Aplikácia na monitorovanie prípravy obhajoby dizertácie

MARTIN BIES

2008

Aplikácia na monitorovanie prípravy obhajoby dizertácie

BAKALÁRSKA PRÁCA

Martin Bies

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA INFORMATIKY

Študijný odbor: INFORMATIKA

Školiteľ bakalárskej práce:
Doc. RNDr. Pavol Ďuriš, Csc.

BRATISLAVA 2008

Abstrakt

BIES, Martin: Aplikácia na monitorovanie prípravy obhajoby dizertácie [bakalárska práca] – Fakulta matematiky, fyziky a informatiky Univerzity Komenského v Bratislave; Katedra informatiky. – Školiteľ: Doc. RNDr. Pavol Ďuriš, Csc. – Bratislava 2008. 34 strán.

Bakalárska práca sa zaoberá návrhom, analýzou a vývojom webovskej aplikácie pre monitorovanie priebehu. Zaoberá sa podrobným popisom programu po technickej aj užívateľskej stránke. K bakalárskej práci je priložená výsledná aplikácia pre monitorovanie prípravy obhajoby dizertácie.

Kľúčové slová: monitorovanie, PHP, SQL

Predslov

Informačné technológie sa stávajú čoraz viac súčasťou našich životov a snažia sa nám ich uľahčiť rôznymi spôsobmi. Jedným z nich je automatizácia činností, ktoré človek pri výkone svojej práce, alebo aj v bežnom živote vykonáva. Druhým je pamätanie si vecí, ktoré by si bežne človek musel držať vo vlastnej pamäti, alebo niekde na mieste, vďaka ktorému nezabudne, čo má kedy vykonať. Väčšina z nás používa kalendáre, kde si zapisuje dôležité veci, posledné roky hlavne elektronické, ktoré sú k dispozícii v rôznych formách. Najväčšou výhodou je že nás na nadchádzajúcu udalosť upozornia. Vďaka nim sa môžeme vyhnúť zbytočným komplikáciám, ktoré vyvstanú často iba z typicky ľudskej vlastnosti – zábudlivosti.

Preto je témou tejto bakalárskej práce práve návrh a vývoj aplikácie, ktorá má za úlohu pomôcť zúčastneným v procese obhajoby dizertačnej práce vykonať si svoje povinnosti včas a efektívnejšie. Predseda komisie má prehľad o priebehu všetkých dizertácií doktorandov na jednom mieste, doktorandi si môžu skontrolovať či proces ohľadne ich dizertácie prebieha tak ako má, alebo sa niekde na niekoho čaká. Oponenti a komisia sú informovaní emailami o tom či si nemajú v najbližších dňoch vykonať nejakú povinnosť. Tento systém poskytuje zúčastneným stranám vhodnou formou práve tie informácie, ktoré potrebujú.

Ďakujem svojmu vedúcemu bakalárskej práce, Doc. RNDr. Pavlovi Ďurišovi, Csc. za rady a pripomienky pri písaní tejto práce. Zároveň by som chcel poďakovať RNDr. Jaroslavovi Janáčkovi za pomoc pri výbere technickej realizácie bakalárskej práce.

Čestne prehlasujem, že túto bakalársku prácu som vypracoval samostatne len s použitím uvedenej literatúry.

V Bratislave 11. júna 2008 _____

Obsah

Úvod

Okrem samotnej aplikácie je súčasť práce aj analýza, návrh, technický aj funkcionálny dizajn a popis vytvorenej aplikácie. Práve týmto témam sa venujem na nasledovných stranách.

V prvej kapitole je uvedené zadanie práce od Doc. RNDr. Pavla Ďuriša, Csc. , moja analýza požiadaviek premietnutá do stručnej kostry aplikácie. Ďalej som uviedol programovacie jazyky a nástroje použité pri práci a dôvody prečo som si ich vybral.

Druhá kapitola obsahuje popis aplikácie, rozdelenie na triedy, popis tried, funkčnosti, databázový model aplikácie aj s detailným popisom tabuliek a ich významu.

V tretej kapitole sa venujem aplikácii z pohľadu užívateľa, popisu jednotlivých stránok a celkovej funkčnosti globálne.

Štvrtá kapitola obsahuje popis zabezpečenia systému a ako je chránená pred základnými typmi útokov.

Názvy tried, premenných, databáz sú zobrazované kurzívou. Niektoré termíny píšem v anglickom jazyku kvôli zaužívanému používaniu týchto termínov aj v slovenských textoch (napríklad session – v slovenskom jazyku sa používajú názvy ako sedenie, alebo relácia, ale session sa mi zdá byť jednoznačnejšie).

Kapitola 1

1.1 Zadanie úlohy

Zadanie úlohy sformulované vedúcim práce Doc. RNDr. Pavlom Ďurišom, Csc.:

Doktorand odovzdá dizertačnú prácu a administrátor do monitorovacieho systému uloží informácie:

1. meno doktoranda (aj email adresu)
2. názov dizertácie
3. názov študijného programu a odboru
4. meno školiteľa (aj email adresu)
5. návrh 3 oponentov dizertácie (vrátane mena, adresy pracoviska a email adresy)
6. autoreferát dizertácie (PDF súbor)
7. dátum odovzdania dizertácie
8. (prípadne ďalšie info)

Systém o tomto automaticky informuje predsedu príslušnej komisie pre obhajoby dizertácií.
Aby nedochádzalo k zbytočným meškaniam, musí byť systém postupne informovaný o tom, či predseda obhajovacej komisie vykonal (zabezpečil vykonanie) činností (A) až (F) v predpokladaných termínoch:

A) Zaslanie (členom komisie) mailov pre schválenie oponentov. Maily musia obsahovať aj dátum, do kedy sa môžu členovia komisie vyjadrovať k oponentom. *Termín*: bezodkladne
B) Informovanie systému o schválení oponentov. *Termín*: podľa dátumu v mailoch z bodu (A)
C) Urgovanie oponentov v prípade nezaslania posudkov načas. *Termín*: podľa vyhlášky/zákona

D) Informovanie systému o obdržaní všetkých posudkov. *Termín*: bezodkladne po obdržaní posudkov.

E) Zaslanie (členom komisie) mailov pre schválenie dátumu/hodiny a miesta konania obhajoby dizertácie. (Dátum je daný príslušným pravidlom podľa vyhlášky/zákona - najskôr 6 týždňov od zverejnenia informácie o konaní obhajoby.) (Maily musia obsahovať aj informáciu, dokedy sa môžu členovia komisie vyjadrovať k dátumu/hodine a miestu konania obhajoby) *Termín:* bezodkladne

F) Informovanie systému o schválení dňa/hodiny a miesta obhajoby *Termín*: podľa dátumu v mailoch z bodu (E).

Systém by mal čo najviac uľahčiť predsedovi činnosti (A), (C) a (E), napríklad: mal by predsedovi automaticky ponúknuť vo vhodnom čase vopred pripravené vhodné maily, ktoré by predseda (po prípadnej modifikácii) zaslal . (V takýchto prípadoch je systém schopný informovať sám seba o vykonaní týchto činností).

Systém musí poskytovať predsedovi komisie možnosť "manuálneho" vkladania informácií o vykonaní činností (A) až (F).

Systém musí poskytovať (cez vhodné web stránky a/alebo automatickým zasielaním mailov) prehľad o tom, či a kedy boli vykonané činnosti (A) až (F). Tento prehľad je určený oponentovi (doktorandovi) a členom komisie. Predovšetkým oponent (doktorand) má potom možnosť kontrolovať činnosť predsedu a v prípade problémov môže žiadať nápravu.

Aké maily, komu, kedy a za akých okolností zasielať a aké informácie z web stránky komu kedy, sprístupniť, musí byť parametrizovateľné a ľahko modifikovateľné.

Ak systém zistí, že doň nebola vložená informácia o vykonaní danej činnosti ((A)-(F)) v danom termíne, potom opakovane (vo vhodných časoch) vyzýva mailami predsedu o zadanie požadovanej informácie. Ak ani po primeranom počte opakovaní nebola zjednaná náprava, potom systém o tomto zverejní informáciu na web stránke a/alebo informuje mailom školiteľa (doktoranda), prípadne aj členov komisie.

1.2 Analýza požiadaviek

Ako základ aplikácie som navrhol webový portál, ktorého užívatelia sú najmä priamo zúčastnení na procese dizertácie, teda administrátor systému, predseda komisie

(čas ukáže pri testovaní aplikácie naživo, či tieto 2 funkcie budú mať 2 rôzni ľudia, alebo nie) a doktorandi. Administrátor systému a predseda komisie majú práva editovať informácie ktoré tento portál poskytuje a doktorandi majú možnosť skontrolovať priebeh dizertačného procesu na tomto portáli. Zvyšní zúčastnení buď môžu mať prístup k informáciám cez portál, alebo sú o dianí informovaní iba emailami. Predpokladám že sa tieto práva budú meniť počas skúšobného obdobia, pokiaľ sa nenájde najvhodnejšia kombinácia.

Tento portál musí byť jednoduchý na používanie, ponúknuť na jednom mieste potrebné informácie v čo najjednoduchšej forme a efektívne. Preto je vhodné spraviť rozhranie, ktoré by sa malo dať ovládať bez problémov intuitívne a neponúkať všetkým zúčastneným zbytočné funkcie, ktoré by ho zneprehľadnili. Neskôr spomeniem správu užívateľov, skupín, menu a stránok, ktoré toto nastavovanie uľahčujú.

1.3 Platforma, programovací jazyk

1.3.1 PHP + MySQL

Najskôr som premýšľal aké technológie použijem na implementáciu aplikácie. Vďaka diskusii s RNDr. Jaroslavom Janáčkom som sa rozhodol pre kombináciu PHP 5.x a MySQL 5.x kvôli nasledovným dôvodom:

- oboje sa používa na školských serveroch, nebude teda treba veľa konfigurácie a pri ďalšej údržbe a zmenách v systéme nie je problém nájsť človeka, ktorý ich ovláda
- oboje sú vo veľkej miere bežne používané, takže na prípadné problémy pri implementácii aplikácie sa dá vo vysokej miere a úspešnosti použiť vyhľadávanie na voľne dostupných zdrojoch na internete.
- sú voľne dostupné a majú otvorený kód
- vďaka obľúbenosti existuje množstvo voľne stiahnuteľných vývojových prostredí
- PHP 5.x oproti PHP verziám rady 4 dáva rozumnejšiu možnosť využitia sily objektovo orientovaného programovania, v tomto prípade hlavne prehľadnosti a následnej lepšej údržbe a rozšíriteľnosti aplikácie.

- dajú sa využiť na rôznych platformách, čo je vhodné vzhľadom k tomu, že školské servery bežia na systéme postavenom na platforme UNIX a ja som pôvodne chcel programovať aplikáciu vo Windows. Vďaka tomuto som mal možnosť pracovať na implementácii pod obidvoma systémami.

Proti tomuto výberu bola iba skutočnosť, že pred začiatkom vývoja aplikácie som mal nízku skúsenosť s tvorbou webovej aplikácie v PHP, ale to takisto aj v iných jazykoch používaných na tvorbu webových aplikácií a každopádne tvorbe muselo predchádzať naštudovanie si daného jazyka, takže v konečnom dôsledku PHP ako jazyk na tvorbu tejto aplikácie bolo pre mňa najvhodnejšie.

1.3.2 Smarty

Pri štúdiu som sa dostal k informáciám o šablónovacom systéme Smarty. Slúži na oddelenie funkcionality od dizajnu aplikácie, konkrétne funkcií aplikácie v PHP od zobrazovania výstupu pomocou HTML. Využívajú sa na to šablóny, ktoré pozostávajú z HTML kódu, pričom premenné z PHP sa posielajú pomocou Smarty šablónam, a z tých sú za behu aplikácie generované HTML stránky. Aj napriek použitiu ďalšieho systému, aplikácia je približne rovnako rýchla a čo je najdôležitejšie, zdrojový kód je naozaj prehľadnejší. Pri väčších projektoch je to vhodné aby sa ľudia mohli rozdeliť na programátorov a dizajnérov a robiť samotnú prácu efektívnejšie. Mne ako samotnému programátorovi aplikácie to aj tak uľahčilo prácu, lebo pri zmene vzhľadu som sa nemusel obávať zmeny funkcionality a teda som aj znížil výskyt chýb a celkovo je aplikácia vhodnejšia k údržbe a pridávaniu funkcionality.

1.3.3 Ostatné

Na zobrazovanie stránok používam validné XHTML a CSS na úpravu dizajnu stránky. Vzhľadom som sa snažil prispôsobiť aplikáciu momentálnemu dizajnu stránky Univerzity, čo sa mi viac-menej podarilo.

Na testovanie som používal prehliadače Opera a Firefox, aktuálne verzie a vyskúšal som aj Internet Explorer.

Napriek tomu že rozhranie aplikácie je v slovenskom jazyku, komentáre v kóde ako aj názvy funkcií a premenných som vytváral v anglickom jazyku tiež z dôvodu lepšej možnosti údržby a ďalšieho rozširovania.

Kapitola 2

2.1 Model aplikácie

Vzhľadom k použitiu PHP 5 je očakávateľné, že som využil výhody objektovo orientovaného programovania. Samotný *index.php* obsahuje iba niekoľko nastavení (cesty k priečinkom a premenné potrebné k nastaveniu pripojenia k databáze) a volanie triedy *application*. Celá funkčná časť aplikácie sa nachádza v priečinku *classes* a je rozdelená na niekoľko tried. Systém Smarty sa nachádza v priečinku Smarty, zdrojový kód Smarty šablón je uložený v priečinku *templates* a skompilované šablóny v priečinku *templates_c*. Autoreferáty dizertačných prác sa ukladajú do priečinka *file_storage* pod menami vygenerovanými aplikáciou.

2.2 Triedy aplikácie

2.2.1 class application

Najvyššia vrstva, obsahuje volania inštancií ostatných tried a inicializáciu databázy v triede MyDB. Klasický postup je, že skúsi vykonať štandardnú metódu každej triedy Process() a pokiaľ táto v poriadku prebehne, vykreslí výslednú stránku tiež štandardnou metódou Draw().

2.2.2 abstract class page

Obsahuje základné metódy *Process()* a *Draw()*, ktoré po nej dedia a ďalej rozvíjajú skoro všetky nasledujúce triedy. Takisto má konštruktor, ktorý do určitej miery štandardizuje vytváranie nadchádzajúcich potomkov a do určitej miery sprehľadňuje celý systém. Funkcia *CheckRights()* kontroluje či má prihlásený užívateľ právo na prehliadanie danej stránky a uľahčuje spravovanie bezpečnosti a pridávanie funkcionality (bližšie pri databázovom modeli). Umožňuje modularitu aplikácie, pokiaľ je nová trieda(modul) potomkom tejto triedy, už na začiatku obsahuje štandardné

metódy modulu a vývojár, ktorý nový modul pridáva má uľahčenú prácu, lebo aplikácia už vie procesovať nový modul.

2.2.3 class mainPage

Podľa hodnoty premennej *action:*

- *login* – zobrazí prihlasovaciu stránku, *mainPage* s premennou *action je* predvolená akcia, pokiaľ užívateľ nie je prihlásený (údaje uložené v *session*)
- *logout* – odhlási užívateľa zo systému (zruší premenné v *session*)
- *home* – po prihlásení zobrazí stránku s novinkami v systéme, ktoré sú uložené v databáze

2.2.4 class MyDB

Obsahuje základné metódy pre prácu s databázou a tým uľahčuje prácu s ňou, inicializuje sa vždy pri štarte aplikácie. Dôležité metódy sú:

- *Connect()* - pripojí sa k databáze s hodnotami zadanými v *index.php*
- *Disconnect()* - zruší spojenie s databázou
- *Query()* - vykoná dotaz na databázu, ktorý je argumentom tejto metódy a vráti výsledok
- *QueryToRow()* - ako v predošlom prípade, ale vráti výsledok vo formáte riadku
- *QueryToArray()* - vráti výsledok ako pole

2.2.5 class menu

Vytvorí položky menu na základe skupiny prihláseného užívateľa a zoznamu položiek menu s vhodnými právami uloženého v databáze. Základná metóda je *GenerateMenu(),* ktorá vráti položky menu vo formáte pre zobrazenie v Smarty šablóne

2.2.6 class changePassword

Zobrazí stránku, kde si prihlásený užívateľ môže zmeniť heslo. Pri zmene hesla sa pravdaže robí bežné overenie na zadanie starého hesla a potvrdenie nového.

2.2.7 class sendEmail

Poskytuje metódy na uľahčenie posielania emailov, parametry emailu(to, cc, subject, body, attachment ...) môžu byť poskytnuté podľa potreby buď metódou post HTML formuláru, alebo priamo volaním metódy *simpleSend()* s parametrami prijímateľ, mailová šablóna a príloha. Pre uľahčenie posielania mailov som vytvoril mailové šablóny, ktoré sú uložené v databáze. Metóda *simpleSend()* vytiahne podľa identifikačného čísla šablóny text subjektu a telo mailu. Telo mailu môže obsahovať aj premenné, v databáze sú v texte vyznačené ako *%VARIABLE%,* napríklad *%NAME%* alebo *%DATE%.* Pred poslaním mailu sa automaticky nahradia konkrétnymi hodnotami, pokiaľ ich metóda *simpleSend()* dostane ako argument v poli *$mailparams*.

2.2.8 class simpleFunctions

Obsahuje pomocné funkcie, ktoré v danom tvare programovací jazyk PHP neponúka, ale v danej aplikácii sa niekoľkokrát používajú a oddelením od zvyšného kódu sa zvyšuje prehľadnosť

- *addToStringArray()* - keďže SQL databáza neponúka dátový typ pole(array), bolo treba vyriešiť ako ukladať množinu prvkov do databázy. Štandardný prístup je, že pre každý prvok sa vytvorí nový záznam v databáze. Tabuľky sa preto delia na viacero s menším počtom stĺpcov, aby pri tomto nevznikalo veľa duplicitnej informácie a pri potrebe vytiahnutia informácie z databázy sa spájajú tabuľky systémom one-to-many. Vzhľadom k tomu že potrebujem ukladať ako polia z väčšej časti identifikátory, tieto sú uložené ako čísla oddelené delimitrom (napríklad zoznam 3 oponentov pozostáva z užívateľov s identifikačným číslom 4, 6, 11, tak výsledná hodnota uložená ako text v databáze je |4|6|11|). Táto funkcia dostane na vstup pole v tvare string s delimitrami a novú hodnotu. Tú pridá do poľa a vráti ho v rovnakom formáte ako je na vstupe.
- *removeFromStringArray()* - podobne ako v predošlom prípade, ale odstráni hodnotu z poľa ak sa tam nachádza a vráti pole v rovnakom formáte

2.2.9 class monitor

Táto trieda nie je prístupná žiadnemu užívateľovi, spúšťa sa z cronu, plánovača úloh pod systémami na báze UNIXu, ako sú školské servery. Spustí sa každý deň a skontroluje tabuľku úloh či sa nevyskytuje úloha ktorú treba vykonať. Vzhľadom k princípu tejto aplikácie, že všetko oznamovanie prebieha formou emailov, v tabuľke úloh sú uložené parametre emailu, dátum po ktorom sa má odoslať a či sa má vôbec odoslať (ak sa úloha splní – administrátor aplikácie, predseda komisie alebo iná osoba, ktorá má na to práva označí úlohu za splnenú, úloha sa označí za splnenú a email sa posielať nebude). Ak nájde úlohu ktorá sa má splniť v daný deň, automaticky odošle predpripravené emaily.

2.2.10 class modifyUser

Trieda obsahuje metódy modifikácie užívateľských kont. Všetky operácie majú pridelené Smarty šablóny. Úkony na zmenu databáz užívateľov sa vyberajú na základe parametra *action:*

- *addUser* – zobrazí formulár na zadanie parametrov užívateľského konta. Je tu na výber či užívateľovi vytvoriť aj konto na prihlásenie do systému, alebo sa zadajú iba informácie ako meno, priezvisko, email, sekcia (aby v jednom systéme mohlo byť viacero komisií pre rôzne odbory) a pracovisko. Je tu aj možnosť automatického poslania notifikačného emailu o vytvorení konta systéme s prihlasovacími údajmi na zadanú adresu, ktorá je štandardne zapnutá a dá sa v konkrétnom prípade vypnúť.
- *addUserToDB* – skontroluje zadané údaje a ak niečo nie je v poriadku, zobrazí formulár v *addUser* ešte raz aj s už vyplnenými údajmi a oznámením, kde sa chyba nachádza. V prípade že sú údaje v poriadku, zapíše do databázy a pošle notifikačný email.
- *listUsers* – vyberie základné informácie o užívateľoch z databázy a odošle ich šablóne
- *userDetails* – vytiahne z databázy všetky údaje o užívateľovi podľa identifikátora a odošle šablóne na zobrazenie, kde je možnosť údaje editovať a následne vyvolať jednu z nasledujúcich možností pre editovanie údajov užívateľa, alebo jeho zmazanie
- *delete* – zmaže užívateľa z databázy podľa identifikátora užívateľa

- *update* – zmení údaje existujúceho užívateľa podľa parametrov prijatých z formuláru a zapíše do databázy

2.2.11 class modifyThesis

Je štruktúrovaná podobne ako predošlá trieda na modifikáciu užívateľov. Predpokladám, že sa ešte bude meniť pri testovaní užívateľmi aby lepšie zohľadnila požiadavky na systém. Tiež je rozdelená na akcie:

- *addThesis* – zobrazí 2-krokový formulár na pridanie užívateľa, v prvom kroku zobrazí zoznam existujúcich sekcií a pole na zadanie názvu práce, pri odoslaní zavolá action *step1*
- *step1* – vyberie z databázy užívateľov podľa zadanej sekcie a zobrazí druhú časť formuláru, kde si užívateľ zúčastnených ľudí (doktorand, oponenti, komisia, …) vyberie zo zoznamov získaných z databázy užívateľov, dopíše ostatné údaje a prípadne upraví monitorovanie činností pre konkrétnu dizertáciu vybraného užívateľa. Po odoslaní formuláru sa zavolá *step2*
- *step2* - skontroluje zadané údaje a ak niečo nie je v poriadku, zobrazí pomocou akcie *step1* formulár znovu a oznámi ktorý údaj treba ešte upraviť. Ak sú údaje v poriadku, nasleduje *step3*
- *step3* - zobrazí možnosti pre nastavenie úloh monitorovacieho systému spolu s predvolenými hodnotami, po potvrdení zapíše údaje do databázy a vykoná potrebné kroky pre následné spracovanie úloh.

  listThesis – vyberie z databázy a pošle šablóne základné informácie o dizertáciách v databáze podľa sekcie. Podľa skupiny užívateľa zobrazí informácie pomocou šablóny s možnosťou editácie alebo bez. Ak je užívateľ predseda komisie, zobrazí sa možnosť označenia úlohy za splnenú a modifikácia úloh
- *update* – podľa hodnôt odoslaných z formulára aktualizuje informácie o práci v databáze, tuto patrí aj aktualizácia parametrov úloh
- *details* – vyberie z databázy informácie o dizertácii podľa jej identifikátora a zobrazí formulár s údajmi, ktoré sa tam dajú aj zmeniť, prípadne zrušiť celý záznam o práci
- *delete* – zmaže záznam o práci z databázy podľa identifikátora práce a príslušné úlohy k nej prislúchajúce

2.2.12 class thesisDetails

Táto trieda zobrazí informácie o konkrétnej práci ako aj o vykonaní, alebo nevykonaní jednotlivých úloh bez možnosti editácie, je určená hlavne doktorandom aby si mohli skontrolovať stav úloh, keďže nemajú prístup k triede *modifyThesis*.

2.2.13 class events

Obsahuje metódy na vykonanie činností pre monitorovanie prípravy obhajoby dizertácii. Aplikácia je naprogramovaná tak, aby sa ďalšie zmeny týkajúce sa monitorovania prác robili s veľkou pravdepodobnosťou iba v tejto triede, s pridaním referencie do ostatných a prípadnou minimálnou zmenou v niektorej inej triede.

Jednotlivé úlohy majú rôzne vstupy podľa potreby

2.2.14 class news

Poskytuje metódy na úpravu noviniek, ktoré sa zobrazujú pri prihlásení do systému. Podľa parametru *action*:

- *list* – zobrazí zoznam noviniek v systéme
- *add* – pridanie novinky
- *delete* – zmazanie novinky z databázy

2.3 Model databázy

2.3.1 Tabuľky

Model databázy je založený na 4 základných tabuľkách a niekoľkých pomocných. 4 základné sú tabuľka užívateľov DM_Users, prác DM_Thesis, užívateľských práv DM_PageRights a úloh DM_Events. Zvyšné slúžia ako pomocné pre tieto a na konfiguráciu aplikácie.

2.3.2 DM_Users

Sú tu uložené údaje o užívateľoch a to tých ktorí sa môžu na portál prihlásiť, ale aj tých ktorí sú iba notifikovaní o činnostiach. Má nasledovnú štruktúru:

- ID – unikátny identifikátor užívateľa v databáze, primárny kľúč
- LOGIN – prihlasovacie meno, nie je povinné, môže existovať užívateľ ktorý nemá prístup do systému
- PASSWORD – 32 znakový md5 hash prihlasovacieho hesla
- NAME – meno užívateľa
- SURNAME – priezvisko užívateľa
- EMAIL – email užívateľa
- LOCATION – pracovisko užívateľa, čisto informatívny charakter
- GROUP_ID – identifikátor skupiny práv do ktorej užívateľ patrí, referenčný kľúč k tabuľke DM_Groups
- SECTION_ID - identifikátor sekcie fakulty do ktorej užívateľ patrí, umožňuje mať v systéme nezávisle na sebe viacero komisií pre rôzne odbory na fakulte, referenčný kľúč k tabuľke DM_Sections

2.3.3 DM_Thesis

Obsahuje vložené informácie o práci, okrem informácií o stave vykonávania úloh, to je uložené v DM_Events

- ID – unikátny identifikátor práce v databáze, primárny kľúč
- NAME – názov práce
- DOKTORAND_ID – identifikátor doktoranda, referenčný kľúč k tabuľke DM_Users
- SECTION_ID - identifikátor sekcie fakulty do ktorej pridaná práca patrí podľa zaradenia, referenčný kľúč k tabuľke DM_Sections
- LECTOR_ID – identifikátor školiteľa, referenčný kľúč k tabuľke DM_Users
- ADDED – časový údaj o pridaní práce do zoznamu, od neho sa odvíjajú niektoré lehoty
- FILENAME – cesta k súboru s autoreferátom
- PROPOSED_OPPONENTS – pole s navrhnutými oponentami pre prácu, vo formáte aký som spomínal v sekcii 2.2.11

- ACCEPTED_OPPONENTS – pole s akceptovanými oponentami pre prácu, vo formáte aký som spomínal v sekcii 2.2.11
- COMMITTEE - identifikátor komisie, referenčný kľúč k tabuľke DM_Committee

2.3.4 DM_PageRights

- ID – unikátny identifikátor stránky, položky menu, alebo triedy v databáze, primárny kľúč
- PAGENAME - názov stránky, zobrazí sa ako názov odkazu, pokiaľ je objektom v menu
- PAGELINK – adresa linku, obsahuje reťazec, ktorý sa nachádza za *index.php* v adrese stránky, napríklad *?site=mainPage&action=home*
- MENU_ITEM – boolovská hodnota, označuje či daný záznam je objektom v menu
- GROUPS – množina identifikátorov skupín, ktoré majú právo prístupu na stránku v zadanom zázname
- POSITION – ak je záznam objektom v menu, určuje jeho poradie pre zobrazenie, čím vyššia hodnota, tým viac ku koncu menu

2.3.5 DM_Events

- ID – unikátny identifikátor úlohy v databáze, primárny kľúč
- EVENT_TYPE – identifikátor úlohy podľa jej typu, určuje ku ktorej akcii daný záznam patrí
- THESIS_ID – identifikátor práce ku ktorej daná úloha patrí, referenčný kľúč k tabuľke DM_Thesis
- MAIL_TO – zoznam adries prijímateľov emailu pre pole To:
- MAIL_CC – zoznam adries prijímateľov emailu pre pole Cc:
- MAIL_SUBJECT – subjekt emailu
- MAIL_BODY – telo emailu
- DATETIME – časová známka, znamená dátum kedy má byť úloha splnená
- STATUS – status úlohy, dosahuje 3 hodnoty:

    - 0 – pokiaľ nie je súčasný dátum väčší ako ten v DATETIME, je to v poriadku, úloha ešte nemusela byť splnená. Tento status je počiatočný po pridaní úlohy
    - 1 – značí nesplnenú úlohu, ak je súčasný dátum väčší ako ten v DATETIME, keď status je 0, odošle sa email a nastaví sa status 1. Tento status pretrváva kým sa úloha nesplní
    - 2 – značí splnenú úlohu, môže byť nastavený systémom alebo užívateľom s potrebnými právami. Keď je status 2, žiadna ďalšia akcia pre danú úlohu nie je potrebná.
- COMMENT – obsahuje text, ktorý sa zobrazí pri vylistovaní úloh k práci

2.3.6 DM_Groups

- ID – unikátny identifikátor skupiny v databáze, primárny kľúč
- NAME – názov skupiny, zobrazuje sa na stránke namiesto identifikátora pre sprehľadnenie

2.3.7 DM_Sections

- ID – unikátny identifikátor sekcie v databáze, primárny kľúč
- NAME – názov sekcie, zobrazuje sa na stránke namiesto identifikátora pre sprehľadnenie

2.3.8 DM_EmailTemplates

Obsahuje šablóny štandardných mailov, v body sa môžu nachádzať premenné v tvare %VARIABLE%, ktoré sú pred odoslaním nahradené hodnotami premenných daných mien, pokiaľ sú pridané ako parameter k funkcii na posielanie emailov.

- ID – unikátny identifikátor šablóny v databáze, primárny kľúč
- SUBJECT – subjekt emailu
- BODY – telo emailu
- COMMENT – komentár k šablóne

2.3.9 DM_Committee

Obsahuje informácie o zložení komisie

- ID – unikátny identifikátor komisie v databáze, primárny kľúč

- CHAIRMAN - predseda komisie, referenčný kľúč k tabuľke DM_Users
- MEMBERS – zoznam identifikátorov členov komisie vo formáte aký som spomínal v sekcii 2.2.11

2.3.10 DM_News

Tabuľka obsahuje novinky, správy ktoré sa zobrazujú na úvodnej stránke po prihlásení do systému

- ID – unikátny identifikátor novinky v databáze, primárny kľúč
- TITLE - titulok správy
- CONTENT – obsah správy
- AUTHOR – autor správy
- DATETIME – automaticky generovaná časová značka, novinky sa zobrazujú zoradené podľa tejto značky

Kapitola 3

3.1 Prehľad funkčnosti aplikácie

Samotná aplikácia slúži na správu informácií o procese prípravy obhajoby dizertačných prác a monitorovanie priebehu. Administrátor alebo predseda komisie pridá do systému prácu a prípadne prestaví niektoré parametre monitorovania z predvolených hodnôt na iné. Podľa nastavenia monitorovania systém od pridania práce kontroluje, či sú potrebné úlohy splnené v danom čase, pričom o ich splnení ho informuje predseda komisie. Systém zadané úlohy kontroluje automaticky na dennej báze a v prípade prekročenia limitu pre splnenie úlohy odošle notifikácie emailom zúčastneným osobám. Takisto denne informuje predsedu komisie zoznamom prípadných meškaní úloh aby sa na nič nezabudlo. Aplikácia podporuje viacero odborov a komisií na jednej inštancii aplikácie

Administrátor systému a predsedovia komisií majú možnosť upravovať zadané informácie a ostatní – doktorandi, školitelia, členovia komisie a oponenti si môžu vybrané informácie ktoré sa ich týkajú prehliadať na tomto portáli, alebo sú informovaní o udalostiach emailom.

3.2 Užívateľské rozhranie

Rozhranie aplikácie je výzorovo jednoduchá webová stránka, prispôsobil som ju momentálnemu dizajnu stránky Univerzity Komenského. Jednou z podmienok pri návrhu aplikácie bolo, aby rozhranie bolo jednoduché a prehľadné a aby užívateľovi poskytlo čo najväčší komfort pri užívaní. Preto som sa pokúsil spraviť túto aplikáciu s menším počtom stránok na zobrazenie, ale o to komplexnejším systémom v pozadí.

Tento portál obsahuje možnosti na vykonanie úloh zadaných v požiadavkoch na aplikáciu, ale napriek tomu je rozhranie užívateľsky príjemné a pri prvom použití sa užívateľ rýchlo zoznámi s funkciami. Vzhľadom k možnosti nastavenia práv na zobrazovanie stránok pre užívateľov v priebehu niekoľkých sekúnd v databáze,

nebudem písať ku ktorým stránkam má ktorá skupina užívateľov prístup, ale vymenujem dostupné stránky. V priebehu testovania budúci rok sa zistí či je tento zoznam dostatočný, alebo sa ešte dorobí webové rozhranie k niektorým ďalším nastaveniam.

3.3 Login

Obsahuje stránku s logom univerzity a formulárom na zadanie mena a hesla. Sem sa presmeruje užívateľ aj po odhlásení zo stránky. Neprihlásený užívateľ nemá k dispozícii žiadne vykonateľné akcie okrem prihlásenia.

3.4 Úvodná stránka

Po prihlásení sa užívateľovi zobrazí na ľavej strane menu, ktoré obsahuje položky podľa skupiny prihláseného užívateľa. Administrátor má v menu všetky položky. V strede stránky sa nachádza obsah zobrazenej stránky. Pri navigácii v stránke sa mení iba tento obsah, hlavička, pätička a menu na ľavej strane sa nemení.

V obsahu je odkaz na stránku so zmenou hesla, kde je jednoduchý formulár, kde treba zadať staré heslo, nové heslo a potvrdiť heslo. K zmene hesla má prístup každý užívateľ.

Ďalej sa tu nachádza panel s novinkami, každá novinka obsahuje titulok, nejaký obsah, autora novinky a kedy bola pridaná do systému. V menu sa nachádza položka Domov, ktorá odkazuje práve na túto stránku,

3.5 Info o práci

Stránka dôležitá hlavne pre doktorandov, tu sa zobrazia informácie o práci prihláseného doktoranda a v akom stave sú úlohy, ktoré sa majú splniť pre túto prácu. Samotnému doktorandovi by mala postačovať táto stránka, kde sa dozvie všetky potrebné informácie.

3.6 Správa užívateľov

Po kliknutí na túto položku menu sa zobrazia 2 možnosti – pridať užívateľa a zobraziť zoznam užívateľov. Pridanie užívateľa zobrazí formulár kde sa zadajú informácie o užívateľovi, dá sa vybrať či vytvoriť profil užívateľa aj s možnosťou prihlásenia, alebo bez. V druhom prípade systém bude komunikovať s užívateľom iba cez posielanie notifikačných emailov. Po pridaní užívateľa sa odošle na zadanú emailovú adresu mail s prihlasovacími údajmi a zobrazí sa možnosť pridať užívateľa, alebo zobraziť užívateľov.

Pri výbere zobrazenia sa zobrazí tabuľka so základnými informáciami o užívateľoch (každý užívateľ 1 riadok) obsahujúca meno, priezvisko, login, skupinu a sekciu do ktorej patrí, tabuľka sa dá abecedne zoradiť podľa ľubovoľného z týchto parametrov. Pri každom užívateľovi sú 2 akcie – detaily/upraviť, ktorá zobrazí formulár so všetkými údajmi užívateľa a umožňuje ich modifikovať, a zmazať, ktorá po potvrdení zmaže užívateľa.

3.7 Administrácia prác

Po kliknutí na túto položku menu sa zobrazia 2 možnosti – pridať prácu a zobraziť zoznam prác. Pridanie práce zobrazí 3-krokový formulár kde sa zadá v prvom kroku názov práce a do ktorej sekcie patrí. V druhom kroku sa objaví ďalší formulár, kde sa zadajú informácie o práci a užívatelia zúčastnení ako doktorand, oponenti, etc. sa zobrazia ako zoznam, z ktorého sa dajú vybrať. Po verifikácii správnych údajov sa zobrazí posledný formulár, kde sa dá prestaviť niektoré prednastavené hodnoty pre úlohy monitorovacieho systému.

Predseda komisie má možnosť zobraziť si zoznam prác a editovať informácie o konkrétnej práci ako aj vložiť informáciu o splnení úlohy. Podľa nastavenia si niektorí ďalší užívatelia môžu zobraziť zoznam prác s ktorými majú niečo spoločné – sú doktorandom, oponentom, školiteľom, alebo členom komisie danej práce a zobraziť detaily o práci.

3.8 Poslať email

Zobrazí formulár na poslanie emailu v mene systému, malo by byť prístupné iba administrátorovi a predsedom komisií.

Kapitola 4

4.1 Bezpečnosť všeobecne

Pri návrhu každej aplikácie treba samozrejme myslieť na bezpečnosť. Informácie o priebehu procesu prípravy na obhajobu dizertačnej práce sú samozrejme záležitosť, ktorá obsahuje kroky ku ktorým by sa nepovolaný človek nemal dostať. Preto som na bezpečnosť aplikácie myslel už pri návrhu a odzrkadlilo sa to v systéme užívateľských práv, práv na prístup k dátam a ich modifikácii. Takisto som ošetroval ukladanie hesiel a obranu voči známym druhom útokov na webovú aplikáciu, čo podrobnejšie rozpisujem v nasledujúcich stranách.

4.2 Užívateľské práva

Vzhľadom k faktu, že užívatelia aplikácie sa dajú rozdeliť do niekoľkých skupín, nestačil mi klasický model, keď existuje užívateľ s administrátorskými právami a zvyšok užívateľov majú rovnaké práva. Vyriešil som to vytvorením tabuliek v databáze tak, aby všetky nastavenia práv boli práve tuto ľahko modifikovateľné a nič nie je potrebné meniť v zdrojovom kóde aplikácie pokiaľ vyvstane potreba niektorej skupine pridať alebo odňať práva na prístup k niektorému modulu aplikácie.

Databáza obsahuje tabuľku grúp (skupín), ktorá sa dá ľubovoľne zmeniť podľa potrieb aplikácie, tu je uložený identifikátor skupiny a jej názov, ktorý sa vyskytuje pri manipulácii s údajmi na stránke. V tabuľke užívateľov má každý užívateľ pridelenú skupinu, ktorú môže adminstrátor aplikácie hocikedy zmeniť cez webové rozhranie. Najvýznamnejšia je tabuľka *DM_PageRights,* kde sa ukladajú informácie o existujúcich triedach aplikácie, či odkaz na ňu má byť v menu pre užívateľa a v akom formáte, ale hlavne množina skupín, ktoré majú k danej stránke prístup.

Volanie konkrétnej triedy, ako aj akcia s ňou spojená sa predávajú aplikácii pomocou GET parametrov a teda sa zobrazujú aj v adrese stránky (napr. *index.php?site=mainPage&action=home*). To zvádza k možnosti prepísať argument a zavolať

akciu, na ktorú užívateľ nemá práva. Avšak to je zabezpečené kontrolou práv v databáze pri volaní inicializácie každej triedy podľa skupiny užívateľa uloženej v session a teda vylúčené.

4.3 MD5

MD5 (Message-Digest algorithm 5) je rozšírená kryptografická hashovacia funkcia so 128 bitovou hashovou hodnotou. MD5 hash sa zvyčajne reprezentuje ako 32 ciferné hexadecimálne číslo.

MD5 hashovanie sa používa pri ukladaní hesla do databázy v tejto aplikácii. Takisto pri ukladaní súborov autoreferátu bolo potrebné zabezpečiť aby nevznikali konflikty medzi názvami prác (dvaja doktorandi nazvú tento súbor napríklad *autoreferat.pdf*). To je zaistené tým, že pred uložením práce sa vygeneruje pre každého užívateľa unikátny hashový reťazec a ten sa stane prefixom názvu súboru na disku.

4.4 HTML injection

Je to útok za využitia vstupov HTML formulárov, kde užívateľ vloží vlastný HTML, alebo JavaScript kód a tým umožní vykonať neželanú akciu, napríklad odoslanie údajov na inú stránku, presmerovanie alebo inú neželanú akciu. Tento útok sa často používa na rôzne webové fóra.

Ochrana pred týmto typom útoku spočíva v prekonvertovaní každého vstupu zo stránky za použitia escape characters ("<" na "<", ">" na "&rt;", "&" na "&",…). PHP poskytuje funkciu *htmlspecialchars(),* ktorá vykoná práve vyššie spomenuté.

Vzhľadom k tomu, že všetky stránky sa spracúvajú cez triedu *application*, všetky parametre GET a POST sú ošetrené v tejto triede pomocou funkcie *htmlspecialchars()*.

4.5 SQL injection

Tento útok na databázu spočíva v zadaní textu, ktorý zmení logiku databázového dotazu a útočník môže získať informácie ku ktorým sa nemal dostať, alebo zmeniť údaje v databáze ku ktorým nemá mať prístup. Dokonca je možné aj zmeniť štruktúru tabuliek, alebo tabuľku vymazať. Uvediem 2 príklady:

*$query = "SELECT * FROM users WHERE name = '" + $userName + "';"*

Tento dotaz má vytiahnuť z databázy údaje o užívateľovi, ktorého *name* je hodnota premennej *$userName.* Pri vhodnom zadaní tejto hodnoty môže ale tento dotaz spraviť niečo iné. Napríklad ak zadá hodnotu *$userName* ako *a' OR '1'='1 ,* výsledný dotaz bude vyzerať takto:

*$query = SELECT * FROM users WHERE name = 'a' OR '1'='1';*

Keďže *'1'='1'* platí vždy, daný dotaz vráti informácie o všetkých užívateľoch. Aj keby sa nezobrazili všetky údaje, dotaz vráti prvého užívateľa v tabuľke, čo zvyčajne býva administrátor.

Nasledovná hodnota *$userName* zmaže tabuľku užívateľov a vytiahne informácie z inej tabuľky. Pravdaže útočník musí správne uhádnuť názvy tabuliek, čo nebýva problém ak je špecifikácia programu voľne dostupná ako v prípade tejto práce.

*$userName = a';DROP TABLE users; SELECT * FROM data WHERE name LIKE '%*

*$query = SELECT * FROM users WHERE name = 'a';DROP TABLE users; SELECT * FROM DATA WHERE name LIKE '%';*

Podobne ako pri HTML injection, v PHP existuje funkcia *mysql_real_escape_string(),* ktorá preformátuje vstup tak, že nebezpečné znaky ako úvodzovky sú vo forme, kde sa berú iba ako text a nie ako príkaz (napríklad vložením \ pred znaky uvodzovky). Znovu je to vyriešené globálne v triede application.

Záver

Navrhol som a vytvoril aplikáciu na monitorovanie prípravy obhajoby dizertačnej práce. Dal som si záležať hlavne na tom, aby som využil možnosti, ktoré mi programovací jazyk PHP5 ponúka, teda celá aplikácia je objektovo orientovaná, pričom som sa ju snažil spracovať tak, aby bola ľahko udržovateľná, nastaviteľná a rozšíriteľná.

Osobne dúfam, že bude ľuďmi pre ktorých bola vytvorená prijatá pozitívne a po prvom testovacom roku bude v takom štádiu, že bude naozaj prácu uľahčovať a zabráni nepríjemnostiam, ktoré sa občas nechcene stávajú kvôli ľudskej zábudlivosti a iným faktorom.

Odovzdaním práce sa jej vývoj neskončil, sám mám ešte niekoľko nápadov na vylepšenie a predpokladám že ďalšie návrhy vyplynú pri testovaní v budúcom roku. Základ tejto aplikácie sa dá použiť po ďalšom rozšírení nielen na monitorovanie prípravy obhajoby dizertácie, ale aj ako monitorovací a notifikačný systém pre iné, všeobecnejšie účely na fakulte a práve týmto smerom vidím možnosť ďalšieho vývoja.

Zoznam použitej literatúry

[1] PHP: Hypertext Preprocessor. **http://www.php.net/**

[2] Smarty : Template Engine. **http://www.smarty.net/**

[3] MySQL :: MySQL 5.0 Reference Manual

http://dev.mysql.com/doc/refman/5.0/en/

[4] Wikipedia **http://www.wikipedia.org/**

Prílohy

K elektronickej verzii tohto dokumentu je priložená samotná aplikácia v stave pred nastavením podľa potrieb fakulty a pred užívateľským testom. Budúci školský rok by nastavená aplikácia mala bežať a testovať sa na fakultnom serveri.