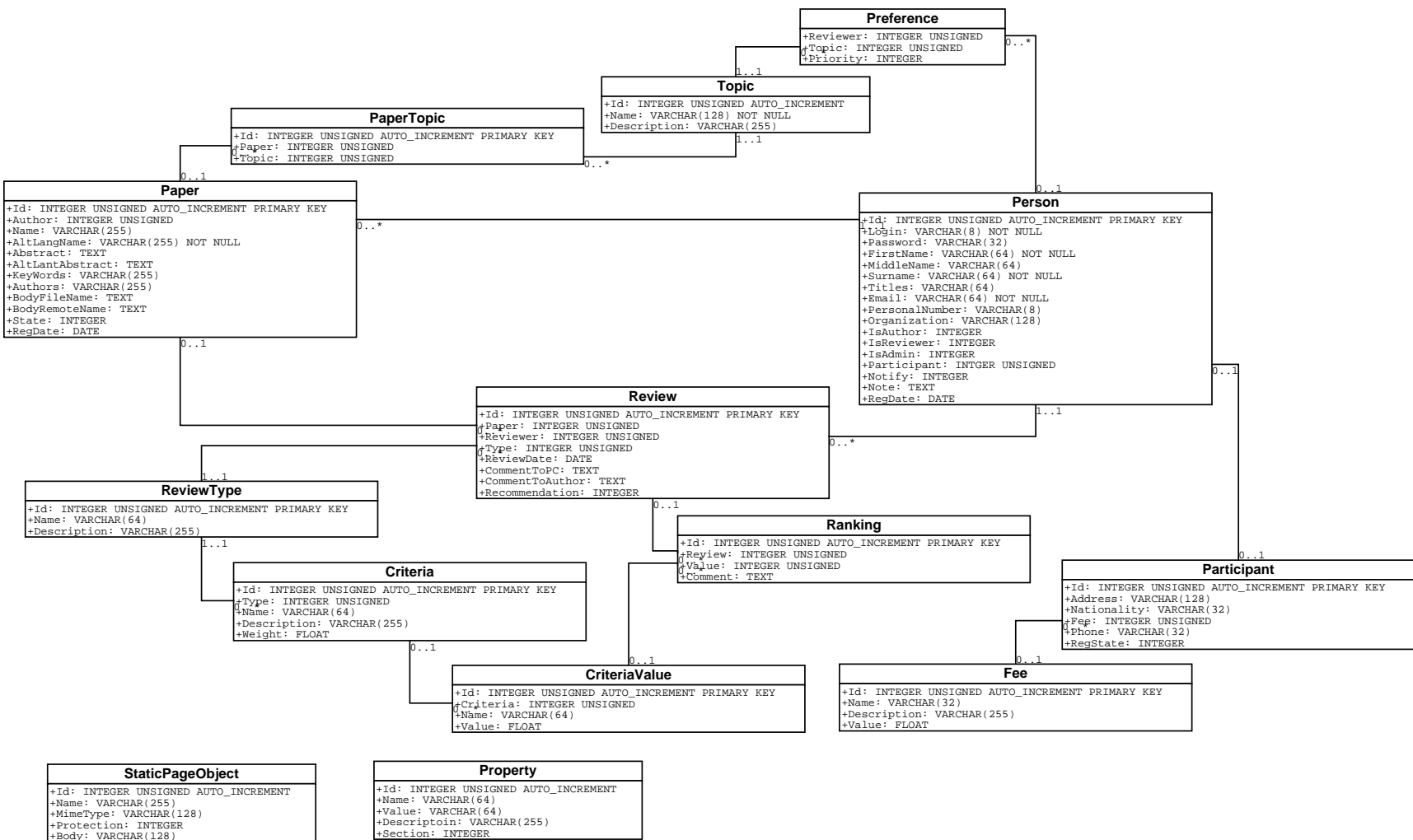
**display\_table.php**

- Texty prerobené na systém využívajúci `lang.php`<sup>16</sup>.
- Miesto tlačítka na mazanie záznamu sa objavuje `select box` na konci riadku tabuľky – mazanie označených záznamov je potom realizované tlačítkom pod tabuľkou.

---

<sup>15</sup><http://www.peaceworks.ca/phormation.php>

<sup>16</sup>Aby **phormation** ostal samostatnou knižnicou, obsahuje vlastný `lang.php`.



Obrázok 12: Fyzický dátový model

- Miesto tlačítka prechádzajúceho na detail sa táto činnosť realizuje s využitím parametra *fkeylinks* v klientskom formulári.
- Riadky tabuľky sú vykresľované za pomoci štýlov (classes *phindex...*).
- Pridané linky na hlavičke tabuľky, ktorými je možné zmeniť zoradovanie (zvoliť stĺpec, resp. obrátiť smer).

**file\_upload.php** bez zmien

### **form.php**

- Využitie *lang.php* pridaného do phormation.
- Parameter *readonly* pre celý formulár, každý widget dostane nastavenie *readonly*.
- Parameter *noread* pre konkrétny field zabezpečí, že sa nenačíta hodnota s databázy – vhodné v spojení s *default*.
- Podpora parametrov *title* a *instr* (titulok a ďalší HTML kód pred formulárom).

**lang.php** Komplet pridané definície pevných reťazcov pre ľahšiu modifikáciu.

**misc\_fcns.php** bez zmien

### **multiform.php**

- Len jeden reťazec presunutý do *lang.php*.
- Nevyužívali sme, takže možno nesedí celkom do filozofie ostatných zmien.

### **sql\_fcns.php**

- Pri spracovávaní dotazov pre form sa vynechával *xref* widget, lebo pracoval s inou tabuľkou než tou, na ktorú sa vzťahuje formulár (*form.php*). My sme pridali aj ignorovanie *ranking* widgetu – z toho istého dôvodu.

**widgets.php** Súbor implementuje jednotlivé prvky formulára (*form.php*).

- Pevné reťazce presunuté do *lang.php*.
- Niektoré widgety obsahujú implementáciu *readonly* parametra, čo je systematickejšie ako ošetrovať túto požiadavku v klientskom formulári a používať v podmienkach *readonly* widget (implementované pre widgety *text*, *select*, *fkey*, *xref*, *file*, *xrefselect*). Tento parameter sa pridáva poľu aj vtedy, ak má celý formulár nastavený parameter *readonly* (poľe *opts* vo *form.php*).
- Vyrobený *html* widget na priame vkladanie HTML kódu.

- Vyrobený *ranking* widget. Tento widget je poutívaný v *review\_detail.php* v samotnej aplikácii a ide o jeden veľký **hack**. Niektoré jeho časti sú celkom korektne spravené. Samotný *form.php* používa funkcie *make\_...*, *validate\_...* a *default\_set\_...*. Ďalšie dve funkcie sú oproti pôvodnému vzoru (xref) mierne modifikované – predáva sa im *fieldname* a z neho sa potom vytvára premenná na priame získavanie hodnôt z HTML formulára, lebo táto priama cesta je najistejšia (inak bolo problémom získať vždy hodnoty tak, ako to bolo potrebné, hoci toto priame riešenie nevyužíva možno isté naplnené údajové štruktúry phormation). Widget prijíma veľké množstvo parametrov, ale mnohé ďalšie veci sú napevno nakódované vo funkciách, čo bolo dané rýchlou potrebou implementácie. Obsahuje aj odkaz na javascriptovú funkciu *otvor\_okno* pre zobrazenie popisu kritéria. Funkcia je v stávajúcom systéme implementovaná vo formulári *review\_detail.php*.
- Pomocná funkcia *make\_option()* pre jednoduchšie ošetrenie parametra *readonly* u viacerých widgetov.
- Podpora parametra *default* pre widgety *fkey* a *ranking*.

### 6.2.2 Kaskádne štýly

**con4u.css** Štýl použitý vo väčšine stránok. Definuje základný look&feel a classy pre phormation zoznamy so striedavo farbenými riadkami.

**report.css** Štýl pre tlačové reporty – použitý v *review\_detail.php*.

### 6.2.3 Podporné PHP skripty

**db.php** Súbor obsahuje údaje potrebné na pripojenie sa k databáze. Je použitý v *include.php*. V súbore sú definované nasledovné premenné:

**db\_host** meno servera, na ktorom ebzi sql server, napríklad „localhost“;

**db\_database** názov používanej databázy, napr. „zp“;

**db\_user** používateľ databázy, nap. „con4u“;

**db\_password** heslo používateľa do databázy.

**include.php** Súbor obsahuje spoločne využívané funkcie a je vkladáný do väčšiny ďalších PHP skriptov implementujúcich formuláre. Okrem toho sa po vložení *include.php* inicializuje premenná *\$conn* používaná na prístup k databáze.

Modul závisí od týchto modulov:

- db.php
- lang.php
- phormation/dbi.php
- con4u.css

**Dôležité funkcie** (pre typické použitie mnohých z nich odporúčame jednoducho prezrieť zdrojový kód formulárov – napr. *topic\_index.php*):

**html funkcie**

**html\_begin()** Zabezpečí kompletný začiatok HTML kódu aj s menu vľavo.

**html\_end()** Ukončuje HTML kód.

**html\_begin\_print()** Začiatok HTML kódu bez menu.

**html\_end\_print()** Ukončenie HTML kódu začatého funkciou *html\_begin\_print()*.

Okrem týchto funkcií boli chvíľu priamo v *include.php* implementované aj funkcie *html\_before\_table()*, *html\_after\_table()*, resp. ekvivalenty pre form, vhodné pre phormation – funkcie realizovali len *html\_begin()* a *html\_end()*. Nakoľko ale nebolo možné predefinovať funkcie vo formulároch, kde okrem štandardného začiatku a konca stránky boli požadované aj iné prvky, od tohto riešenia sa ustúpilo a funkcie sa podľa potreby implementujú v jednotlivých formulároch – väčšinou práve ako volania *html\_begin()* a *html\_end()*.

**menu()** Funkcia vypisujúca menu (použitá v *html\_begin()*). Menu má tieto súčasti:

- linky na statické stránky (custom pages, plne konfigurovateľné) podľa rolí používateľa;
- linky na stránky pre registráciu autorov a účastníkov konferencie (ak je registrácia povolená v systémových vlastnostiach);
- linka na prihlasovaciu obrazovku (alebo linka pre odhlásenie prihláseného používateľa);
- linky pre stránky pre autorov, oponentov, administrátorov

**db\_connect()** Napojenie na databázu podľa údajov uvedených v *db.php*.

**db\_disconnect()** Odpojenie sa od databázy.

**autorizačné funkcie**

**get\_my\_id()** Vrátí primárny kľúč v tabuľke Person, ktorý zodpovedá aktuálnemu používateľovi systému (menu a heslu).

**for\_roles(\$roles)** Pre role uvedené v poli *\$roles* povolí ďalší beh, inak presmeruje HTTP požiadavku na *index.php*. Role sú identifikované nasledovnými reťazcami:

- **administrator**
- **reviewer**
- **author**
- **participant**

**is\_administrator()** Funkcia vracia *true*, ak používateľ disponuje rolou administrátor. Obdobné funkcie sú aj pre ďalšie tri role (s analogickými názvami).

**get\_property(\$name)** Vrátí hodnotu ku špecifikovanému kľúču (k menu) danej vlastnosti z tabuľky *Property*.

**TODO:**

- Oddeliť funkcie pre definovanie vzhľadu stránok od spoločne používaných funkcií systému.

**Hacks:**



Žiadne (v spomínaných funkciách, v iných je hackov viac než dosť)

**lang.php** Súbor obsahuje definície reťazcov, ktoré sú použité v ďalších PHP súboroch. Umiestnenie všetkých reťazcov na jedno miesto uľahčuje napríklad lokalizáciu. Je pravdepodobné, že oddelenie všetkých reťazcov nie je plne realizované. Tento súbor sa používa v *include.php*, takže všetky skripty, ktoré *include.php* používa, automaticky používajú aj *lang.php*.

#### 6.2.4 Implementácia formulárov

Väčšina PHP skriptov implementuje priamo jeden formulár. Niektoré formuláre na seba naväzujú a predávajú si navzájom parametre – využíva sa metóda GET aj POST (podľa povahy parametrov). Závislosti na *include.php* nie sú uvádzané, lebo prakticky všade je tento súbor vkladajú a používaný. Rovnako tak nie je explicitne zdôrazňované, že formulár *x\_index.php* obsahuje odkazy na *x\_detail.php*.

**abstract\_doc.php** Miesto vkladania PHP do HTML ide v tomto prípade o vkladanie PHP do RTF a výstupom je potom RTF podľa požiadavky 3.11. Ťažko sa v tom orientuje, ale ide o štandardné riešenie.

**abstract\_report.php** Realizuje formulár zobrazujúci všetky abstrakty. Nepoužíva phormation.

##### parametre

(get) alt nastavené na true zobrazuje miesto primárneho jazyka názvy prác a text abstrakov v alternatívnom jazyku.

**adhoc\_sql.php** Modul slúži na zadávanie a zobrazovanie výsledku ľubovoľného SQL výrazu.

##### Parametre:

desc=<table name> slúži na zobrazenie príkazu desc v novom okne;

SQL slúži na zobrazenie výsledku ľubovoľného výrazu.

**TODO:** Prepísať rozhodovaciu podmienku, ktorá rozhoduje či bude použité select like výstup, alebo či sa zobrazí úspešnosť prevedeného SQL výrazu a počet ovplyvnených riadkov.

**Hacks:** Nie je použitá databázovo nezávislá funkcia *dbi\_fetch\_array*, ale *mysql\_fetch\_array*, kvôli použitému flagu `MYSQL_ASSOC`, ktorý zabezpečí rozumný zoznam fieldov v *array\_result* (v pôvodnom prevedení *dbi\_...* sú uvádzané len tie fieldy, ktoré mali vo výsledku nenulové hodnoty, inak sa meno fieldu vo výsledku neuviedlo)

**allpaper\_index.php** Modul slúži na zobrazovanie zoznamu všetkých prác evidovaných systémom.

##### Parametre:

alt=(yes, no) zapína - vypína zobrazovanie jazykovo alternatívnych názvov prác. Pokiaľ sa parameter neuvedie, zobrazujú sa aj jazykovo natívne a alternatívne názvy článkov v systéme.

**archive.php** Archivuje abstrakty prác (v oboch jazykoch) a práce každého študenta. Archív obsahuje pre každého študenta adresár "<priezvisko meno>\\" v ňom súbory:

- "<názov práce> abstract" – abstrakt v slovenčine;
- "<názov práce> abstract in alt. lang" – abstrakt v alternatívnom jazyku;
- "<názov práce>" samotná praca (uploadnutý súbor).

#### používa property

**tmp\_folder** adresár, ktorý bude dočasne vytvorený (musí byť vytvoriteľný apache) a v ňom bude vytvorená vyššie popísaná štruktúra súborov;

**archive\_name** path a meno archívu (musí byť vytvoriteľný apache);

**archiver** meno a parametre archivera pre rekurzívnu archiváciu súborov; "<archiver> <archive\_name> \*" musí byť legálny príkaz;

**delete\_files** príkaz pre vymazanie adresára a celého jeho obsahu rekurzívne (rm -r v UNIXe, deltree pre win32);

**Obmedzenie** mená autorov (študentov) musia byť jedinečné a autor môže mať len jednu prácu s daným názvom

**Bug/TODO:** phormation zle interpretuje znak backslash „\“, nezapisuje ho do databázy a zle ho vypisuje z databázy. Problémy sa môžu vyskytnúť pri zadávaní mena súboru.

**assignment.php** Umožňuje administrátorovi priradovať oponentov jednotlivým prácam. Nepoužíva knižnicu phormation.

- Volá sa bez parametrov po zvolení Pridelenie posudkov. Parameter *rtype* sa používa pri výbere typu posudku, ktorý sa bude priradovať. *Id* identifikuje prácu/oponenta s ktorou/ktorým sa pracuje. *Adding* informuje o pridávaní položky. V závislosti od kombinácií hodnôt jednotlivých vstupných parametrov sa volá jedna z funkcií.
- Funkcia *papers11()* vykonáva priradenie medzi prácami a oponentami. Je potrebné zabezpečiť modifikáciu len tých priradení, ktoré sa skutočne zmenili.
- Funkcia *papers12()* zobrazuje informácie o konkrétnom oponentovi a všetkých prácach, pričom umožňuje vykonať priradenie medzi ním a ľubovoľnou prácou (prácami).
- Funkcia *papers13()* zobrazuje informácie o konkrétnej práci a všetkých oponentoch, pričom umožňuje vykonať priradenie medzi prácou a ľubovoľným oponentom (oponentami).
- Funkcia *papers2()* zobrazuje informácie o všetkých posudkoch určitého typu.
- Funkcia *jumpMenu(targ, selObj, restore)* zabezpečuje výber a zmenu typu vypisovaných posudkov.

**criteria\_desc.php** Malé pop-up okno otvárané z *review\_detail.php* pri kliknutí na *Popis* pri konkrétnom kritériu. Parametrom je id kritéria.

**criteria\_detail.php** Zobrazuje informácie o kritériu a zoznam preň vytvorených hodnotení. Kombinuje phormation FORM a phormation TABLE.

- Je volaný z *reviewtype\_detail.php* a po návrate z editácie stupňa hodnotenia. Parameter *record\_id* v prípade, že je kladný určuje položku v tabuľke Criteria. Ak je záporný určuje typ posudku, pre ktorý bude vytvárané nové kritérium.
- Využíva *criteriavalue\_detail.php*.
- Funkcia *confirm\_delete()* prekrýva štandardnú otázku na zmazanie záznamu z knižnice phormation.
- Funkcia *my\_delete\_fnc(\$record\_id)* realizuje vymazanie hodnotenia, ktorého id dostane a tiež odkazov naň v tabuľke Ranking.
- Tlačidlo delete (ozn. ako clone) je použité na vytvorenie kritéria s rovnakým zoznamom hodnotení.
- Hodnoty sú pri výbere v posudku zobrazované v poradí, ktoré určuje ich hodnota.

**criteriavalue\_detail.php** Vstup stupňa hodnotenia pre kritérium. Volá sa z *criteria\_detail.php*. Parameter *record\_id* v prípade, že je kladný určuje položku v tabuľke CriteriaValue. Ak je záporný určuje kritérium, pre ktoré bude vytváraný nový stupeň hodnotenia.

**custom\_detail.php** Modul slúži na prácu so stránkami editovateľnými priamo v systéme. Umožňuje editovanie a prezeranie týchto stránok. Implementuje aj riadenie prístupu podľa role prezerajúceho.

#### vstupné parametre

**page\_id** číslo stránky, ktorú ideme zobraziť, alebo editovať;

**body** uploadnutý súbor;

**text** textová forma súboru;

**action** hodnota „Edit“ znamená, že sa uloží text z premennej text, hodnota „Add“ uloží uploadnutý súbor body.

**TODO:** Čo najviac celý modul prerobiť. Tento modul mal malú prioritu a nebol v ňom vôbec použitý phormation.

**Hacks:** Celé je to hack.

**custom\_index.php** Klasický index formulár implementovaný pomocou phormation pre tabuľku *StaticPageObject*.

**download.php** Odošle súbor na serveri (parameter *file*) používateľovi s lokálnym názvom (parameter *filename*) ako MIME typ application/binary.

**emptyabstracts\_index.php** Zobrazuje tabuľku prác (a ich autorov), ktoré nemajú vyplnený abstrakt s linkami na prácu (*paper\_detail.php*).

**import.php** Modul slúžiaci na importovanie údajov do systému. Je možné importovať zoznam študentov, učiteľov, alebo importovať spolu študentov, ich práce, ich učiteľov ako vedúcich konkrétnych prác. Pri importovaní sa kontroluje, či už daný objekt nie je zavedený. Ak áno nebude vytvorený duplicitne.

#### vstupné parametre

**file** súbor obsahujúci vstupné dáta

**button** typ importu

**delimiter** oddeľovací znak medzi položkami importovaného súboru (default „;“)

**revtype** má význam pri importe študentov a prác. Učiteľovi uvedenému pri práci bude pridelený posudok na túto prácu typu, ktorý je uvedený v revtype.

**Ostatné závislosti** Modul k svojej činnosti môže vyžadovať podporu knižnice iconv v PHP. Pomocou tejto knižnice sa prevádza kódovanie z windows-1250 na iso-8859-2. Ak táto knižnica nie je dostupná, je potrebné zadávať súbory zásadne v kódovaní iso-8859-2.

**TODO:** Možnosť lepšej konfigurácie formátu vstupných súborov. V terajšej fáze je tento formát natvrdo zapísaný v zdrojovom kóde.

**login.php** Modul, ktorý prihlasuje a odhlasuje používateľa. Pri prihlásení nastaví cookies name a pass. Pri nesprávnom prihlásení ich zruší. Pri odhlásení ich takisto zruší.

**TODO:** Mohlo by sa namiesto posielanie loginu a hesla pri každom prístupe používať nejaké formy sessions.

**Hacks:** Spoliehame sa na to, že všetci používatelia majú zapnuté cookies.

**nopapapers\_index.php** Zoznam študentov, ktorí nemajú pridelenú prácu, s linkami na ich profil (user\_detail.php).

**paper\_detail.php** Modul, ktorý slúži na modifikáciu údajov o príspevku. Používa klasický detail z knižnice phormation. Widget na uploadovanie súboru je z phormation. Ako tabuľku na generovanie jedinečných mien súborov používa PaperCounter.

**paper\_index.php** Modul na výpis zoznamu príspevkov jedného autora. Používa klasický index z knižnice phormation.

**properties\_detail.php** Klasický detail formulár týkajúci sa tabuľky *Property* využívajúci knižnicu phormation.

**properties\_index.php** Klasický index formulár týkajúci sa tabuľky *Property* využívajúci knižnicu phormation.

**review\_detail.php** Zabezpečuje vstup posudku v závislosti od typu oponenta.

- Je volaný z *review\_index.php* s parametrom *record\_id*, ktorý určuje záznam v tabuľke Review.
- Využíva *review\_report.php*.

- Funkcia *preloz\_pre\_skolu(\$hodnota,&\$cvid)* zaokrúhľuje (postup podľa dodaného xls súboru) vypočítané celkové hodnotenie a prevádza ho na slovný popis. Slovný popis je robený v stupnici jednotlivých hodnotení (nie celkového) a predpokladá sa, že budú zhodné. Pretože je povinnosťou oponenta zvoliť hodnotenie každého kritéria, automatické dopĺňanie pre celkové hodnotenie nie je (teda nebude) implementované.
- Používa property *bool\_review\_comment*.
- Ponuka celej stupnice so zvýraznením odporúčaním bude pravdepodobne nejaký hack, ktorý popíšem neskôr. Momentálny stav je taký, že používa stĺpec Recommendation z tabuľky Review a zobrazuje iba vypočítanú hodnotu.
- Podobne tlačidlo submit/compute bude realizované cez dve tlačidlá.
- Kritérium s nulovou váhou je považované za celkové hodnotenie a zobrazené ako posledné. Zoradenie kritérií je realizované na základe poľa description, ktoré na začiatku obsahuje číselnú hodnotu.
- Zoradenie stupnice hodnotenia pre kritérium je na základe číselných hodnôt, ktoré majú priradené jednotlivé stupne.

**review\_index.php** Zobrazuje oponentovi zoznam posudkov, ktoré má vytvoriť resp. ktoré vytvoril.

- Zobrazuje sa po prihlásení oponenta a po návrate z *review\_detail.php*.
- Využíva *review\_detail.php*.
- Funkcia *format\_rev\_date(\$date)* zobrazuje dátum vyplnenia posudku. Ak je dátum „0000-00-00“, považuje sa posudok za nevyplnený.
- Funkcia *format\_file(\$body)* vypisuje názov súboru s prácou ako linku, cez ktorú je možné súbor stiahnuť.
- Používa property *review\_deadline*.

**review\_report.php** Report využívajúci štýl *report.css* slúžiaci na tlač hotového posudku. Naimplementované bez pomoci phormation.

**reviewtype\_detail.php** Zobrazuje informácie o posudku a zoznam preň vytvorených kritérií. Kombinuje phormation FORM a phormation TABLE.

- Je volaný z *reviewtype\_index.php* a po návrate z editácie kritéria. Parameter *record\_id* určuje položku v tabuľke ReviewType.
- Využíva *critria\_detail.php*.
- Funkcia *my\_delete\_fnc(\$record\_id)* zabezpečuje vymazanie kritéria, ktorého id dostane, jeho hodnôt a tiež odkazov na ne v tabuľke Ranking.
- Kritéria sú v posudku zobrazované v poradí, ktoré určuje obsah poľa description, kde by na začiatku malo byť uvedené poradové číslo.

- Súčet váh musí byť 1, čo by sa mohlo aj testovať, no nerobí sa to.

**reviewtype\_index.php** Zobrazuje administrátorovi zoznam typov posudkov, ktoré existujú.

- Zobrazuje sa po zvolení Typy posudkov v menu administrátora a po návrate z *reviewtype\_detail.php*.
- Využíva *reviewtype\_detail.php*.

**system\_info.php** Určené pre správcu, poskytuje globálne informácie o systéme: počet študentov, učiteľov, používateľov, prác posudkov, počet študentov bez práce, nevyplnených abstraktov, nedokončených abstraktov a linky na tabuľky obsahujúce: všetky práce, študenti bez práce, nevyplnené abstrakty, nedokončené posudky implementované v nasledujúcich súboroch.

**table\_index.php** Zobrazuje tabuľky *Práce* a *Nedokončené posudky* určené pre správcu.

- Stĺpce sú: názov práce, študent (autor práce), dátum odovzdania, typ posudku, učiteľ (posudzovateľ) a v prípade tabuľky "Práce" aj dátum posudzovania.
- Riadky: jeden riadok pre každý posudok (pár: práca - posudzovateľ, keďže práca má aj oponenta aj vedúceho); v prípade tabuľky "Nedokončené posudky" linka na nedokončený posudok (*review\_detail.php*).

**topic\_detail.php** Modul na modifikovanie jednej témy konferencie. Používa klasický detail z knižnice phormation.

**topic\_index.php** Modul na výpis zoznamu všetkých tém konferencie. Používa klasický index z knižnice phormation.

**user\_detail.php** Modul slúži na editovanie detailu užívateľa systému. Chovanie modulu je silne závislé od role používateľa, ktorá daný modul zavolala. Pre neadministrátora je povolené editovať iba svoj profil. Administrátor môže meniť akékoľvek Id a môže zmeniť rolu v požadovanom profile.

#### Parametre:

**who=(author,reviewer,participant)** vytvára nového používateľa s danou rolou;  
**record\_id=** editovanie existujúceho používateľa, ktorý ma Id=record\_id.

**Statické parametre:** (Parametre napísane v kóde)

**self\_reg\_perm[]=(author,participant,administrator,reviewer)** pole, ktoré určuje role, ktoré sa môžu samostatne registrovať, tj. nepotrebujú administrátorské práva na vytvorenie.

#### Property parametre:

**bool\_editlogin** povolenie užívateľom editáciu login mena;  
**bool\_personalnumber** povolenie existencie osobného čísla;  
**bool\_topics** povolenie výberu tém pri editácii profilu reviewera.

**TODO:** Dopísať kód, ktorý zabezpečí zmenu role "Participant".

**Hacks:** Vlastné `xrefselectWidgets` hook funkcie – nutné kvôli problémom s referencovanou tabuľkou, ktorá pri použití štandardných hook funkcií obsahovala všetky referencie, aj keď daná priorita preferencie bola 0. Ďalším dôvodom bola nutnosť zobrazovania preferencií, ktoré ešte nemuseli byť zapísané v databáze pri registrácii.

**Výsledok:** Vlastná hook funkcia ukladá preferencie iba ak majú váhu viac ako 0. Pri load hook funkcií sa načítavajú všetky preferencie z referencovanej tabuľky a z tabuľky `Topics`.

**user\_index.php** Modul slúži na zobrazovanie zoznamu všetkých používateľov, alebo len požadovaných filtrom a jednoduché mazanie vybraných používateľov.

**Parametre:**

**adm=(true,false)** zapne/vypne zobrazovanie všetkých administrátorov;

**rev=(true,false)** zapne/vypne zobrazovanie všetkých reviewerov;

**par=(true,false)** zapne/vypne zobrazovanie všetkých participantov;

**aut=(true,false)** zapne/vypne zobrazovanie všetkých authorov.

**TODO:** Dopísať delete query, aby sa zmazali všetky závisle záznamy z referencovaných tabuliek.

### 6.3 Vlastnosti systému (properties)

Po mene vlastnosti nasleduje jej vysvetlenie, napokon je tučne uvedená prednastavená hodnota v systéme pre odovzdávanie zadání (takto sú vlastnosti definované v súbore **SQL/properties.sql**).

**archiver** Archivačný príkaz (default: **tar czf**).

**archive\_name** Meno pre cieľový súbor archívu. **/tmp/archiv.tgz**

**bool\_authorreg** Môžu sa registrovať autori (zobrazenie linky na Registráciu autorov – určené pre konferenciu). **false**

**bool\_authors** Určuje, či sa bude používať pole *authors* z tabuľky `Person`. Ide o informačné pole pre prípad viacerých autorov jednej práce (konferencia). **false**

**bool\_editlogin** Je povolené meniť login pre neadministrátora? **false**

**bool\_keywords** Používa sa v konferencii kľúčových slov (informačné pole tabuľky `Paper`). **false**

**bool\_login\_index** Má byť index stránka presmerovaná na login? **true**

**bool\_nonewpaper** Nemôžu autori vytvárať nové príspevky (študenti práce)? **true** (tj. nemôžu)

**bool\_paperedit** Je možné modifikovať práce (po „uzávierke príspevkov“ nastaviť na **false**)? **true**

**bool\_paperindex** Má sa zobrazovať `paper_index`, ak je počet prác rovný jednej? **false**

**bool\_participants** Pracuje sa s pojmom participant? **false**

**bool\_personalnumber** Pracuje systém s poľom osobné číslo (tabuľka Person)? **true**

**bool\_review\_comment** Zobrazujú sa *Comment to author* a *Comment to PC* v review\_detail? **false**

**bool\_topics** Pracuje systém s pojmom témy? **false**

**delete\_files** Príkaz pre rekurzívne mazanie (kvôli archivácii). **rm -r**

**odbor** Používané v reportoch pre review. **Informatika**

**review\_deadline** Dátum ukončenia posudzovania. **15. 4. 2002** (čisto informatívna hodnota vypisovaná v *review\_index.php*)

**rname\_oponent** Databázové meno typu posudku pre oponenta. **Oponent**

**rname\_veduci** Databázové meno typu posudku pre vedúceho. **Veduci**

**skolsky\_rok** Text dopĺňaný do posudkov (report, dokument). **2001/02**

**tmp\_folder** Dočasný adresár pre prácu archivátoru. **/tmp/con4u**

**veduci\_rev\_type** Id typu posudku (ReviewType), ktorý bude pri importe ponúkať ako default. **1 (toto treba upraviť podľa skutočného id!!!)**

## 6.4 Konfiguračné poznámky

Z implementačných dôvodov je v súbore `/etc/php.ini` bezpodmienečne nutné aktivovať nasledujúcu voľbu:

```
allow_call_time_pass_reference = On
```

Phormation využíva často volanie odkazom v mieste volania – prerobiť všetok kód v tejto knižnici tak, aby bolo možné túto voľbu vypnúť (čo je čistejšie riešenie a do budúcnosti odporúčané), nie je celkom triviálne a po miernych pokusoch sme to vzdali, nakoľko to nebolo náš primárny cieľ.

O ďalších voľbách súvisiacich s inštaláciou je zmienka v Príručke, ale táto sa úzko dotýkala implementácie, preto je vysvetlená aj tu.

## 6.5 Navrhované vylepšenia, chyby

- Nefunguje dobre *custom\_detail.php*. Nebol čas sa tým bližšie zapodívať, zrejme došlo k problému pri nejakom „vylepšovaní“. Problém sa prejavuje PHP warningom po odoslaní stránky.
- Pri povolenom editovaní loginu sa nikde nezabezpečuje jeho jednoznačnosť!
- Property *rname\_veduci* a *veduci\_rev\_type* majú podobnú funkciu, bolo by dobré tento systém (spolu s property *rname\_oponent*) zjednotiť. Pomocou *rname\_...* je toto nastavovanie pre používateľa prirodzenejšie.



- Vôbec nebol implementovaný participant.

Okrem týchto návrhov sú niektoré spomenuté v sekciách TODO pri popise implementácie formulárov.

## 7 Test systému

V dobe odovzdávania bol systém už istý čas nasadený a používaný, pričom jeho funkčnosť je zatiaľ v zásade bez vážnejších problémov, hoci sa už realizovali isté námety používateľov a administrátorské rozhranie, ako aj rozhranie pre učiteľa sa naďalej doladuje.

### 7.1 Akceptačný test Systému na odovzdávanie projektov

Deň pred odovzdaním bol na testovacej prevádzke realizovaný akceptačný test, kedy sme podľa požiadaviek uvedených v kapitole 5.2 vykonali test a zaznačili odchýlky od požadovaného správania, prípadne okomentovali nejasnosti. Funkcie, ktoré spĺňajú špecifikáciu, sme neodvzdali.

- Z nepochopených príčin sa našla práca, ktorú nebolo možné uploadnúť (požiadavka 1.3). Väčšina uploadov ale ide, zatiaľ neznáma príčina.
- Vyskytli sa isté problémy v importe (nekonzistencie, istých študentov neimportlo s prácou, ako keby ich poznalo, atď), neznáma príčina, nevyriešené (zistené v poslednom týždni) – ide o požiadavky 1.5 a 3.9, nie je vylúčené aj ohrozenie 3.8.
- Neriešené 2.4.
- Posudok na tlač nebol nikdy schválený vo vytlačenej forme, ale aj prípadné úpravy by mali byť triviálne (2.6).
- Doposiaľ sa nerobila archivácia, prebehli len interné testy, nebolo schválené zákazníkom (3.6).
- Systém je možné nasadiť, ale proces nebol schválený zákazníkom – treba akceptovať (3.7).
- Pri konfigurácii sa akceptovali nastavenia vhodné pre odovzdávanie projektov a je možné, že rôzne kombinácie spôsobnia nie vždy celkom očakávané chovanie (3.12).
- Import formulár nie je lokalizovaný a je možné, že sa to týka viacerých formulárov, nakoľko mnohé sa vyvíjali do poslednej chvíle (4.1).
- Nerealizovaná mail notifikácia (4.2).

Tento zoznam nevyklučuje mnohé iné problémy, ktoré súvisia s akceptáciou systému pre odovzdávanie projektov. V aplikácii takejto povahy môžu vzniknúť mnohé stavy, ktoré nevieme dopredu odhadnúť.

### 7.2 Testovanie nepoužitých funkcií

Pri akceptačnom teste sa testovalo s takým nastavením systému (*properties*), ktoré zodpovedalo školskému prostrediu odovzdávania projektov. Okrem tohto testu prebehol test chovania pri inom nastavení, ako aj nahodilý test vecí, ktoré nás voľne napadli, pričom sme využívali znalosť systému ako bielej skrinky. Opäť uvádzame len anomálie od očakávaného chovania, ako aj podozrenia na prípadné výskyty chýb, ktoré sme ale nestihli úplne overiť.

- Formulár *user\_detail.php* by bolo dobré otestovať v spojitosti so všetkými nastaveniami relevantných properties. Mnohé veci v ňom by možno mali byť na istých properties závislé a nie sú. Toto je nutné lepšie premyslieť, čo nechávame na ďalší vývoj.

## 8 Záver

V čase písania tohto záveru je projekt v pilotnej prevádzke – a samozrejme sa na ňom objavujú chyby. Produkt Con4U rozhodne nie je uzavretý, ale je tu nutnosť uzavrieť tímový projekt. Snažili sme sa naimplementovať maximum z požadovaných požiadaviek a na tomto mieste aj zdokumentovať maximum toho, čo môže pomôcť prípadnému ďalšiemu vývoju.

Proces dokumentácie počas našej práce bol trochu slabší, lebo sme sa skutočne sústreďovali na rýchly vývoj, aby zákazník mohol pozorovať jednotlivé nové prvky.

Podarilo sa nám vytvoriť systém, ktorý – hoci s pochopiteľnými problémami tesne po nasadení – môže byť dobrým pomocníkom pri odovzdávaní projektov, ako aj pri procese ich posudzovania. Možnosti jeho ďalšieho rozširovania a vylepšovania tu nepochybne sú a snažili sme sa ponechať našim nasledovníkom v tejto úlohe dostatok priestoru.