KATEDRA INFORMATIKY
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO, BRATISLAVA

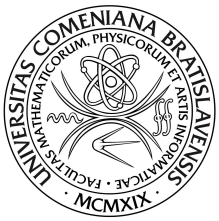
PROGRAMOVÝ SYSTÉM NA MONITOROVANIE PRÍPRAVY OBHAJOBY DIZERTÁCIE

(Bakalárska práca)

JAKUB HUSÁR

Vedúci: doc. RNDr. Pavol Ďuriš, PhD.

Bratislava, 2009



KATEDRA INFORMATIKY
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO, BRATISLAVA

PROGRAMOVÝ SYSTÉM NA MONITOROVANIE PRÍPRAVY OBHAJOBY DIZERTÁCIE

(Bakalárska práca)

JAKUB HUSÁR

Študijný odbor: Informatika 9.2.1

Vedúci: doc. RNDr. Pavol Ďuriš, PhD.

Bratislava, 2009

Čestné prehlásenie

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

. . . . . .

Poďakovanie

Ďakujem môjmu školiteľovi doc. RNDr. Pavlovi Ďurišovi, PhD. za pomoc pri výbere témy a cenné informácie ohľadom priebehu prípravy obhajoby dizertačných prác. Ďalej ďakujem RNDr. Jaroslavovi Janáčkovi za pomoc pri implementácii aplikácie do fakultného systému a taktiež svojim najbližším za trpezlivosť počas písania práce.

Abstrakt

Autor: Jakub Husár

Názov bakalárskej práce: Programový systém na monitorovanie prípravy obhajoby dizertácie

Škola: Univerzita Komenského v Bratislava

Fakulta: Fakulta matematiky, fyziky a informatiky

Katedra: Katedra informatiky

Vedúci bakalárskej práce: doc. RNDr. Pavol Ďuriš PhD.

Rozsah práce: 29 strán

Bratislava, 2009

Cieľom bakalárskej práce, ktorá nadväzuje na bakalársku prácu Aplikácia na monitorovanie prípravy obhajoby dizertácie písanú študentom Martinom Biesom v školskom roku 2007/08, je vývoj a implementácia webovej aplikácie na monitorovanie prípravy obhajoby dizertačných prác. Táto aplikácia je určená pre osoby priamo zúčastnené na tomto procese a uľahčuje im vykonávanie a sledovanie priebehu plnenia jednotlivých schvaľovacích procesov, potrebných pre prikročenie k samotnej obhajobe dizertácie. Práca sa venuje otázkam jeho vývoja a implementácie, pričom dôraz je kladený predovšetkým na popis použitých tried, databázového modelu a otázkam ohľadom bezpečnosti a stability systému.

KĽÚČOVÉ SLOVÁ: monitor prípravy obhajoby dizertácie, webová aplikácia, PHP, SQL

Obsah

1 Úvod **1**

2 Zadanie **3**

2.1 Zadanie požiadaviek 3

2.2 Spôsob realizácie požiadaviek 5

2.3 Role a ich funkcionalita 6

3 Vývoj **7**

3.1 Rozdelenie aplikácie na triedy 7

3.2 Použitie skriptu monitor.php 11

3.3 Databázový systém 12

4 Realizácia **18**

4.1 Použité technológie 18

4.1.1 XHTML, CSS a UTF-8 18

4.1.2 Databázový systém 19

4.1.3 Skriptovací jazyk 19

4.1.4 Templatový systém 19

4.2 Základný popis vzhľadu prostredia 19

4.2.1 Vzhľad a funkcionalita prihlasovacej stránky 20

4.2.2 Plavná stránka 21
4.3 Testovanie systému 21

5 Bezpečnosť systému **22**
5.1 Základy bezpečnosti 22
5.1.1 Autentifikácia užívateľov 22
5.1.2 Prístupové práva užívateľov 23
5.2 Útoky na aplikačnú vrstvu 23
5.2.1 Injection 24
5.2.2 Cross-site scripting 25
5.2.3 Session stealing 26

6 Záver **27**

Literatúra **28**

Prílohy **29**

Kapitola 1

Úvod

Informačné technológie sa stali pevnou súčasťou našich životov a aj preto sa ľudia v mnohých oblastiach intenzívne zaujímajú, akým spôsobom využiť výhody, ktoré so sebou prinášajú. Prostredníctvom nich máme možnosť uľahčiť si viaceré netriviálne činnosti. Proces prípravy obhajoby dizertačných prác je jednou z takýchto činností a preto sa tu ponúka možnosť využitia webovej aplikácie, ktorá by dopomohla, aby si zúčastnení v procese obhajoby dizertačnej práce vykonali svoje povinnosti jednak načas a na druhej strane pokiaľ možno čo najefektívnejšie.

Systém na monitorovanie prípravy obhajoby dizertácie uľahčuje prácu predsedovi obhajovacej komisie tým, že mu je nápomocný pri rozosielaní mailov o jednotlivých úlohách, automatickým dohliadaním na ich včasné plnenie, evidenciou o ich úspešnom splnení a v neposlednom rade prehľadným zobrazovaním informácií o prácach a úlohách k nim prislúchajúcich. Doktorandom a školiteľom umožňuje touto formou sledovať v akom štádiu príprav sa nachádza ich práca, zatiaľ čo oponentov a členov obhajovacej komisie informuje prostredníctvom predpripravených mailov o tom, či nemajú v najbližších dňoch vykonať nejakú povinnosť. V skratke povedané, poskytuje všetkým zúčastneným stranám práve tie informácie, ktoré sú potrebné na úspešné skompletizovanie procesu prípravy obhajoby dizertácie.

Predložená práca nadväzuje na bakalársku prácu Aplikácia na monitorovanie prípravy obhajoby dizertácie študenta Martina Biesa v školskom roku 2007/08. Cieľom bakalárskej práce je vývoj a implementácia programového systému na monitorovanie prípravy obhajoby dizertačných prác. Zaoberám sa v nej komplexným programovým systémom evidencie a správy

dizertačných prác. Zameriavam sa na analýzu požiadaviek, popis použitého databázového modelu, na použité triedy programového systému a bezpečenosť vlastnej aplikácie.

Práca pozostáva z úvodu, záveru, 4 hlavných kapitol, ako aj prílohy, ktorej súčasťou je CD so samotnou aplikáciou.

Sformulované zadanie práce od Doc. RNDr. Pavla Ďuriša, PhD. a analýza požiadaviek na systém z neho vyplývajúcich bude rozobraná v kapitole 2.

V nasledujúcej kapitole opíšem jednotlivé triedy aplikácie s dôrazom na zmeny v nich uskutočnené. Ďalej popíšem základný skript, ktorý používam na monitorovanie včasného plnenia úloh v procese obhajoby. Nakoniec rozoberiem implementovaný model databázového systému.

V kapitole s názvom Realizácia v skratke predstavím technológie, ktoré boli použité na vývoj a implementáciu celého projektu. V druhej časti opisujem základný vzhľad použitý v prostredí systému a napokon uvediem spôsob akým som testoval tento systém.

Piata kapitola je zameraná na popis najznámejších bezpečnostných rizík a to, akými formami sa im snažíme v aplikácii predchádzať.

V závere práce sa vraciam k problémom, na ktoré som pri vývoji systému narazil a stručne popisujem ďalšie možnosti rozšírenia systému.

Kapitola 2

Zadanie

Nakoľko sa aplikácia, ktorou sa v tejto práci zaoberám, venuje monitorovaniu príprav obhajoby dizertácie, čo pre mňa ako laika bol zozačiatku neznámy proces, bolo potrebné poznať kroky uskutočňované v štandardnom prípravnom procese a to, v akej súslednosti na seba sa tieto kroky vykonávajú. Toto všetko bolo nutné, aby som mal prehľad, akým smerom aplikáciu vyvíjať. Rovnako tak bolo treba poznať alternatívy, akými sa rieši situácia, keď sa úloha v rámci procesu príprav dostane do časového meškania. Preto bolo potrebné sformulovať základné body od ktorých by sa odvíjal vývoj systému. Tieto body boli zhrnuté do nasledovných požiadaviek.

2.1 Zadanie požiadaviek

Doktorand odovzdá dizertačnú prácu a administrátor do monitorovacieho systému uloží nasledujúce 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 po pridaní práce do systému

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 10

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.

2.2 Spôsob realizácie požiadaviek

Na základe požiadaviek pre systém, popísaných v predchádzajúcej sekcii, môžeme prejsť k samotnej analýze týchto požiadaviek a to, akým spôsobom ich budeme realizovať. Zo zadaných požiadaviek nám vyplýva, že v procese prípravy obhajoby dizertácie rozlišujeme 5 rôznych rolí, podľa toho, ktorá osoba akú funkciu v danom procese vykonáva. Týchto 5 rolí je doplnených rolou, ktorá všetky údaje spravuje - administrátorom.

Nato, aby bola aplikácia ľahko spravovateľná je vhodné každého zúčastneného na príprave obhajoby dizertácie zvlášť evidovať v rámci záznamov databázy. Preto samotnému procesu pridávania dizertácie do systému bude predchádzať ich zaznamenanie. Po zaznamenaní všetkých osôb spojených s konkrétnou dizertáciou je zvlášť vhodné, evidovať záznamy o fungujúcich obhajovacích komisiách. Preto administrátor v sekcii pre správu komisii zoskupí pridaných užívateľov, ktorí budú ďalej vystupovať ako celok - jedna komisia.

Až následne je správcom cez rozhranie pre administráciu prác pridaná nová práca do systému. Tu už nemusia byť vypĺňané jednotlivé záznamy o osobách, ale tie budú automaticky selektované zo záznamov priradených používateľov. Ďalej podľa konzultovaných informácii, môže jedna osoba zastávať v rámci procesov obhajoby viacero funkcii a tak namiesto toho, aby bol administrátorom zakaždým vytváraný nový používateľ s novo vytvorenou rolou, môže ich vrámci jedného užívateľa zastávať hneď niekoľko. Tieto budú vrámci evidencie užívateľov reprezentované množinou všetkých rolí, ktorých právomoci sú mu budú následne pridelené.

Po zaevidovaní práce tak ako je to popísané v požiadavkách sa k hlavnému slovu dostáva

samotný proces prípravy, ktorý je v požiadavkách detailne popísaný. Ten bude odštartovaný automatickým poslaním mailovej notifikácie predsedovi komisie, ktorý preberá zodpovednosť za jej ďalší priebeh. Všetky jeho kroky systém eviduje a zaznamenáva prostredníctvom databázy. V rámci tejto prípravy sa ale samotné úlohy zadávané predsedom komisie budú môcť dostať do časového sklzu. V takom prípade bude na odbremenenie predsedu skúmať túto skutočnosť pravidelne sa spúšťajúci skript, upozorňujúci na novovyskytnuté udalosti predsedu komisie prostredníctvom vopred predpripravených mailov.

2.3 Role a ich funkcionalita

Užívateľ v role administrátora bude mať na starosti chod celého systému. To značí mať kontrolu nad správou všetkých užívateľov, monitorovaním dizertačných prác, zložení jednotlivých komisii, ale aj možnosť nastavovať systém pridávaním a modifikáciou aktuálnych noviniek, vrátane praktickej možnosti zmeny formátu mailových predlôh, ktoré sa môžu postupom času preformulovať bez nutnosti pristupovania k štruktúram databáz. Všetky tieto možnosti budú jednoducho štrukturované a zoskupené do blokov podľa kategórii.

Ďalšou rolou je rola predsedu komisie, ku ktorej bude priradená možnosť prezerania si zoznamu doktorandov s prácami, ktoré boli pridelené pod jeho kompetenciu. Pri takýchto prácach si bude môcť vykonávať svoje právomoci a ako jediný z užívateľov zadávať úlohy pre oponentov a členov komisie. Aby sme predsedovi uľahčili jeho úlohu v systéme,

V prípade role doktoranda a role školiteľa ide o užívateľov s veľmi podobnými možnosťami. Rozdiel je iba v tom, že zatiaľ čo doktorandovi sa umožňuje sledovanie postupu prác iba vlastnej dizertácie, v prípade školiteľa sa to týka všetkých prác v rámci ktorých figuruje ako vedúci.

Poslednými rolami sú užívatelia zo skupín oponentov a členov komisie. Títo na rozdiel od predchádzajúcich užívateľov nemajú aktívny prístup, ale sú aktívnymi členmi každého z procesov prípravy obhajoby a záleží aj od ich prístupu, či bude proces úspešne skompletizovaný.

Kapitola 3

Vývoj

3.1 Rozdelenie aplikácie na triedy

Aby bol systém na jednej strane prehľadný a na druhej strane využil výhody objektovo orientovaného programovania, boli jeho jednotlivé časti rozdelené do samostatných celkov - tried. V nasledujúcej časti rozoberiem najdôležitejšie z nich a metódy v nich definované.

application

Trieda *application* je takpovediac akýmsi prepojením medzi základným súborom *index.php*, ktorý je volaný po každej akcii vo vnútri systému a samotnými triedami ktoré v aplikácii využívame. Úlohou tejto triedy je vykonať inicializáciu na požadovanej stránke. Najdôležitejšou metódou triedy je *GetPageById*, ktorá v rámci inicializácie dohliada na bezpečnosť systému, tým že ošetruje prenesené dáta prostredníctvom URL adresy a formulárov. Rovnako tak má funkciu dohliadať na korektne zadané adresy stránok a tie nepresne zadané v rámci bezpečnosti presmerovať na prihlasovaciu stránku systému.

page

Základ celej aplikácie tvorí abstraktná trieda *page*, od ktorej je odvodená veľká väčšina tried v systéme. Okrem štandardného konštruktora, priradzujúceho zvolené hodnoty globálnym

premenným obsahuje metódy *SayError* a *SayReport*, ktoré upozorňujú užívateľa na výsledky vykonávaných akcií, kde v prvom prípade sú to chybové upozornenia a v tom druhom oznámenia o úspešnom vykonaní akcie. Ďalšou veľmi dôležitou metódou pre beh celého systému je *CheckRights*, ktorej úlohou je kontrolovať užívateľské práva pre prístup k jednotlivým stránkam.

Trieda obsahuje v neposlednom rade metódu *Draw*, ktorej hlavnou úlohou je vykresľovať obsah volaných stránok. Vykresľovanie sa delí na 2 časti: vykreslenie statickej časti, rovnakej pre všetky stránky a obsahu, pre každú stránku unikátny.

mainPage

Trieda využívaná na kontrolu autentifikácie užívateľov. V rámci nej sú vykonané všetky potrebné náležitosti pri prihlásení užívateľa akým je napríklad vytvorenie sedenia a priradenie najpoužívanejších hodnôt do superglobálnej premennej $_SESSION. Rovnako tak je z hľadiska na bezpečnosť toto sedenie prostredníctvom tejto triedy aj rušené, po tom, čo užívateľ zašle požiadavku na odhlásenie. Táto trieda je taktiež okrem vyššie spomenutej funkcionality využívaná aj na zobrazovanie úvodnej stránky systému.

menu

Táto trieda má za úlohu vygenerovať menu prislúchajúce momentálne prihlásenému užívateľovi. Ide o triedu neinicializovanú prostredníctvom triedy *Application*, ako v prípade veľkej časti iných tried, ale vytvorenú v rámci volania konštruktora tried, ktoré majú v popise práce vyobrazovať obsah v rámci aplikácie a tým pádom zobrazovať aj príslušné menu na bočnom panely.

navigation

Pre uľahčenie orientácie v rámci systému pomocou tejto triedy volaním metódy *Generate-Navigation* generujeme navigáciu v systéme, ktorá sa vytvára na základe momentálne zobrazenej stránky. Táto trieda sa taktiež vytvára v rámci volania konštruktora tried, ktoré využívajú metódu *Draw* na vykresľovanie stránok a tým pádom aj vykresľovanie navigácie v systéme.

modifyUser

Veľmi dôležitá trieda pre správcu systému na administráciu užívateľov. Poskytuje všetky potrebné možnosti pre prácu s nimi počnúc pridávaním nových užívateľov, zobrazovaním ich kompletného zoznamu s možnosťou prezerania si ich detailov, prípadnú modifikáciu a možnosť zmazania.

V prípade zoznamu užívateľov bola pre zlepšenie orientácie správcu systému okrej klasického zoznamu vytvorená aj možnosť selekcie v rámci jednotlivých odborov a užívateľských skupín. Rovnako tak je k dispozícii vzostupné a zostupné zoraďovanie podľa jednotlivých atribútov vyobrazeného zoznamu.

Aby sa nemohlo dospieť napríklad do situácie, že predseda komisie ide odosielať mail s informáciami členom komisie a jeden z nich bol medzi časom vymazaný z evidencie a tak by dochádzalo k nekompatibilite dát, je v prípade zmazávania užívateľa v rámci metódy *delete* spravená kontrola na aktívnu činnosť daného užívateľa, ktorá buď dané zmazanie odobrí, alebo naopak nepovolí.

modifyThesis

Taktiež jedna z dôležitých tried, ktorá je určená pre správcu systému na administráciu dizertačných prác a pre predsedu komisie pre poskytnutie prehľadu o pridelených prácach a možnosti evidencie vykonávania úloh na nich. Správcovi poskytuje všetky potrebné nástroje pre správu akými sú pridávanie nových prác, zobrazovanie ich kompletného zoznamu s možnosť prezerania si ich detailov, až po možnosť zmazania už neaktuálnych prác.

Rovnako ako v prípade zoznamu užívateľov aj pri zozname prác je pre správcu na lepšiu prehľadnosť vytvorená možnosť selekcie a zoraďovania prác v rámci zobrazovaného zoznamu.

Čo sa týka predsedu komisie, tak on, prostredníctvom akcie s názvom *request* zadefinovanej v tejto triede, môže zadávať informácie o plnení úloh na jemu pridelených prácach, ktoré sa následne vyhodnocujú v rámci triedy *events.*

events

Táto trieda má prostredníctvom metódy *makeInitEvent* za úĺohu vytvárať prvotné záznamy o úlohách po tom, čo bola dizertačná práca zadaná do systému. Ako ale bolo spomenuté už pri

triede *modifyThesis*, trieda *events* slúži prevažne na spracovávanie informácii o plnení úlohy zadávaných predsedom komisie.

sendMail

Táto trieda slúži na odosielanie notifikačných mailov adresovaných jednotlivým užívateľom. Zároveň ju využíva predseda komisie pri odosielaní zadaní jednotlivých úloh, kedy sa volá akcia *sendRequest*, ktorá predvyplní formulár prichystaný na odoslanie daných informácii.

modifyCommittee

Tak ako aj v prípade ostatných tried určených na správu informácii o niektorej zo zložiek systému, aj v tejto triede má administrátor systému možnosť spravovať zloženia obhajovacích komisií. Znamená to, že prostredníctvom tejto triedy je administrátorovi povolené pridávanie nových zložení komisií a v prípade, že dané zloženie komisie je už neaktuálne zmazanie tejto komisie.

modifyNews

Vďaka tejto triede má administrátor systému možnosť spravovať novinky zobrazujúce sa na hlavnej stránke, čo znamená, že má k dispozícii akcie ako pridávanie, úprava, mazanie.

modifyTemplate

V rámci inštalácie systému sú v databáze prednastavené formy odosielaných mailov. Prostredníctvom tejto triedy má administrátor možnosť úpravy jednotlivých z nich a tak ich prispôsobiť požiadavkám predsedov komisií.

errorPage

Táto trieda bola vytvorená za účelom vyriešenia situácie, keď sa vyskytuje žiadosť na systém o vytvorenie neexistujúcej stránky. V takomto prípade je vytvorená inštancia tejto triedy a vykoná sa metóda Process v rámci nej, ktorá rieši danú situáciu.

3.2 Použitie skriptu monitor.php

V tomto prípade už nejde o triedu, ale o nezávislý kľúčový skript, zabezpečujúci dodržanie jednej zo základných požiadaviek na systém, ktorou je monitorovanie dodržovania termínov stanovených pri jednotlivých úlohách predsedom komisie. Aby sme mohli toto zabezpečiť bez nutnosti každodenného pristupovania do systému, je potrebné nájsť spôsob akým by sa toto pristupovanie nahradilo. Zo základných znalostí linuxovských systémov, na ktorých bežia aj fakultné servery, vieme, že tu existuje spôsob ako tento problém vyriešiť. Je ním použitie programu *cron*, ktorý poskytuje možnosť spúšťať skripty, príkazy či skupinu príkazov automaticky a pravidelne v stanovenom čase. V našom prípade by sme chceli docieliť, aby sa v jednodňových intervaloch spúšťal skript *monitor.php*. Toto vieme zrealizovať pridaním nasledovného riadku do tabuľky cronu:

*30 7 * * * /usr/local/bin/php -q /docmonitor/classes/monitor.php*

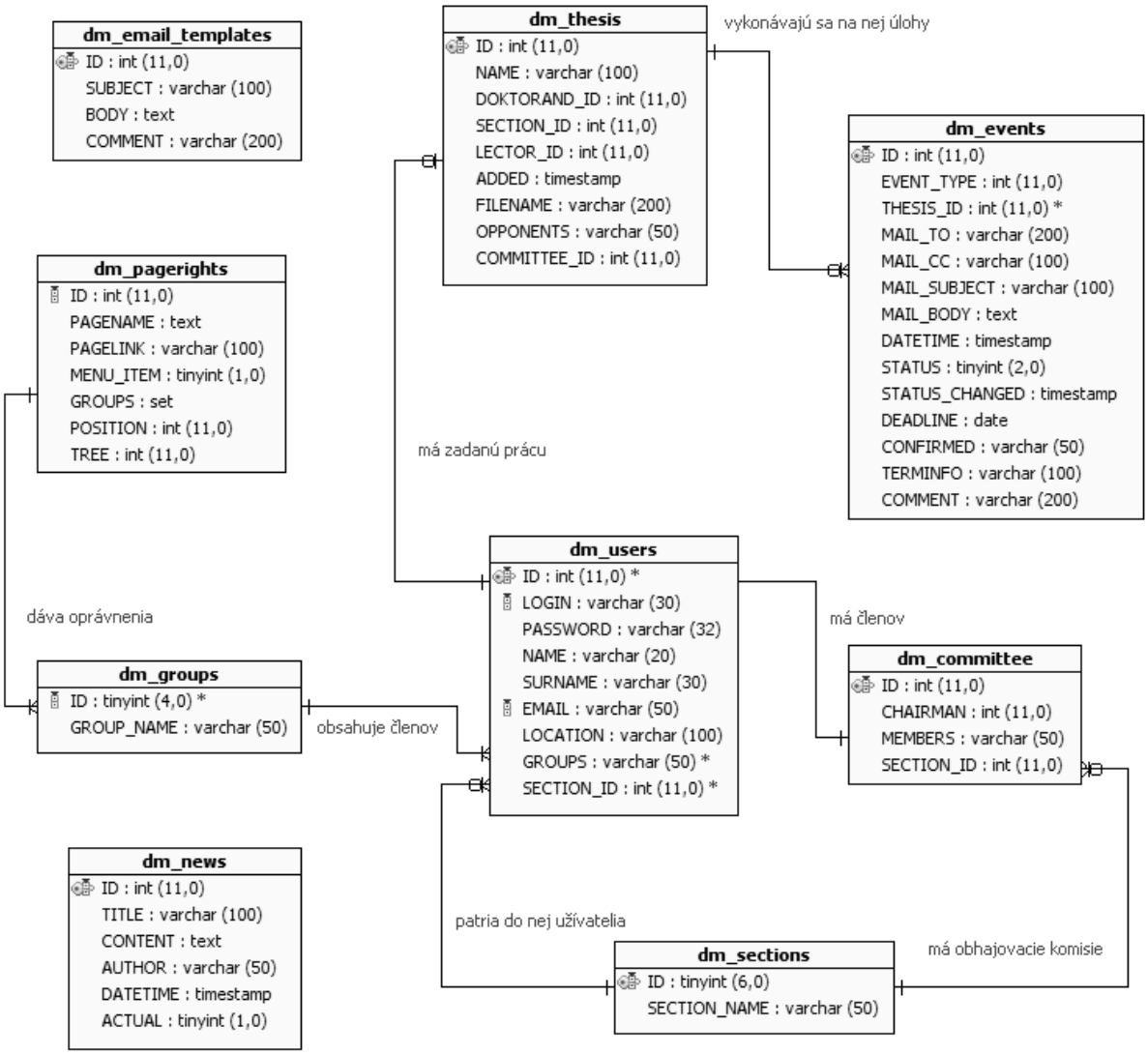
kde hovoríme, aby sa každý deň o 7:30 ráno vykonal skript *monitor.php* pomocou interpretera zo zadanej cesty.

A o čom vlastne tento skript je?

Ako bolo už spomenuté, má na starosti monitorovanie dodržiavania termínov úloh, zadaných predsedom komisie. Preto hneď na začiatku skriptu prechádzame všetky zaznamenané úlohy v tabuľke *dm_events* a selektujeme všetky položky ktoré majú status rovný 1, čiže tie, ktoré sú v stave čakania na ich splnenie. V rámci týchto vyberieme iba tie, ktorých položka *DEADLINE,* reprezentujúca termín na splnenie úlohy, predstavuje dátum, ktorý bol už medzičasom prekročený. V prípade, že niektorá z úloh spĺňa tieto podmienky a teda je evidovaná ako po termíne, je potrebné, aby sme poslaním mailu upozornili na túto skutočnosť predsedu komisie, pripravili mail na prípadné urgovanie osôb, ktoré si danú úlohu nesplnili a zmenili status v danej úlohe na 2, čiže úlohu po termíne. Tieto skutočnosti sú naprogramované v rámci tohto skriptu a tým pádom dávajú záruku, že omeškania v procese príprav obhajoby dizertácie budú včasne systémom rozpoznané a kompetentné osoby budú upozornené.

3.3 Databázový systém

Veľmi dôležitým pilierom webovej aplikácie je do veľkej miery správne navrhnutá štruktúra databázy. Návrh použitý v systéme pozostáva z nasledovných 9 tabuliek a vzťahmi medzi nimi:



Na nasledujúcich riadkoch sa budem venovať popisu jednotlivých tabuliek a atribútov

Tabuľka dm_users

V rámci tejto tabuľky udržiavame informácie o všetkých užívateľoch systému.

ID primárny kľúč, ktorý je zároveň unikátnym, na základe ktorého rozlišujeme medzi jednotlivými užívateľmi a jednoznačne nám predstavuje konkrétnu osobu v systéme

LOGIN reťazec, ktorý predstavuje prihlasovacie meno užívateľa; taktiež som ho stanovil za unikátny nakoľko keby dvaja užívatelia s rovnakým loginom si čisto teoreticky zvolili rovnaký login, nastal by problém v tom, že užívateľ s vyšším ID by sa automaticky prihlásil na konto druhého užívateľa s nižším ID

PASSWORD tento atribút v sebe udržiava hashovaciu hodnotu užívateľovho hesla; na hashovanie bola zvolená funkcia MD5

NAME, SURNAME meno a priezvisko konkrétneho užívateľa

EMAIL tento atribút uchováva užívateľovu emailovú adresu; aj v tomto prípade som uvážil za vhodné dať tento atribút ako unikátny, nakoľko podľa mňa nie je pravdepodobné, aby 2 rôzny užívatelia využívali tú istú emailovú adresu

LOCATION informácia o pracovisku užívateľa(nepovinný atribút)

GROUPS reťazec zložený z ID skupín, do ktorých je zaradený užívateľ; jednotlivé ID sú vzájomne oddelené zvislou čiarou

SECTION_ID ID odboru, ku ktorému patrí daný užívateľ; predpokladá sa že každý užívateľ patrí práve do 1 odboru

Tabuľka dm_thesis

V tejto tabuľke udržiavame informácie o všetkých doposiaľ odovzdaných prácach

ID opäť primárny a zároveň unikátny kľúč, ktorý jednoznačne identifikuje dizertačnú prácu(z dôvodu podobného popisu využitia, pri niektorých ďalších tabuľkách atribút ID nebude uvádzaný)

NAME meno dizertačnej práce

DOKTORAND_ID ID užívateľa, ktorému prislúcha daná práca

SECTION_ID ID odboru pod ktorý spadá práca; je totožné s doktorandovým ID odboru; zduplikované pre časté používanie v rámci selektovania

LECTOR_ID ID užívateľa, ktorý je školiteľom danej práce

ADDED dátum pridania práce do systému

FILENAME reťazec reprezentujúci súbor uploadovaný do systému; aby sa nestávalo, že by sa názvy súborov zhodovali a tak sa vzájomne prepísali, je pred meno súboru zahashovaná hodnota ID prislúchajúceho doktoranda

OPPONENTS reťazec zložený z ID užívateľov, ktorí sú oponentmi zadanej práce

COMMITTEE_ID ID komisie, ktorá je priradená k danej práci

Tabuľka dm_events

Tabuľka s informáciami o úlohách vykonávajúcich sa na jednotlivých dizertačných prácach

TYPE_EVENT typ úlohy o ktorú ide; môže ísť o 1 z 3 úloh: schvaľovanie oponentov, odovzdanie posudkov, schvaľovanie termínu obhajoby

THESIS_ID ID práce ohľadom ktorej sa vykonáva príslušná úloha

MAIL_TO, MAIL_CC emailové adresy užívateľov, ktorým je adresovaný mail o úlohe a adresa predsedu komisie, ktorému je určená kópia mailu

MAIL_SUBJECT, MAIL_BODY pole s predmetom a obsahom mailu odosielaného predsedom komisie príslušným zložkám v procese obhajoby

STATUS, STATUS_CHANGED atribúty, ktoré označujú v akom stave sa nachádza plnenie danej úlohy(hodnoty 0-3 označujú: úlohu pripravenú na odoslanie, čakajúcu na odpovede adresátov, prekračujúcu stanovený termín a splnenú úlohu) a to, kedy bol tento status naposledy menený

DEADLINE v tomto atribúte si ukladáme dátum zvolený predsedom komisie, do ktorého je potrebné úlohu skompletizovať

CONFIRMED tento atribút reprezentuje reťazec zložený z ID užívateľov, ktorí si už splnili svoju úlohu; na základe hodnoty tohto atribútu vieme zistiť, či je úloha už skompletizovaná resp. na koho sa v nej ešte čaká

TERMINFO pole, do ktorého sa v prípade úlohy o schvalovaní termínu obhajob zaznamená dátaum navrhovaného termínu konania obhajob dizertácie

Tabuľka dm_email_templates

Tabuľka, v ktorej máme prednastavené formy emailov rozosielaných pri rozličných udalostiach uskutočnených v systéme, tieto predlohy sa prostredníctvom administračného rozhrania dajú podľa potrieb modifikovať

SUBJECT, BODY predmet a obsah mailovej predlohy, v rámci ktorej sa vyskytujú podreťazce ohraničené špeciálnymi znakmi '%', ktoré sú pri konkrétnej realizácii emailu nahrádzané skutočnými hodnotami, vzhľadom na typ úlohy ktorá sa v tom momente vykonáva

REQUIRED_ATTR v tomto atribúte si pri jednotlivých emailových predlohách uchovávam zoznam kľúčových premenných, ktoré vzhľadom na neskoršiu kompatibilitu musia byť obsiahnuté v každej pozmenenej predlohe; tento zoznam je používaný na kontrolu korektnosti zmeny

Tabuľka dm_news

V tejto tabuľke udržiavame informácie o novinkách, ktoré administrátor podľa situácie dáva zobrazovať na hlavnej stránke

TITLE, CONTENT titulok a obsah novinky

AUTHOR atribút obsahujúci meno administrátora, ktorý novinku pridal do systému

DATETIME dátum a čas pridanie, resp. poslednej modifikácie príslušnej novinky

ACTUAL informácia o tom, či je daná novinka momentálne aktuálna a tým pádom, či sa má zobrazovať na hlavnej stránke

Tabuľka dm_committee

Tabuľka s informáciami o komisiách, ktoré boli doteraz vytvorené v systéme

CHAIRMAN ID užívateľa, ktorý tejto komisii predsedá

MEMBERS reťazec reprezentujúci členov komisie, pozostávajúci z identifikačných čísiel jednotlivých členov medzi sebou oddelených zvislými čiarami

SECTION_ID identifikátor odboru, v rámci ktorého funguje príslušná komisia

Tabuľka dm_pagerights.

Táto tabuľka je preddefinovaná a drží v sebe informácie o prístupových právach k jednotlivým stránkam, zobrazovaní položiek v menu aplikácie a stromovej štruktúre stránok využitú v navigácii systému

ID jednoznačný identifikátor stránky, ktorý udáva číslo vrcholu v stromovej štruktúre stránok, na základe ktorého sa vieme odkazovať z atribútu TREE na tzv. "predkov" a týmto spôsobom vytvárať navigáciu

PAGENAME reťazec reprezentujúci názov stránky zobrazovaný v systéme ako titulok, ale aj ako nápis v menu a v navigácii

PAGELINK atribút obsahujúci skrátenú časť URL adresy, ktorá sa dopĺňa do hyperlinkov položiek menu a navigácie, a v rámci prístupových práv je kontrolovaná na možnosť prístupu jednotlivých užívateľských skupín

MENU_ITEM táto 1 bitová hodnota nám vypovedá o tom, či je daná stránka súčasťou menu v systéme alebo nie

GROUPS veľmi dôležité pole vzhľadom na bezpečnosť aplikácie, ktoré určuje prostredníctvom množiny hodnôt, ktoré užívateľské skupiny majú prístup k tej-ktorej stránke

POSITION v tomto atribúte v prípade, že je stránka súčasťou menu, špecifikujeme akúsi hodnotu pozície tejto menu položky; na základe týchto hodnôt vieme pri výpise

zoradiť všetky položky podľa špecifikovaného poradia; v prípade, že stránka nie je menu položka, zostáva atribút prázdny

TREE tento atribút v sebe obsahuje identifikátor stránky, ktorá je v stromovej štruktúre v pozícii akéhosi "otca", čo vieme dobre využiť na vytváranie navigácii na stránke; koreň tohto stromu má v tomto prípade zadanú 0, čo v našom prípade reprezentuje Hlavná stránka

Tabuľka dm_groups

Tabuľka užívateľských skupín, ktoré systém podporuje; táto tabuľka je statická a počas behu systému sa v nej nevykonávajú úpravy, v prípade zmien treba zasiahnuť do štruktúr v tabuľke.

ID identifikátor skupiny zvolenej pre užívateľa, ktorý sa využíva pri zisťovaní prístupových práv k jednotlivým stránkam aplikácie

GROUP_NAME pomenovanie jednotlivých užívateľských skupín, ktoré sa zobrazuje pri pridávaní nového užívateľa systému

Tabuľka dm_sections

Tabuľka obsahujúca zoznam študijných odborov, s ktorými aplikácia spolupracuje; táto tabuľka je takisto statická a modifikácia podporovaných odborov v nej je treba tiež vykonávať manuálne.

ID číslo jednoznačne identifikujúce jednotlivé odbory, s ktorými systém pracuje

SECTION_NAME príslušné názvy študijných odborov

Kapitola 4

Realizácia

4.1 Použité technológie

V tejto sekcii popíšem technológie, pomocou ktorých bola aplikácia vyvíjaná a ktoré pri svojej činnosti využíva.

4.1.1 XHTML, CSS a UTF-8

Na tvorbu web stránok bola použitá syntax XHTML (extensible hypertext markup language), ktorý má prísnejšiu syntax ako štandardné HTML, avšak pre prípad, žeby sa aplikácia rozširovala, je dobré mať použitý radšej striktnejší variant kódovania, v ktorom sa podľa vlastnej skúsenosti programátor ľahšie zorientuje.

Ďalšou použitou technológiou na vytváranie webovej prezentácie sú kaskádové štýly - CSS (cascading style sheets), ktoré slúžia k oddeleniu obsahovej časti web stránky od vzhľadovej časti. Ak aplikácia má mať nejaký štandardný vzhľad, ktorý sa aplikuje na všetky zobrazované stránky, je výhodné pre programátora zadefinovať si tento vzhľad v rámci CSS a tým pádom zmena vzhľadu prezentácie je sprevádzaná iba malými zmenami v tomto kóde.

Pre kódovanie celej aplikácie bolo zvolené kódovanie UTF-8, nakoľko systém počnúc prezentačnou vrstvou až po databázové systémy pracuje s diakritikou a tým pádom nevznikal problém v komunikácii týchto dvoch zložiek.

4.1.2 Databázový systém

Komplexné internetové aplikácie obvykle využívajú k svojmu behu databázový systém v ktorom sú uložené dáta s ktorými aplikácia interaguje. Je vhodné aby komunikácia s takýmto databázovým systémom bola čo najrýchlejšia a zároveň aby bola nezávislá od typu webového servera a platformy, kde systém budeme prevádzkovať. Aj preto padlo rozhodnutie pre použitie voľne šíriteľnej a veľmi populárnej databázy MySQL.

4.1.3 Skriptovací jazyk

PHP5 je objektovo orientovaný skriptovací jazyk určený predovšetkým na vytváranie dynamických webových stránok, využívajúci silu objektového programovania. Tento jayzk spracováva dáta, ktoré odovzdáva templatovému systému, prostredníctvom ktorého je generovaná výsledná webová stránka. Ďalším plus navyše je, že spolupracuje s najrozšírenejšími webovými servermi a zároveň je nezávislý na platforme, čím sa aplikácia stáva flexibilnou vzhľadom na možnosť budúcej zmeny operačných systémov respektíve webových serverov na fakulte.

4.1.4 Templatový systém

Templatový systém je jednou zo základných technológii, ktoré by mala využívať každá komplexnejšia webová aplikácia. Pre programátora znamená tento systém možnosť oddeliť prezentačnú vrstvu (XHTML kód) webovej aplikácie od programového kódu(PHP kód) použitého na danej stránke. Teda úplné oddelenie designu od funkcionality, ktoré sa zúročí predovšetkým v tom, že aplikácia sa stáva prehľadnejšou, čo v prípade ďalšieho rozširovania môže ušetriť enormné množstvo času. V našom prípade bol zvolený templatový systém SMARTY, ktorý v sebe zahŕňa všetky vyššie spomenuté výhody a navyše poskytuje nástroje uľahčujúce programátorovi prehľad nad komunikáciou týchto 2 zložiek aplikácie.

4.2 Základný popis vzhľadu prostredia

Systém na monitorovanie prípravy obhajoby dizertačných prác je navrhnutý tak, aby spracovával a prezentoval dáta sprostredkované administrátorom a jednotlivými zložkami užívateľov.

Pojmom užívateľ označujeme buď predsedu obhajovacej komisie, jej členov, doktoranda, školiteľa, alebo oponenta dizertačnej práce, s tým, že na aktívnom prístupe do systému sú zúčastnení užívatelia zo skupín predsedov komisie školiteľov a doktorandov. Vytvorenie prístupu aj pre užívateľov zo zvyšných skupín by nespôsobilo žiadne konflikty, ale systém je realizovaný tak, že ráta s ostatnými zložkami len do pozície akýchsi externých užívateľov.

Systém by ale nemohol správne fungovať bez role správcu systému, ktorý vykonáva základné akcie pre správne fungovanie celého systému. Znamená to, že v administračnej časti sa musia nachádzať nástroje pre spravovanie užívateľov, dizertačných prác, obhajovacích komisií, mailových predlôh, ale aj noviniek v systéme. Administrátor navyše musí mať možnosť meniť alebo mazať nesprávne zadané, alebo už neaktuálne údaje.

Rozhranie systému má byť pre správcu, ale aj pre ostatných užívateľov prichádzajúcich do kontaktu so systémom prehľadné a navigácia v ňom by mala byť intuitívna. Zároveň sa prihliada na to, aby bola aplikácia jednoducho a komfortne použiteľná aj pre bežného užívateľa bez odbornejšieho počítačového vzdelania.

V nasledujúcej časti uvádzam popis 2 základných štýlov použitých na vyobrazenie webových stránok.

4.2.1 Vzhľad a funkcionalita prihlasovacej stránky

Aplikácia na monitorovanie prípravu obhajoby dizertácie je systém určený výlučne pre osoby, ktoré sú priamo zúčastnené na tomto procese, preto musí byť prístup k nemu vybavený prihlasovacím formulárom, ktorý si od žiadateľa o vstup do tohto systému vyžiada prihlasovacie meno a heslo. Vstup bude umožnený iba užívateľom, ktorí správne vyplnia tieto údaje a následne odošlú požiadavku stlačením tlačítka pre prihlásenie, po ktorom systém overí správnosť údajov porovnávaním s dátami v databáze. Ak boli vyplnené údaje korektné, presmeruje užívateľa na hlavnú stránku a zobrazí položky prislúchajúce skupine prihláseného užívateľa, v opačnom prípade opäť zobrazí prihlasovací formulár s hlásením nepresne zadaných údajov.

Táto stránka je zároveň aj cieľovou stránkou po odhlásení užívateľa aplikácie a v taktiež prípade, že návštevník si vyžaduje skript, ktorý vyžaduje prihláseného užívateľa, alebo nepodlieha právam skupiny prihláseného užívateľa.

4.2.2 Plavná stránka

Po úspešnom prihlásení je užívateľ presmerovaný na úvodnú stránku obsahujúcu prvky ako navigáciu v systéme, menu prislúchajúce užívateľským právam, aktuálne novinky a možnosti zmeny hesla a odhlásenia zo systému.

Nakoľko v prípade systému ide o aplikáciu využívanú na účely fakulty, je aj dizajn prispôsobený na spôsob prevedenia fakultných stránok. Už z loga a nápisu na hornom panely je zrejmé, že sa jedná o stránky využívané na akademickú činnosť. Tieto majú na užívateľa pôsobiť jednoduchým, ale zato prehľadným dojmom. Globálna navigácia v podobe menu aplikácie sa vyobrazuje na bočnom panely. Je vypísaná primerane veľkým písmom, tak aby nepôsobila rušivo a zároveň aby bola aj prehľadná a ľahko čitateľná. Pre lepšiu orientáciu je v systéme implementovaná aj navigácia, ktorá užívateľovi uľahčuje orientáciu v ňom.

V tomto štýle sú zobrazované aj všetky ostatné stránky programového systému.

4.3 Testovanie systému

Na testovanie programátorskej logiky použitej v systéme bol použitý voľne šíriteľný balíček Wamp, ktorý slúži na inštaláciu Apache serveru, interpreteru jazyka PHP a MySQL databázy pod systémom Windows a tým umožňuje lokálnu simuláciu fungovania reálneho webového servera. Vďaka tomu som mohol prostredníctvom zadávania mnou zvolených testovacích vstupov prekontrolovať funkcionalitu všetkých častí systému. Tá pri volení náhodných údajov nevykazovala náznaky nepredvídaného správania, čím sa potvrdila logická štruktúra implementovaných metód.

Čo sa týka testovania obsahu a spôsobu vykresľovania stránok boli použité najpoužívanejšie webové prehliadače od rôznych vývojárov: Internet Explorer 7, Opera 9.64 a Mozilla Firefox verzia 3.0. Vykresľovanie všetkých podstatných záležitostí bolo týmito prehliadačmi interpretované rovnako a ich rozdiely ktoré medzi nimi vznikali nespôsobovali zhoršenie v zobrazovaní obsahu ani v orientácii v systéme.

Kapitola 5

Bezpečnosť systému

Nakoľko u vyvíjanej aplikácie sa predpokladá jej reálne využitie v praxi, bezpečnosť systému je jednou z tém, ktorej treba venovať patričnú dávku pozornosti. Toto tvrdenie len potvrdzuje fakt, že ide o aplikáciu, ktorá bude používaná výlučne na univerzitnej pôde a tým pádom bude obsahovať údaje, ku ktorým by nemali mať prístup nepovolané osoby. Aj preto sa v nadchádzajúcej kapitole budem venovať bezpečnostným otázkam týkajúcich sa tejto aplikácie.

5.1 Základy bezpečnosti

Pri každej aplikácii, ktorá spracováva užívateľské dáta a o to viac v prípade aplikácie pracujúcej s takými dôležitými dátami, akými sú dáta súvisiace s procesom prípravy obhajoby dizertácie je nutné, aby dáta boli prístupné iba tým užívateľom, ktorý majú k nim oprávnenia. Toto vieme zabezpečiť pomocou autentifikácie jednotlivých užívateľov a kontrolou ich prístupových práv.

5.1.1 Autentifikácia užívateľov

Ako vo väčšine prípadov aplikácii aj v tomto prípade autentifikácia užívateľov pozostáva zo zadania mena a hesla. Tieto sú užívateľovi pridelené administrátorom systému po tom, čo je odovzdaná doktorandská práca. Tento systém prideľovania zabezpečí to, že v systéme

budú iba tí užívatelia, ktorí budú zúčastnení na procese obhajoby a tým sa do veľkej miery obmedzí nebezpečenstvo útoku zvnútra systému.

Ak vyplnené údaje boli korektne zadané a samotné prihlásenie sa do systému prebehne úspešne, užívateľ sa dostane na hlavnú stránku kde sa mu zobrazia položky poľa toho, do akých užívateľských skupín je zaradený a teda aké prístupové práva k jednotlivým akciám vlastní.

5.1.2 Prístupové práva užívateľov

Na to, aby systém fungoval tak, ako bolo navrhnuté, je potrebné rozdeliť užívateľov do určitých skupín, podľa toho, aké má mať kto v systéme právomoci. V našom systéme rozlišujeme 6 základných skupín užívateľov: administrátor, predseda komisie, člen komisie, doktorand, školiteľ a oponent. Z toho iba u 3 skupín, ktorými sú skupiny administrátorov, predsedov komisie a doktorandov, sa ráta aj s pridelením prihlasovacích údajov.

Aby sme mohli medzi týmito užívateľmi rozlišovať je potrebné po prihlásení sa do systému získať údaje o užívateľovi, medzi inými aj tie, ktoré hovoria o tom, v akých skupinách je zaradený. Tieto pre odbremenenie opakovaného overovania sú uložené do premennej session, ktorú máme nastavenú počas celého sedenia používateľa(možné útoky na session rozoberám v sekcii session stealing). Následne pri každom pokuse o prístup k niektorej stránke sa ešte počas volania konštruktora triedy, v rámci ktorej je definovaná stránka ktorú požadujeme, konfrontujú tieto údaje o prístupových právach s funkciou *CheckRights(page),* kde *page* určuje jednoznačnú identifikáciu skupiny stránok, ku ktorej sa chce pristupovať. V prípade zistenia, že skupina ktorej práva vlastní užívateľ je obsiahnutá pri danej stránke, prístup k nej je odobrený a pokračuje v sa v jej inicializácii. V opačnom prípade je vyobrazené upozornenie o nepovolanom prístupe k stránke a zvolí sa prihlasovací formulár pre možnosť prihlásenia užívateľa s príslušnými právami.

5.2 Útoky na aplikačnú vrstvu

Každá aplikácia bez ohľadu na to, aké má funkčné predurčenie, by mala spĺňať aspoň základné bezpečnostné kritéria. Pri zabezpečení by sa totiž ak už na nič iné, malo prihliadať na to, že globálna informačná bezpečnosť sa týka každého jedného subjektu v rámci siete

internet a jeho nedostatočné zabezpečenie len môže dopomáhať k rozširovaniu internetovej kriminality. V nasledujúcich sekciách opíšem, ktorým hrozbám som sa snažil predísť a aký spôsob bol na to zvolený.

5.2.1 Injection

Prvý typ útoku (injection) využíva nedostatočné ošetrenie vstupu od užívateľa pri vytváraní dotazov adresovaných do iných systémov s ktorými pri prevádzke aplikácie komunikujeme. Najčastejšími variáciami tohto typu útoku je SQL injection a HTML injection.

SQL Injection

V prípade SQL injection sa jedná o nedostatočné ošetrenie parametrov pri vytváraní SQL dotazov v našom prípade komunikujúcich s MySQL databázou.

Ukážka možnosti zneužitia nezabezpečeného dotazu (premenná param obsahuje text získaný zo vstupu):

*query = "SELECT * FROM tabulka WHERE tabulka.pole = '".param."';";*

Pokiaľ by užívateľ prostredníctvom vstupu priradil do premennej *param* napríklad hodnotu *"' OR 1=1 OR tabulka.pole = '"*, môže týmto spôsobom prečítať všetky dáta obsiahnuté v celej tabuľke, nakoľko výsledným dotazom bude:

*query = "SELECT * FROM tabulka WHERE tabulka.pole = '"' OR 1=1 OR tabulka.pole = '"';";*

Rovnako tak môžu byť dáta aj nedovolene vkladané, prepisované, alebo aj mazané.

Tento problém má ale riešenie a to, nevytvárať SQL dotaz čisto len jednoduchým skladaním reťazcov, ale prevádzať vstupné dáta cez ošetrujúce funkcie obsiahnuté aj v knižniciach jazyka PHP.

Či už ide o spôsob získavania dát prostredníctvom URL adresy (superglobálna premenná GET), alebo prostredníctvom odoslaného formulára (superglobálna premenná POST), bolo potrebné zvalidovať tieto formy vstupov. Validovanie týchto dát pozostáva z dvoch druhov procesov.

Prvým je takzvaný “cleanup” proces, ktorého úlohou je vyhnúť sa akýmkoľvek špeciálnym znakom a tak sa pri vytváraní ľubovoľnej stránky ešte pred jej zobrazením je volaná trieda *Application*, v rámci ktorej sa vykonáva inicializovanie volanej stránky, ktorého súčasťou je aj ošetrenie všetkých prenášaných parametrov funkciou *mysql_real_escape_string().* Táto funkcia zabezpečí, že každý vytvorený vstup do systému bude ešte predtým ako sa dostane do kontaktu s databázou ošetrený voči tzv. špeciálnym znakom a tak výsledný vstup nebude môcť mať nežiadúce účinky vzhľadom na funkčnosť aplikácie.

Druhým dôležitým procesom v rámci vytvárania dotazov je kontrola typu premennej obsiahnutej v rámci dotazu, ktorá sa vykonáva pri všetkých dôležitých dotazoch založených na vstupných dátach od užívateľa. Nakoľko drvivá väčšina dotazov v ktorých sa odkazuje na užívateľov vstup filtruje na základe identifikačného čísla niektorej z položiek, spomeniem za všetky funkciu *is_numeric()*, kontrolujúcu či premenná, ktorú máme v pláne využiť v rámci dotazu je z oboru celých čísel, čím dokážeme predísť zbytočným komplikáciám.

HTML Injection

v rámci aplikácie potrebujeme na mnohých miestach vypisovať evidované dáta a tak zobrazujeme aj vstupy, ktoré boli v aplikácii zadané užívateľom. V prípade, že by sa vypisovalo presne to, čo bolo zadané užívateľom, hrozila by možnosť, že užívateľ by do formulára pre vstup dát zadal okrem reálnych dát aj časti HTML kódu, ktorý by sa mohol pri najbližšom výpise vykonať a spôsobiť tak prezentačné problémy v aplikácii.

Riešením aplikovaným do systému je podobne ako pri SQL injection funkcia *htmlspecialchars()* volaná v rámci načítavaní stránky v triede *Application.* Táto funkcia prekonvertuje všetky vyhradené HTML znaky na neškodné HTML entity a tak pri výpise týchto dát sa zobrazí rovnaký kód aký bol zadávaný na vstupe, namiesto toho, aby sa vykonal. Týmto spôsobom vykonáme akúsi prevenciu voči potenciálnym záškodníkom.

5.2.2 Cross-site scripting

Taktiež označovaný ako útok XSS. Ide o pokročilejšiu techniku útoku s využitím javascriptu, kde cieľom útočníka je spustiť skript, ktorý môže mať rôznu úroveň nebezpečia. Najčastejšie používané sú skripty, ktoré spôsobia buď presmerovanie na inú webovú stránku, vypísanie

určitého chybového hlásenia, alebo podstrčiť užívateľovi nenápadný skript s cieľom prinútiť ho prezradiť svoje identifikačné údaje. Toto útočník docieli tak, že skript zakomponuje na stránku, ktorú si obeť zobrazí a tým ho spusti. Spôsobov vloženia skriptu je viacero. Pokiaľ by systém nekontroloval vstupné údaje proti výskytom skriptov, mohol by sa skript zapísať do databázy, alebo súboru. Inou možnosťou je keď útočník zakomponuje svoj skript do adresy stránky.

Tento typ útoku je využívaný hlavne pri nedostatočnom ošetrení vstupných hodnôt vo formulároch a v prípade, že sa spracováva časť URL adresy ako vstup bez predošlého ošetrenia výskytu špeciálnych znakov. Preto v prípade nášho systému sú všetky vstupy vykonané touto formou rovnako ako v predošlom prípade útokov ošetrované ešte počas inicializácie zadanej stránky. Je na to použitá funkcia *htmlspecialchars(),* ktorá prekonvertuje všetky špeciálne znaky na príslušné HTML entity, ktoré sa pri následnom vyvolaní namiesto vykonania iba čisto vypíšu.

5.2.3 Session stealing

Rôzne informácie o tom, či je užívateľ prihlásený a aké oprávnenia pritom má, sa uchovávajú v superglobálnej premennej SESSION. Avšak použitie tejto premennej prináša jeden z ďalších typov útoku na webový systém, a to tzv. session stealing. Ten spočíva v tom, že stačí nejakým spôsobom získať cudzí session identifikátor a tým pádom sa vydávať za osobu, ktorej po jej prihlásení bol session identifikátor oficiálne pridelený.

Aby sme predišli tejto hrozbe v našom systéme, tak po každom prihlásení užívateľa je okrem pridelenia identifikátoru sedenia(session) zaznamenaná aj IP adresa, z ktorej k tejto aplikácii pristupuje. Tým pádom pri načítavaní stránky v ktorej by mal útočník ambíciu zneužiť session vytvorené inou osobou a predstierať korektne prihláseného užívateľa, sa nekontroluje čisto iba session identifikátor, ale zároveň aj IP adresa klienta, čo výrazne zvýši odolnosť pred týmto druhom útoku.

Kapitola 6

Záver

Ciele vytýčené v úvode práce boli zrealizované a tak vznikla stabilná a bezpečná aplikácia, ktorá má všetky predpoklady byť reálne využívaná v praxi a tým pádom zefektívniť samotný proces prípravy obhajoby dizertácii.

Programový systém bol vytvorený na platforme Windows a jeho vyhotovenie spĺňa všetky vopred určené požiadavky. V súčasnosti je potrebná jeho inštalácia a sprevádzkovanie na webovom priestore fakulty, čo bude predmetom práce najbližších dní zo strany predkladateľa tejto práce v spolupráci so správcom systému fakulty.

Samotná aplikácia je štandardne nastavená tak, že pri voľbe študijných odborov podporuje Informatiku a Matematiku. Výber týchto študijných smerov je však možné aj zmeniť alebo rozšíriť manuálnym zásahom do štruktúry databázového systému.

Možnosti na rozšírenie tejto práce sú potenciálne široké, nakoľko webové stránky možno prispôsobovať stále novším a vyvíjajúcim sa technológiám ako i aktuálnym potrebám fakulty, týkajúcich sa informovanosti o aktuálnom dianí v procesoch obhajoby, či už zo strany študentov, pedagógov a pracovníkov predmetnej fakulty.

Literatúra

[Bie08] Martin Bies. *Aplikácia na monitorovanie prípravy obhajoby dizertácie*. FMFI UK, Bratislava, 2008.

[GBR05] Andi Gutmans, Stig Seather Bakken a Derick Rethans. *Mistrovství v PHP 5*. CP Books, 2005.

[Kab03] Mohammed J. Kabir. *Secure PHP Development: Building 50 Practical Applications*. Wiley Publishing, Inc., 2003.

[WL02] Hugh E. Williams and David Lane. *Programujeme webové aplikace pomoci PHP a MySQL*. Computer Press, 2002.

[NGS06] Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz a Michael K. Glass. *Vytváříme webové aplikace v PHP5, MySQL a Apache.* Computer Press, 2006.

[PHP] PHP: Hypertext preprocessor. `http://www.php.net/`.

[SMTY] SMARTY: Template Engine. `www.smarty.net/`.

Prílohy

K tomuto dokumentu je priložené CD so samotnou aplikáciou, ktorá má prednastavené hodnoty. Táto aplikácia bola zároveň implementovaná na fakultný server a je ju možné nájsť na doméne `http://cvika.dcs.fmph.uniba.sk/~docmonitor/`.