

**Fakulta matematiky, fyziky a informatiky
Univerzity Komenského
Bratislava**



**Probatur – online testovanie
študentov**

Bakalárska práca

Autor: Juraj Porubský
Vedúci: RNDr. Richard Ostertág

Bratislava
máj 2007

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

V Bratislave 31. mája 2007 _____

Ďakujem svojmu vedúcemu bakalárskej práce, RNDr. Richardovi Ostertágovi za cenné rady a pripomienky pri písaní tejto práce.

Úvod

Kapitola 1

Návrh softvéru

Pred samotným programovaním sme najprv dopodrobna rozobrali funkcionálnosť, ktorú očakávame od nášho softvéru. Ukázalo sa, že náš softvér sa delí na tri logicky ucelené časti:

- Administrátorská:
Spravuje zoznam učiteľov, študentov, predmetov, priradenia študentov k predmetom, importovanie informácií zo systému Felix.
- Učiteľská:
Dovoľuje učiteľovi spravovať si svoje predmety, ako napríklad vytvárať testové otázky, generovať testy na základe rôznych kritérií, vyhľadávať študentov a testy k nim.
- Študentská:
Cieľom je vytvoriť študentské rozhranie, ktoré poskytne študentovi informácie o zapísaných predmetoch, možnosť napísať test, zobrazí známky a bodovanie už napísaných testov.

Podľa nich sme si prácu aj rozdelili. Martin Roško spracoval administrátorskú, František Šmitala študentskú a ja učiteľskú časť.

1.1 Voľba softvéru a IDE rozhrania

Dôležitou požiadavkou nášho softvéru je, aby bol prístupný aj z internetu (až na samotné testovanie, ktoré môže prebiehať len na lokálnej sieti). Pre tento dôvod sme sa rozhodli, že systém naprogramujeme ako webovskú aplikáciu. Toto rozhodnutie prináša ďalšie výhody a to, že si užívateľ nepotrebuje sťahovať žiaden softvér na to, aby mohol využívať náš program, a že systém je multiplatformový.

Výber skriptovacieho jazyka a databázy:

Skriptovací jazyk sme si zvolili PHP. Zvažovali sme aj ASP.NET, ale nakoľko sme už všetci mali skúsenosti s PHP, rozhodnutie bolo jednoznačné a v prospech PHP. Výber databázového servera bol podobný. Rozhodli sme sa tiež hlavne na základe našich predchádzajúcich skúseností a to pre MySQL. Okrem MySQL sme zvažovali ešte PostgreSQL. Dobrý výber nám potvrdila ešte aj skutočnosť, že MySQL zvláda vyhodnocovať väčšinu dotazov v tretinovom čase oproti PostgreSQL.

Šablónový systém:

Používanie šablónových systémov má vo webovských aplikáciach veľkú výhodu. Umožňuje oddeliť výzor stránky, teda jej štruktúru, od programovej-logickej časti. Týmto získame väčšiu flexibilitu pri opätovnej zmene dizajnu stránky. Použili sme šablónový systém Smarty (1.2 Smarty).

Výber IDE:

IDE je skratka od *Integrated Development Environment* a označuje vývojové prostredie. Vývojové prostredie pre PHP som si zvolil PSPad. Dôvodov bolo niekoľko. Grafický výzor má jednoduchý, ale nie na úkor funkcionality. Obsahuje zvýrazňovač syntaxe, dopĺňanie kódu, odsádzanie viacerých riadkov naraz, čo mi napríklad v editore UltraEdit chýbalo. Spravovanie MySQL databázy sme robili v phpMyAdmin, čo je webovské rozhranie na správu MySQL databázy.

1.2 Smarty

Smarty je šablónový systém napísaný v PHP. Oddeľuje PHP, ako logického celku, od HTML, prezentačného a generuje webovský obsah nahradzovaním špeciálnych Smarty znakov vo vnútri dokumentu. Smarty znaky sú direktívami a sú uzatvorené do `{ }`. Direktívami môžu byť premenné (začínajúce `$`), funkcie, logické tvrdenia alebo riadiace štruktúry. Smarty šablóny sa ukladajú v súboroch s koncovkou `'.tpl'`. Po zavolaní metódy *display*, ktorej úlohou je previesť šablónu do HTML kódu, sa na základe šablony vygeneruje pre programátora transparentný PHP súbor, ktorý sa následne spustí. Jeho výsledkom je konečný HTML kód, ktorý sa zobrazí užívateľovi. Smarty šablóny, týkajúce sa učiteľského rozhrania, sú uložené v `'/teacher/smarty/templates'`.

Miesto, do ktorého sa majú generovať PHP súbory, vytvorené pomocou Smarty, je '/teacher/smarty/templates_c'.

1.3 Použité štandardy

Dôležitou súčasťou vyvíjania softvéru je dodržiavanie určitých štandardov. Štandardy týkajúce sa webovských aplikácií zasahujú hlavne do dvoch oblastí :

- HTML kód – definuje štruktúru stránky. Existuje viacero štandardov: HTML, XHTML. XHTML síce vyžaduje striktnéjšie definovanie jednotlivých elementov, ale prináša jednoznačnosť interpretovania štruktúry webovským prehliadačom. Preto bola voľba štandardu jednoznačná a v prospech XHTML.
- CSS štýly – používajú sa na oddelenie obsahu od výzoru, tým že definujú vzhľad jednotlivých HTML elementov. Všetky vlastnosti HTML elementov týkajúce sa výzoru sme definovali pomocou CSS štýlov.

Pri vývoji učiteľského rozhrania bolo potrebné definovať aj štandard na výmenu štrukturovaných dát medzi klientom a serverom (2.3.2.2.4 Sekcia importovanie). Pre tento účel sme definovali štandard využívajúci XML formát, ktorého vnútornú štruktúru sme definovali DTD dokumentom.

Kapitola 2

Štruktúra programu

V tejto kapitole je rozobraná štruktúra jednotlivých rozhraní systému. Administrátorské a študentské rozhrania, naopak od učiteľského, sú opísané stručnejšie, nakoľko nie sú predmetom tejto práce. Podávajú však ucelený pohľad na funkčnosť systému, preto vznikla potreba ich do tejto kapitoly zaradiť.

2.1 Rozhranie pre administrátora

Vzhľadom na predpokladané veľké množstvo užívateľov, je rozhranie pre správcu systému prehľadné a navigácia je intuitívna. Jeho súčasťou sú aj nástroje pre zmenu nastavení systému, napríklad možnosť editovať prístupové údaje k vzdialenému serveru, poskytujúcemu dáta na import alebo zmena hesiel jednotlivých užívateľov, vrátane hesla samotného administrátora.

2.1.1 Prihlasovacia stránka

Administrátorské rozhranie je určené výlučne pre správcu systému, preto je prístup k nemu vybavený prihlasovacím formulárom, ktorý žiada od žiadateľa pre vstup do tejto časti systému prihlasovacie meno a heslo. Vstup je umožnený iba užívateľom, ktorí správne vyplnia tieto údaje a následne odošlú požiadavku stlačením tlačítka pre prihlásenie, po ktorom systém overí správnosť údajov. Ak boli vyplnené správne, presmeruje administrátora na hlavnú stránku administrátorského rozhrania, v opačnom prípade zobrazí opäť prázdny prihlasovací formulár.

2.1.2 Študenti

Študenti a učitelia sa prihlasujú do systému pre online testovanie cez študentské, respektíve učiteľské rozhranie. Pre správne fungovanie sú však vytvorené a každému užívateľovi pridelené kontá, tak aby bolo možné, po prihlásení sa študenta alebo učiteľa, ho jednoznačne identifikovať v systéme. Každý užívateľ má pridelené jednoznačné prihlasovacie meno a heslo. Študent má pridelené predmety, ktoré má v danom roku zapísané z ktorých môže byť skúšaný. Úlohou administrátora je zabezpečiť, aby sa každý užívateľ dokázal do systému prihlásiť, a aby mal pridelené všetky predmety, ktoré pri svojej práci potrebuje. Aby bola práca administrátora efektívna, musí byť sekcia so študentami a učiteľmi prehľadná, jednoduchá na ovládanie a intuitívna.

V sekciách študentov a učiteľov, ktoré sú od seba oddelené, je hlavným navigačným prvkom zoznam užívateľov s akciami, ktoré sa na nich dajú vykonať. Pod akciami sa rozumie pridávanie, editovanie, zmena hesla a mazanie užívateľov. Súčasťou týchto sekcií je vyhľadávací formulár a formulár pre pridávanie užívateľov do systému.

2.1.3 Odbory

Štúdijné odbory sú v systéme vytvorené pre zvýšenie prehľadnosti a zefektívnenie práce so študentskými kontami. Každý študent môže patriť pod maximálne jeden štúdijný odbor. Odbory majú pridelených svojich garantov, ktorými môžu byť iba učitelia, nachádzajúci sa v systéme.

2.1.4 Predmety

V študentskom rozhraní systému sa študentom po prihlásení zobrazí zoznam predmetov, z ktorých môžu byť testovaní v priebehu roka. Učiteľské rozhranie poskytuje učiteľom možnosti vytvárania sád otázok a generovanie testov pre jednotlivé predmety. Preto spolu so študentskými a učiteľskými kontami v systéme, je veľmi dôležitá aj správa predmetov. Znamená to, že každý predmet, ktorý má byť využívaný v systéme pre online testovanie, musí mať prideleného učiteľa a zoznam študentov, ktorí z neho môžu byť skúšaný.

2.1.5 Import

Pridávanie všetkých dát do systému manuálne by zabralo obrovské množstvo času a zvýšila by sa pravdepodobnosť chýb pri zadávaní údajov. Preto je systém pre online testovanie študentov navrhnutý tak, aby dokázal komunikovať a využívať služby už existujúceho systému pre prihlasovanie sa študentov na skúšky alebo ľubovoľného iného systému, ktorý dokáže komunikovať so systémom pre testovanie študentov pomocou jednoduchého protokolu.

2.1.6 Záloha

Pri návrhu systému, je dôležité myslieť dopredu, teda, ak sa bude využívať dlhšiu dobu, postupne sa naplní údajmi, ktoré sa stanú medzičasom nepotrebné, alebo budú dôležité iba kvôli štatistikám. Sem patria údaje o študentoch a skúškach a údaje, ktoré sú na ne priamo naviazané, napríklad testy a odpovede na ne. V tejto sekcii bude možné vykonať akcie, ktoré zabezpečia zálohu dát, načítanie zálohy do databázy a jej odstránenie.

2.1.7 Nastavenia

Kvôli bezpečnosti je v systéme možnosť zmeny prístupového hesla do administrátorského rozhrania. Používanie rovnakého hesla počas dlhšej doby by dalo potenciálnemu útočníkovi väčšiu šancu na jeho získanie a zároveň, ak by nebola v systéme možnosť zmeny hesla, už raz útočníkom získané heslo by nebolo možné zmeniť. V sekcii s nastaveniami sa preto okrem nastavení údajov potrebných pre komunikáciu s externým systémom poskytujúcim službu importu dát, nachádza aj formulár pre zmenu prístupového hesla do administrátorskej časti.

2.2 Rozhranie pre študentov

Rozhranie pre študentov je vytvorené web rozhraním cez stránku z časti prístupnej len z lokálnej siete Fakulty matematiky, fyziky a informatiky Univerzity Komenského.

2.2.1 Prihlásenie

Prvá stránka slúži na prihlasovanie do systému. Po prihlásení korektným menom a heslom, ktoré sú také isté ako na stránke prihlasovania sa na skúšky, sa študentovi zobrazí hlavná stránka.

2.2.2 Hlavná stránka

Hlavná stránka slúži na zobrazovanie informácií, ako sú práve prebiehajúce skúšky, výsledky zo skúšok atď. Štruktúra je nasledovná. Na stránke najvyššej úrovne je výpis všetkých predmetov aktuálnych pre študenta. To znamená predmety, na ktoré je zapísaný. Tieto predmety sú klikacie a po kliknutí na niektorý z nich sa zobrazia bližšie informácie (časť 2.2.3).

2.2.3 Informácie o predmete

Stránka s informáciami o predmete má tieto položky. Od prednášajúceho sa tu nachádza slovná poznámka k nemu, to znamená napríklad odporúčaná literatúra. Ďalej tu je počet otázok v teste, bodovanie (za úplnú správnu odpoveď, za podmnožinu správnej odpovede, za nesprávnu odpoveď a za nezodpovedanú otázku) a samozrejme hodnotenie (stupnica) a výsledná známka (teda posledná). Je tu tiež zoznam termínov (minulých aj budúcich). V prípade, že sa študent niektorého z tohto termínu zúčastnil, tak je tento termín klikací a po kliknutí naň sa zobrazí stránka s bližšími informáciami o termíne (časť 2.2.4). V prípade, že práve prebieha skúška z tohto predmetu, tak po kliknutí na zodpovedajúci termín sa spustí samotné testovanie. Toto však je možné iba z lokálnej siete Fakulty matematiky, fyziky a informatiky Univerzity Komenského. Špeciálnym typom skúšky je demoskúška, ktorá demonštruje, aké asi otázky z daného predmetu môžu študenti dostať a tiež slúži na oboznámenie sa s prostredím webovskej aplikácie. Vytvorenie demotestu je na učiteľovi (vytvára sa iba jeden) a nie je povinný.

2.2.4 Informácie o termíne

Na tejto stránke si študent môže prezrieť svoje výsledky. Ale nie podrobne aj s testom, ale iba počet bodov z každej otázky a celkový počet bodov spolu s pridelenou známkou. Okrem toho tu sú poznámky učiteľa k termínu a tiež poznámky ku konkrétnemu testu študenta.

2.2.5 Vypracovávanie testu

Po kliknutí na odkaz pre spustenie testu sa vyberie test pre tohto študenta z dopredu vygenerovaných testov. Test pozostáva z dvoch častí. Z prvej časti, so zoznamom otázok a druhej časti, kde sa zobrazuje otázka.

Aby sa zabránilo zneužitiu systému a niekto nevypracovával test pod iným menom ako svojim, tak sa po usadení študentov skontroluje ich identita na základe ISIC karty.

Hneď po usadení sú študenti vyzvaní, aby sa prihlásili. Potom je možnosť prihlasovania zablokovaná a bude možná jedine za prítomnosti proktora. Študenti, ktorí by skončili skôr, než je skontrolovaná ich identita, musia zavolať najprv proktora a ten ju skontroluje. Pri prerušení testovania (WC...) sa študenti neodhlasujú, ale majú možnosť

zablokovania vypracovávaného testu. V prípade väčšieho výpadku sa môže prihlasovanie opäť povoliť a bude opäť skontrolovaná identita študentov.

2.2.5.1 Zoznam otázok

V prvej časti sa nachádza zoznam všetkých otázok farebne rozlíšených (význam farebného rozlíšenia v časti 2.2.5.2). Kliknutím na niektorú z otázok (sú zobrazené ako číslo otázky a začiatok otázky kvôli úspore miesta, ale zároveň aby si študent vedel spomenúť, čo ktorá otázka aspoň zhruba obsahuje) sa v druhej časti otázka zobrazí celá. Kliknutím na tlačidlo "odoslať test" sa test odošle a už nie je možná jeho ďalšia úprava. Kliknutím na tlačidlo "locknúť test" sa test zablokuje a čaká na meno a heslo prihláseného používateľa.

2.2.5.2 Zobrazenie otázky

V druhej časti sa nachádza plné znenie testovej otázky spolu s možnými zaklikávacími odpoveďami typu a,b,c... Ďalej tu je textové pole určené na poznámku, ktorá sa bude môcť využiť napríklad na komentár k nejednoznačnej otázke a podobne. Tlačidlom "ulož" sa čiastkové odpovede spolu s komentárom uložia na server.

V tejto časti je aj základná navigácia (predchádzajúca a nasledujúca otázka) a možnosť "vyfarbiť" otázku, teda prideliť jej akýsi status. Tento napomáha študentovi označiť si nejakou farbou otázky, ktoré už má vypracované, alebo tie, ktoré má vypracované len z časti, tie ktoré vôbec nevie a podobne. Toto je čisto interná záležitosť študenta, pretože tieto statusy sa neberú do úvahy ani pri hodnotení, ani nikde inde. Samozrejmosťou je časomiera, ktorá ukazuje zostávajúci čas a čas strávený nad danou otázkou. Periodicky je kontrolované, či nevypršal čas, alebo či test už nebol ukončený. V prípade, že bude nutné rozpracovaný test znova načítať, bude sa to robiť tým istým spôsobom ako spúšťanie testu, len na serveri už budú uložené medzivýsledky a tieto sa vezmú do úvahy.

2.2.6. Hodnotenie testu

Po kliknutí na odoslať v sekcii vypracovávaní testu sa test začne okamžite hodnotiť. Všetky kritéria hodnotenia nastavuje skúšajúci pri generovaní testu. To znamená, že

program už len prečíta parametre uložené v databáze a na základe nich test ohodnotí. Každú otázku zaradí do kategórie (správne zodpovedaná otázka, podmnožina správnych odpovedí, nesprávna odpoveď, nezodpovedaná otázka) a na základe tejto kategórie určí počet bodov. Suma týchto bodov určuje celkový dosiahnutý počet bodov.

Po odoslaní testu sa študentom zobrazí hodnotenie a opravený test, teda budú vidieť celý test len na čítanie s vyznačenými študentovými odpoveďami a s označenými správnymi odpoveďami.

Študent je potom presmerovaný späť na stránku s informáciami o predmete (časť 2.2.3), kde pribudne už výsledok so známkou. Pre neho je daný termín už uzavretý, a preto tam možnosť na spustenie testu už nie je. Týmto pre študenta vlastne termín skončil a prípadne môže prísť na ďalší termín. Dodatočné informácie o tejto skúške, ako napríklad priradenie známky, rôzne úpravy zo strany skúšajúceho (penalizácia za podvádžanie, uznanie viacerých možných správnych odpovedí a podobne) nájde na už spomínanej stránke s informáciami o predmete.

2.3 Štruktúra učiteľského rozhrania

V tejto podkapitole je detailne rozobraná štruktúra sekcií učiteľského rozhrania. Sekcie sú zoskupené do troch skupín na základe toho, či sa na predmet viažu alebo neviažu alebo či obsahujú podsekcie na predmet viazané aj neviazané. Tie, čo sa neviažu, podávajú všeobecné informácie o učiteľovi. Viazané, naopak, dovoľujú spravovať rôzne nastavenia týkajúce sa konkrétnych predmetov.

2.3.1 Sekcie neviazané na predmet

2.3.1.1 Sekcia informácie

Sekcia informácie je považovaná za hlavnú stránku učiteľského rozhrania, preto je v tejto sekcii obsiahnuté aj samotné prihlasovanie a odhlasovanie. Pokiaľ je užívateľ neprihlásený, nezobrazí sa mu nič iné okrem prihlasovacieho formulára.

Priebeh prihlasovania:

Užívateľ vyplní položky meno a heslo v prihlasovacom formulári a odošle ich na server. Keďže v systéme vystupuje viacero rolí (učiteľ a proktor) a učiteľ má väčšie práva ako proktor, systém sa snaží najprv prihlásiť užívateľa ako učiteľa. Ak neuspeje, tak až potom ako proktora.

Po úspešnom prihlásení:

Sekcia informácií má ponúknuť prihlásenému užívateľovi informácie o prihlásení a informácie o urgentných udalostiach v predmetoch, ktoré sa ho týkajú. Medzi informácie o prihlásení patrí meno užívateľa a práva, ktoré má. Do urgentných udalostí spadá zoznam práve prebiehajúcich skúšok a zoznam budúcich skúšok. Ku každej práve prebiehajúcej skúške sa zobrazí odkaz na pokračovanie testovania, ktoré prepne užívateľa do testovacieho režimu, a odkaz na zoznam študentov daného termínu. Zoznam študentov je prístupný iba pre užívateľa, ktorý má práva učiteľa daného predmetu. Naopak voľba pokračovania testovania je prístupná aj proktorom. V zozname budúcich skúšok je to podobné. Každá buduca skúška obsahuje možnosti zobrazenia zoznamu študentov a spustenia testovania. Zoznam študentov je, podobne ako u práve prebiehajúcich skúšok, prístupný iba pre učiteľov. Spustenie testovania je prístupné aj pre proktora. Po urgentných udalostiach sa v sekcii informácie nachádza odkaz *importovať termíny skúšok*, ktorý z externého systému naimportuje zoznam termínov skúšok z predmetov, v ktorých vyučuje prihlásený učiteľ. Externým systémom sa myslí napríklad systém Félix používaný na prihlasovanie študentov na skúšky.

2.3.1.2 Sekcia predmety

Táto sekcia zobrazuje učiteľom zoznam predmetov, v ktorých vyučujú. Ku každému predmetu sa zobrazí odkaz [detaily] na detailnú správu predmetu, ktorú môže spravovať len učiteľ, preto je prístupná len pre učiteľov.

2.3.1.3 Sekcia proktori

Význam proktorov:

Proktori sú osoby dozerajúce na správny priebeh skúšky. Dozor na skúške nemusí robiť len učiteľ daného predmetu, preto bolo potrebné, aby vznikla daná funkcia. Proktorov si

vyberá každý učiteľ sám a do tejto funkcie sa môžu dostať buď iný učitelia alebo osoby vykonávajúce čisto funkciu proktorov. Pri výbere proktora, nadobúda proktor práva prístupu do sekcie testovania všetkých predmetov, ktoré vyučuje učiteľ, ktorý si daného proktora pridal.

Sekcia proktori sa delí na dve podsekcie:

- *zoznam proktorov* obsahujúci mená všetkých osôb, ktoré môžu na predmetoch, ktorých je prihlásený učiteľ učiteľom, vykonávať funkciu proktora. Ku každému sú dva odkazy – [detaily] a [odstrániť]. Prvý zobrazí informácie o osobe, druhý danú osobu zo zoznamu proktorov odstráni.
- *pridávanie proktorov* obsahuje zoznam osôb v systéme, ktoré môžu byť na funkciu proktora dosadené. V tomto zozname sa zobrazia mená osôb, ktoré vystupujú v systéme ako učitelia alebo ako proktori, ktorých si prihlásený učiteľ ešte do zoznamu proktorov nepridal. Každá osoba v zozname obsahuje takisto dva odkazy – [detaily] a [pridať]. Funkčná stránka prvého z nich zostáva nezmenená, zatiaľ čo druhý odkaz osobu do zoznamu proktorov pridá.

Ako už bolo vyššie naznačené, táto sekcia je prístupná len pre učiteľov.

2.3.1.4 Sekcia zmena hesla

Väčšina programov obsahujúcich autentifikáciu užívateľov zahŕňa v sebe aj možnosť zmeny hesla. Potreba meniť heslo môže vyplývať z nespokojnosti užívateľa so starým heslom alebo z pohľadu bezpečnosti. Formulár na zmenu hesla obsahuje tri vyplňovacie položky:

- *staré heslo*, potrebné na overenie totožnosti užívateľa
- *nové heslo*
- *zopakovanie nového hesla* znižujúce pravdepodobnosť preklepu pri zadávaní nového hesla

Pokiaľ sa totožnosť užívateľa overí a nové heslo sa zhoduje so zopakovaným heslom, nové heslo sa pre užívateľa uloží v databáze.

2.3.2 Sekcie viazané na predmet

2.3.2.1 Sekcia info o predmete

Všeobecné nastavenia predmetu sa uskutočňujú práve v tejto sekcii. Dovoľujú učiteľovi nastaviť popis k predmetu a dĺžku trvania skúšok. Popis k predmetu slúži na to, aby študenti získali bližšie informácie o predmete, ako napr. o ciele predmetu, spôsobe výučby, či podmienkach úspešného absolvovania predmetu. Dĺžka trvania skúšok sa nastavuje pre celý predmet jednotne, keďže sa predpokladá objektivnosť podmienok pre všetkých študentov. Prístup do tejto sekcie je povolený len učiteľom.

2.3.2.2 Časť testov

Najdôležitejšou časťou učiteľského rozhrania je práve časť týkajúca sa testov, ktorá má učiteľovi umožniť testy generovať, spravovať ich a nastavovať parametre potrebné pre časť testovania. Táto časť zahŕňa v sebe štyri sekcie: *hodnotenie*, *generovanie*, *demo test* a *importovanie*. Všetky, keďže sa netýkajú samotného testovania, sú prístupné iba pre učiteľa.

2.3.2.2.1 Sekcia hodnotenie

Hlavným cieľom nášho systému je ponúknuť učiteľovi aj študentom rýchle spracovanie testov. Aby program po vypracovaní testu vedel určiť, akú známku má dať študentovi, musí učiteľ zadať stupnicu známok a systém bodovania testových otázok. Oboje sa dajú nastaviť práve v tejto sekcii. V stupnici známok sa dá ku každej známke nastaviť spodná hranica bodov potrebná na jej dosiahnutie. Keďže je prípustné, že sa v testových otázkach môže nachádzať viacero správnych odpovedí, bodovanie otázok sa rozdeľuje do štyroch skupín:

- *celá správna odpoveď* – študent označil všetky správne odpovede a žiadne iné
- *čiastočná odpoveď* – študent označil neprázdnu podmnožinu správnych odpovedí
- *neodpovedaná otázka* – študent neoznačil ani jednu odpoveď
- *zlá odpoveď* – študent označil aspoň jednu odpoveď nepatriacu do správnych odpovedí

Učiteľ jednotlivým skupinám nastavuje počet bodov, ktoré študent získa za každú otázku patriacu do danej skupiny. Poradie vymenovania skupín určuje aj prioritu zaradenia odpovede do skupiny.

2.3.2.2.2 Sekcia generovanie

V sekcii generovanie prebieha už samotné generovanie testov a skladá sa z dvoch častí: *generovanie a nastavenie generovania*.

Nastavenie generovania:

Dovoľuje učiteľovi nastaviť spôsob generovania testov. Základnými parametrami sú: počet otázok a hodnoty *min* a *max* pre každý okruh tém z predmetu. Počet otázok intuitívne určuje počet otázok, z koľkých sa má test skladať. Hodnota *min* pri okruhu tém hovorí, koľko otázok minimálne sa má z daného okruhu alebo z jeho podokruhov vybrať. Hodnota *max*, naopak, určuje maximálny počet otázok. Okruhy tém sú vypísané hierarchicky, aby učiteľovi ponúkli prehľad o ich štruktúre. Algoritmicky si hierarchické zobrazenie okruhov tém vyžaduje čas $O(n \log n)$ na abecedné utriedenie okruhov tém + $O(n)$ na vytvorenie hierarchickej štruktúry, pričom n reprezentuje počet okruhov tém. Teda výsledná časová zložitosť algoritmu je $O(n \log n)$.

Generovanie:

Táto časť má za úlohu umožniť učiteľovi mazať už vygenerované testy a dovoliť mu nové testy generovať. Keďže testy môžu byť viacerých typov: papierové a vypracovávané pri počítači, správa testov je rozdelená do dvoch samostatných častí. Prvá týkajúca sa testov pri počítači, druhá papierových testov. Pri generovaní papierových testov sa výsledok generovania učiteľovi ukáže vo formáte PDF, z ktorého si testy môže vytlačiť. Bližšie informácie o papierových testoch sú popísané v sekcii importovanie. Ku každému typu testov je uvedený počet už vygenerovaných testov, tieto testy má učiteľ možnosť všetky vymazať kliknutím na tlačidlo *vymaž*. Ďalej nasleduje možnosť nového generovania. Učiteľ zadá počet testov, ktoré sa majú vygenerovať a kliknutím na tlačidlo *generuj* sa samotné generovanie na základe nastavení spustí. Pri testoch vypracovávaných pri počítači je v tejto podsekcii zobrazená ešte jedna voľba a to generovať demo test. Bližšie informácie o demo teste sa nachádzajú v sekcii demo

test. Všetky typy testov sa generujú rovnakým algoritmom a pred samotným generovaním sa do pamäte načítajú potrebné údaje získané funkciou *GetDataForGenerating*.

Tieto údaje sa v spomínanej funkcii upravujú tak, aby vznikol zoznam okruhov tém obsahujúcich zoznam podokruhov a zoznam otázok patriacich samotnému okruhu tém alebo jeho podokruhom. Jednotlivé otázky obsahujú zoznam sád odpovedí. Okrem týchto údajov je potrebné každému okruhu ešte vypočítať jeho hĺbku, ktorá je pri generovaní potrebná. Časová zložitosť algoritmu pripravujúceho dáta pre generovanie je $O(s + k.o)$, kde s reprezentuje počet sád odpovedí, o celkový počet otázok v predmete a k maximálnu hĺbku domén.

Popis algoritmu generovania testov:

Algoritmus obdrží dáta získané funkciou *GetDataForGenerating* a prebieha v dvoch krokoch: Najprv náhodne vyberá z každého okruhu toľko otázok, koľko určuje hodnota *min* okruhu. Výber otázok z okruhov tém musí prebiehať zostupne podľa hĺbky okruhu. V druhom kroku sa zo zoznamu otázok, ktoré zostanú, začne náhodne vyberať toľko otázok, aby počet otázok vybraných do testu spĺňal učiteľove požiadavky týkajúce sa počtu otázok v teste. Po zaradení otázky do testu sa náhodne vyberie z nej sada odpovedí, ďalej sa otázka zo zoznamu otázok okruhu tém, ktorému patrí, vymaže a vymaže sa taktiež zo zoznamov otázok všetkých nadokruhov, lebo je potrebné zabezpečiť, aby sme do testu nevybrali viackrát tú istú otázku. Počas behu algoritmu môže nastať situácia, že sa test na základe učiteľových požiadaviek vygenerovať nedá. Táto situácia môže byť spôsobená napr. tým, že v okruhoch neexistuje toľko otázok, koľko učiteľ zadal, aby sa do testu z neho vygenerovalo. Alebo že na základe hodnôt *min* každého okruhu sa do testu vygeneruje viac otázok ako je celkový počet otázok v teste. V prípade, že algoritmus nie je schopný vygenerovať žiaden test podľa učiteľových požiadaviek, systém o tom informuje učiteľa a generovanie testov neprebehne.

Časová zložitosť algoritmu generovania testov môže kolísať. Minimálne je $o(k.d + k.q)$, maximálne je $O(k.o)$, kde d určuje počet okruhov tém, q počet otázok, z ktorých sa má test skladať, o celkový počet otázok uložených v databáze a k maximálnu hĺbku okruhov tém.

2.3.2.2.3 Sekcia demo test

Aby mal študent pred skúškou možnosť zoznámiť sa so štýlom a s prostredím testovania, učiteľ má možnosť vytvoriť jednoduchý demo test, ktorý má slúžiť na tento účel. Testové otázky vôbec nemusia s predmetom súvisieť, jedná sa iba o ukážku testovania. Výsledky demo testov sa v databáze dlhodobo neukladajú, preto ich výsledok nijako nemôže ovplyvniť študentovu známku. Generovanie demo testu prebieha v sekcii generovanie, v tejto sekcii sa vygenerovaný demo test učiteľovi iba zobrazí. Pokiaľ je učiteľ s demo testom nespokojný, kliknutím na tlačidlo *vymaž demo test* sa demo test vymaže.

2.3.2.2.4 Sekcia importovanie

V sekcii generovanie pri voľbe generovania papierových testov sa vytvorené testy zapíšu do PDF dokumentu, ktorý si učiteľ neskôr môže vytlačiť. Rozvrhnutie papierového testu je nasledovné. V hlavičke sa nachádza názov predmetu, ku ktorému patrí, a jednoznačný identifikátor testu. V tele strany sa nachádza zoznam otázok a odpovedí k nim. Študent si pri vypracovávaní testu opíše jeho identifikátor, zaznačí si svoje ISIC číslo slúžiace na študentovu identifikáciu a ďalej zaznačí odpovede, ktoré si myslí, že sú správne. Po vypracovaní testu je potrebný import do systému, ktorý by študentov test vyhodnotil. Učiteľ, na základe študentami vypracovaných testov, vytvorí XML dokument reprezentujúci zoznam testov. Štruktúra XML dokumentu je daná DTD dokumentom '/dtd/tests.dtd'. Pri importe údajov sa zadáva termín skúšky, na ktorom sa import prevádza, a kontroluje sa, či identifikátor testu určuje test, ktorý bol vytvorený pre predmet patriaceho učiteľovi. Pokiaľ sú všetky údaje platné, výsledok testov sa zaznačí do databázy a ku každému testu sa prepočítajú body, ktoré študent získal. Nakoniec sa určí výsledná známka.

2.3.2.3 Sekcia termíny skúšok

Testovanie prebieha na skúškach, ktorých termíny sa importujú zo systému Félix používaného fakultou matematiky, fyziky a informatiky Univerzity Komenského na prihlasovanie študentov na skúšky. Keďže bol náš systém navrhovaný na spoluprácu so spomínaným systémom, vytváranie termínov skúšok nie je v našom systéme

obsiahnuté. Sekcia termíny skúšok obsahuje tri podsekcie: *zoznam termínov skúšok*, *zoznam študentov na termíne*, *pridávanie študentov na termín*.

Zoznam termínov skúšok:

Obsahuje zoznam termínov skúšok z aktuálne zvoleného predmetu. Tento zoznam sa dá jednoduchým filtrom upravovať. Filter dovoľuje učiteľovi nastaviť dátum, od ktorého sa majú termíny skúšok zobrazovať. Prihlásením učiteľa do systému sa nastaví tento dátum na dnešný. Zoznam termínov skúšok je utriedený vzostupne a obsahuje skúšky trojakého typu:

- budúca – spustenie testovania pre študentov daného termínu ešte neprebehlo
- prebiehajúca – testovanie študentov práve prebieha
- ukončená – testovanie už bolo ukončené

Pre každý termín skúšky sa zobrazí odkaz [zoznam študentov] odkazujúci do podsekcie zoznamu študentov na termíne a v prípade, že sa nejedná o ukončenú skúšku, zobrazí sa aj druhý odkaz. Budúcej skúške pribudne voľba [spustiť], ktorá spúšťa testovanie pre študentov zapísaných na termíne. Prebiehajúcej skúške sa naopak zobrazí voľba [pokračovať], ktorá prepne učiteľa do módu testovania, nakoľko sa testovanie študentov daného termínu už začalo.

Zoznam študentov na termíne:

Zobrazí sa zoznam prihlásených študentov na zvolenom termíne utriedený vzostupne podľa mena. K menu študenta pribudne odkaz [detaily], ktorý zobrazí detailné informácie o študentovi. Pokiaľ je skúška typu budúca, je učiteľovi dovolené upravovať zoznam študentov, a preto ku každému študentovi pribudne aj odkaz [odstrániť], ktorý príslušného študenta z termínu odstráni. Ďalej na konci stránky pribudne aj odkaz *pridať študenta*, ktorý zobrazí tretiu podsekciiu a to pridávanie študentov na termín. Pokiaľ termín skúšky prebehol, zobrazí sa odkaz [test], ktorý zobrazí študentov test.

Pridávanie študentov na termín:

Pokiaľ skúška ešte neprebehla, učiteľ má práva zoznam prihlásených študentov meniť. A práve pridávanie študentov sa deje v tejto podsekcii. Učiteľovi sa zobrazí podľa mena

utriedený zoznam študentov, ktorí majú daný predmet zapísaný a zároveň nie sú zapísaní na zvolenom termíne. Ku každému študentovi sa zobrazia dva odkazy: [detaily] a [pridať]. Prvý z nich zobrazí detailné informácie o študentovi, druhý študenta na termín pridá.

2.3.2.4 Sekcia databáza otázok

Databáza otázok je veľmi dôležitou súčasťou systému. Dovoľuje učiteľovi vytvárať, editovať a mazať otázky, z ktorých sa neskôr bude test generovať. Aby sa učiteľovi uľahčilo a sprehľadnilo spravovanie otázok, bolo potrebné zaviesť ďalšie štruktúry, na základe ktorých sú otázky členené. V sekcii databázy otázok celkovo nájdeme tri rozdielne entity: *okruhy tém*, *otázky* a *sady odpovedí*. S každou entitou sa dajú vykonávať tri akcie: *vytvoriť novú entitu*, *editovať* a *vymazať entitu*. Pri spravovaní databázy otázok je dôležité dávať si pozor na archiváciu. To znamená, že systém nedovolí učiteľovi vymazať z databázy otázku, ktorá už bola v teste použitá. Iba ju označí za zmazanú. Presný postup riešenia potrebnej archivácie je popísaný v odsekoch prislúchajúcich ku konkrétnym entitám.

Okruh tém:

Okruhy tém slúžia na zoskupovanie otázok do celkov, môžu mať stromovú štruktúru, tzn. že jeden okruh tém môže mať viacero podokruhov, tie sa tiež môžu deliť na ďalšie a ďalšie. Atribútami okruhu tém sú *názov* a *popis*. Názov určuje jeho pomenovanie, popis slúži na podrobnejšiu špecifikáciu (zobrazuje sa pri pohybe myši ponad názov okruhu tém). Pri vytváraní a editácii okruhu tém učiteľ zadáva hodnoty obom atribútom. Názov musí mať nenulovú dĺžku, popis ju nulovú mať môže. Pri editovaní okruhu tém nás archivácia nezaujíma. Zmena názvu, či popisu, nijako neovplyvní otázky. Mazanie okruhu tém je však zložitejšie. Keďže učiteľ zamýšľa vymazať nielen okruh tém, ale aj všetky podokruhy a otázky, ktoré im patria, vzniká nám zložitejší problém. Riešenie ako zabezpečiť archiváciu použitých otázok v teste je nasledovné: Vytvoríme si zoznam všetkých podokruhov patriacich okruhu, ktorý chceme vymazať. Do zoznamu pridáme aj samotný okruh tém, ktorý chceme vymazať. Pre každý okruh nachádzajúci sa v zozname zistíme, či v ňom existuje otázka, ktorá je v nejakom teste použitá. Ak nie, tak vymažeme všetky spomínané otázky a k nim aj všetky sady

odpovedí. Nakoniec vymažeme daný okruh. Ak sa však aspoň jedna otázka použitá v teste nájde, okruh tém sa označí za zmazaný a z databázy vymažeme iba tie otázky, ktoré v žiadnom teste neboli použité. Ostatné iba označíme za zmazané. S otázkami mažeme aj sady odpovedí, takisto iba tie, ktoré v testoch použité neboli. Ostatné iba označíme za zmazané. Túto procedúru vykonávame postupne pre všetky okruhy tém zo zoznamu.

Otázky:

Každá otázka patrí práve do jedného okruhu tém. Pri vytváraní otázky, ako aj pri editovaní, sa určuje *znenie otázky*, čo je jej jediným atribútom. Jedinou podmienkou je, aby bola dĺžka znenia otázky nenulová. Pri editovaní aj mazaní vznikajú archivačné problémy. Dôvod riešenia archivačných problémov v editovaní je z dôvodu uchovávaní presného znenia otázok v teste. Pokiaľ učiteľ zmení znenie otázky, ktorá bola v teste použitá, pôvodné znenie sa musí v databáze uchovať.

Postup pri editovaní otázky: Zistí sa, či bola otázka v nejakom teste použitá. Ak nie, staré znenie otázky sa v databáze prepíše novým. Ak však použitá bola, vytvoríme novú otázku s novým znením a k nej aj všetky sady odpovedí patriace k starej otázke. Stará otázka sa označí za zmazanú. Všetky sady odpovedí patriace k starej otázke a nepoužité v testoch sa z databázy vymažú. Tie, ktoré v testoch použité sú, sa označia za zmazané. Postup pri mazaní otázky: Takisto ako pri editovaní otázky, aj pri mazaní zistíme, či bola otázka v testoch použitá. Ak nie, tak ju aj so sadami odpovedí jednoducho vymažeme. Ak však áno, označíme otázku za zmazanú. Sady odpovedí, ktoré neboli v žiadnom teste použité, vymažeme. Ostatné označíme za zmazané.

Sady odpovedí:

Každá otázka môže existovať vo viacerých verziách, pretože môže obsahovať viacero sád odpovedí. Sada odpovedí sa skladá z odpovedí a je len na učiteľovi koľko odpovedí do nej pridá. Teoreticky ich môže byť ľubovoľne veľa, prakticky by však boli veľké množstvá odpovedí neopodstatnené. Keďže dopredu nevieme, koľko odpovedí si učiteľ do sady pridá, bolo potrebné naprogramovať formulár tak, aby dynamicky dokázal meniť svoju veľkosť. Zvažoval som dve možnosti ako potrebnú dynamickosť docieľiť:

- pomocou PHP

➤ pomocou JavaScriptu

Riešenie pomocou PHP sa ukazovalo byť neefektívne a zdĺhavé. Totiž každým pridaním novej odpovede do sady by bolo potrebné odosielať už vytvorené odpovede na server a od servera tieto údaje znovu prijímať. Zbytočne by sa plytvalo sieťovou komunikáciou, ktorá pri tom vôbec nie je potrebná, preto som zvolil riešenie pomocou JavaScriptu. JavaScript je skriptovací jazyk, ktorý sa vykonáva na strane klienta. Pomocou neho sa dá elegantne meniť obsah stránky a nevyžaduje sa sieťová komunikácia.

Už teda vieme spôsob, akým sa pridávajú a odoberajú odpovede v sadách odpovedí, teraz si popíšeme ich atribúty. Atribútmi odpovedí sú: *odpoved'* a *pravdivosť*. Odpoveď značí text odpovede a pravdivosť označuje, či daná odpoveď je pravdivá. Sady odpovedí sa dajú takisto editovať a mazať. Pri týchto akciách je potrebné pamätať na archiváciu.

Pri editovaní sa prezrie zoznam testov, či v nich nie je editovaná sada odpovedí použitá. Ak nie je, staré odpovede sa vymažú a vytvoria sa nové. Ak však použitá je, sada odpovedí sa označí za zmazanú, vytvorí sa nová a editované odpovede sa priradia k novej sade odpovedí.

Pri mazaní prebieha proces obdobne. Ak sa zistí, že sada odpovedí nebola v žiadnom teste použitá, jednoducho sa z databázy vymaže. Ak použitá bola, označí sa za zmazanú.

Pri zobrazovaní sa názov okruhov tém, otázok a sád odpovedí skrakuje a to z dôvodu, aby sa zamedzilo zhoršeniu dizajnu stránky spôsobeného vložením dlhého textu. Názov sa preto skrakuje na dĺžku štyridsiatich znakov a posledné tri znaky skráteného textu sú tri bodky.

2.3.2.5 Sekcia vyhľadávanie

Programy uchovávajúce a spracovávajúce údaje väčšinou v sebe zahŕňajú možnosť vyhľadávania údajov, ktoré umožňuje rýchlejší prístup k požadovanej veci. Preto sme sa rozhodli o zaradenie tejto sekcie do systému. Vyhľadávací formulár sa skladá z troch položiek. Prvá určuje hľadaný text. Pokiaľ je prázdna, zobrazí všetky hľadané objekty, inak zobrazí iba tie, ktoré obsahujú hľadané slovo. Druhá položka udáva typ objektu, v ktorých sa má vyhľadávať. Na výber sú štyri možnosti: *sady odpovedí*, *otázky*, *okruhy*

tém, študenti. Posledná, tretia, položka definuje typ triedenia. Triedenie môže byť: *vzostupné a zostupné*.

Rozdelenie hľadania podľa typu hľadaného objektu:

- *sada odpovedí* – výsledok hľadania zobrazí sady odpovedí, ktorých znenie odpovedí sa zhoduje s hľadaným textom. V zozname ku každej sade odpovedí pridá dva odkazy: [editovať] a [odstrániť]. Prvý prepne učiteľa do sekcie databázy otázok a zobrazí mu editovacie okno s danou sadou odpovedí. Druhý zavolá funkciu z databázy otázok na odstránenie sady odpovedí, avšak po vykonaní akcie prepne učiteľa naspäť do sekcie vyhľadávanie.
- *otázky* - výsledkom hľadania sú otázky, ktorých znenie sa zhoduje s hľadaným textom. Zoznam otázok je klikací a prepne užívateľa do sekcie databázy otázok, konkrétne zobrazí sady odpovedí k danej otázke. Ku každej otázke sú navyše dva odkazy: [editovať] a [odstrániť]. Ich funkčná stránka je obdobná ako u sád odpovedí, lenže obe akcie sa týkajú otázok.
- *okruhy tém* – vyhľadávanie zobrazí zoznam okruhov tém, ktorých názov sa zhoduje s hľadaným textom. V zozname výsledkov hľadania je učiteľovi umožnené sa prepnúť do sekcie databázy otázok, konkrétne do zoznamu podokruhov a otázok patriacich danému okruhu. Navyše každý okruh tém, ako v predchádzajúcich prípadoch, obsahuje dva odkazy: [editovať] a [odstrániť]. Ich funkcionality je podobná, akurát že objektami ich práce sú okruhy tém.
- *študenti* – vyhľadávanie zobrazí mená študentov, ktorí majú zapísaný daný predmet a ktorých mená súhlasia s hľadaným textom. K zoznamu študentov sa zobrazí odkaz [detaily], ktorý o študentovi zobrazí informácie.

2.3.2.6 Sekcia štatistika

Sekcia štatistika má za úlohu informovať učiteľa o rôznych štatistických údajoch vzťahujúcich sa k predmetu. Tieto údaje majú pomôcť učiteľovi odhaliť zlé nastavenia predmetu, akými napr. môžu byť príliš ľahké alebo príliš ťažké otázky v testoch. Umožnenie prístupu k týmto údajom prináša učiteľovi prehľad o predmete, ktorý vyučuje, a tým umožňuje jeho lepšie spravovanie. Štatistické údaje sa dajú získavať z rôznych uhlov pohľadu. My sme sa však snažili zahrnúť sem tie, ktoré sa nám zdali byť

najúžitečnejšie pre učiteľa. Sekcia štatistika sa delí na dve podsekcie: *štatistika podľa úspešnosti*, *štatistika zastúpenia známk*.

Štatistika podľa úspešnosti:

Zobrazuje objekty a k nim prislúchajúci priemerný počet bodov získaný v testoch. Pred zobrazením tohto typu štatistiky je ešte potrebné upresniť svoje požiadavky vyplňovacím formulárom, ktorý obsahuje štyri položky:

- Aký objekt je predmetom štatistického skúmania. Možnosťami sú: *sady odpovedí*, *otázky*, *okruhy tém*, *študenti*, *termíny*.
- Dátum určujúci obdobie, od ktorého sa zohľadňujú štatistické údaje.
- Dátum určujúci obdobie, do ktorého sa zohľadňujú štatistické údaje.
- Spôsob triedenia, ktorý výstup štatistiky triedi podľa priemerného počtu bodov. Na výber sú dve možnosti: *vzostupne* a *zostupne*.

Niekedy potrebujeme vymedziť časový úsek, z ktorého štatistika nás zaujíma. Pre tento účel slúžia druhá a tretia položka vyplňovacieho formulára.

Rozdelenie podsekcie podľa objektu štatistického skúmania:

- *sady odpovedí* – výsledkom štatistického skúmania je zoznam sád odpovedí patriacich predmetu a k nim pridelený priemerný počet bodov, ktoré študenti v testoch k daným sadám odpovedí získali. V zozname ku každej sade odpovedí pridá dva odkazy: [editovať] a [odstrániť]. Prvý prepne učiteľa do sekcie databázy otázok a zobrazí mu editovacie okno s danou sadou odpovedí. Druhý zavolá funkciu z databázy otázok na odstránenie sady odpovedí, avšak po vykonaní akcie prepne učiteľa naspäť do sekcie štatistika.
- *otázky* - výsledkom štatistického výstupu sú otázky patriace k predmetu, k nim pridelený priemerný počet bodov, ktoré študenti v testoch k daným otázkam získali. Zoznam otázok je klikací a prepne užívateľa do sekcie databázy otázok, konkrétne zobrazí sady odpovedí k danej otázke. Ku každej otázke sú navyše dva odkazy: [editovať] a [odstrániť]. Ich funkčná stránka je obdobná ako u sád odpovedí, lenže obe akcie sa týkajú otázok.
- *okruhy tém* – štatistika zobrazí zoznam okruhov tém, k nim prideli priemerný počet bodov, ktoré študenti v testoch k otázkam patriacim k daným okruhom

tém získali. V zozname výsledkov štatistiky je učiteľovi umožnené prepnutie do sekcie databázy otázok, konkrétne do zoznamu podokruhov a otázok patriacich danému okruhu. Navyiac každý okruh tém, ako v predchádzajúcich prípadoch, obsahuje dva odkazy: [editovať] a [odstrániť]. Ich funkcionálnosť je podobná, akurát že objektami ich práce sú okruhy tém.

- *študenti* – štatistika zobrazí mená študentov, ktorí majú zapísaný daný predmet, a priemerný počet bodov získaných v testoch. Pre každého študenta sa zobrazí odkaz [detaily], ktorý o študentovi zobrazí informácie.
- *termíny* – výsledkom je zoznam termínov skúšok z predmetu. Ku každému z nich je uvedený priemerný počet bodov, ktoré zúčastnení študenti daných termínov v testoch získali, a zoznam odkaz [zoznam študentov], ktorý učiteľa prepne do sekcie termínov skúšok a zobrazí mu zoznam študentov daného termínu.

Štatistika zastúpenia známk:

Zobrazuje tabuľku známk a k nim priradený počet testov, ktoré boli príslušnou známkou ohodnotené. Výber testov, ktoré sa majú brať do úvahy, je nastaviteľný podľa termínu skúšky. Učiteľ má aj možnosť zobraziť zastúpenie známk všetkých testov, tzn. bez ohľadu na termín. Zoznam všetkých termínov skúšok, patriacich predmetu, je zobrazený vo vyplňovacom formulári pre túto podsekciiu.

2.3.3 Sekcie viazané aj neviazané na predmet

2.3.3.1 Sekcia informácií o užívateľoch

Zaradenie sekcie informácií o užívateľoch do systému bola podmienená potrebou zobrazovať učiteľovi informácie o iných užívateľoch. Užívateľom sa myslia študenti, proktori a iní učitelia. Keďže informácie o proktoroch a o učiteľoch sú podávané rovnako, proktorov aj učiteľov budeme v tejto sekcii označovať jednotne a to proktori. Keďže je každý typ užívateľov v inom vzťahu k predmetu, ktorý prihlásený učiteľ vyučuje, niektoré podsekcii viazané na predmet sú, iné nie sú. Sekciu si rozdelíme na štyri podsekcii: *informácie o proktorovi*, *informácie o študentovi*, *zoznam testov* a *test*. Prvé z nich podáva informácie o proktoroch, ďalšie tri sa týkajú študentov. Nakoľko

obsah tejto sekcie je považovaný iba ako doplnok iných sekcií, v menu neexistuje položka na zobrazenie sekcie informácií o užívateľoch a bolo potrebné vytvoriť mechanizmus, ktorý by zabezpečoval, aby sa obsah tejto sekcie javil učiteľovi ako súčasť inej sekcie, z ktorej bola sekcia informácií o užívateľovi zavolaná. Pre spomínaný cieľ sa pri volaní tejto sekcie udávajú hodnoty parametrov *menuSet* a *menuIndex*, ktoré sú posielané serveru metódou GET protokolu HTTP. MenuSet určuje typ menu, ktorý sa má zobrazit' a menuIndex hovorí o tom, ktorá položka z menu sa má označiť za práve zobrazovanú.

Informácie o proktorovi:

Podáva základné informácie o proktoroch akými sú: meno proktora, jeho ISIC/ITIC číslo, zoznam predmetov, v ktorých vystupuje ako učiteľ a zoznam predmetov, v ktorých vystupuje iba ako proktor. Tieto údaje majú byť učiteľovi nápomocné hlavne pri pridávaní proktora. Bližšie informácie o význame proktorov sú uvedené v sekcii proktori.

Informácie o študentovi:

Zobrazuje základné informácie o študentovi. Medzi tieto informácie patrí: meno, ISIC číslo a zoznam zapísaných predmetov. Pokiaľ prihlásený učiteľ vyučuje predmet, ktorý má študent zapísaný, vedľa názvu predmetu sa zobrazí aj odkaz [testy], ktorý prepne učiteľa do podsekcie zoznamu testov príslušného predmetu.

Zoznam testov:

Informuje o všetkých testoch, ktoré študent napísal. Zoznam je utriedený podľa dátumu skúšky zostupne. Každý záznam tabuľky testov zobrazuje dátum skúšky, známku, ktorú študent za daný test dostal, a odkaz [details], ktorý presmeruje učiteľa do podsekcie test, a tým zobrazí details príslušného testu. Okrem zoznamu testov sa na stránke, kvôli prehľadnosti, zobrazí aj meno študenta a názov predmetu, ktorého zoznam testov sa práve zobrazuje.

Test:

Táto podsekcia zobrazuje jeden konkrétny test, ktorý študent vypracoval. Okrem vypracovaného testu, z dôvodu prehľadnosti, zobrazuje aj všeobecné informácie o študentovi, akými sú: meno študenta, predmet (z ktorého je test vypracovaný), dátum skúšky (kedy prebiehal test), známka z testu a súčet počet bodov získaných v jednotlivých testových otázkach. Učiteľ má možnosť k celému testu dopísať komentár, ktorý sa v databáze uloží a neskôr si ho študent môže prečítať. Za všeobecnými informáciami nasleduje samotný test zložený z testových otázok, komentáru k nim, počtu bodov získaných z otázky a k otázke prislúchajúcich odpovedí. Komentár píše študent z dôvodu nejednoznačnosti otázky alebo keď má nejaké pripomienky. Učiteľ môže neskôr komentár zvážiť a zmeniť bodovanie otázky. K odpovediam je znázornené, ako ich študent označil a či ich označil, resp. neoznačil, tak ako mal. Pri zmene bodovania a po kliknutí na tlačidlo *Ulož body* sa body k jednotlivým otázkam v databáze uložia a známka z testu sa na základe škály známok a nového bodovania prepočíta.

Kapitola 3

Triedy

PHP, takisto ako väčšina programovacích jazykov, umožňuje objektovo orientované programovanie, ktoré prináša programátorom výhody hlavne v prehľadnosti a logickej štruktúre programu. Preto je aj náš softvér naprogramovaný týmto spôsobom. Pred začatím implementácie konkrétneho rozhrania sme navrhli triedy, ktoré sú potrebné pre všetky tri rozhrania. Bližšie informácie o týchto triedach sú popísané v časti 2.1. Ostatné triedy, typické len pre učiteľské rozhranie, sú spomenuté v časti 2.2.

Definícia *Trieda* je konštrukcia, ktorá sa používa na zoskupenie príbuzných premenných a metód.

Definícia *Dedičnosť* je spôsob tvorenia nových tried používajúc triedy, ktoré už boli definované. Nové triedy, nazývané tiež odvodenými triedami, zdedia atribúty a metódy rodičovskej triedy, teda z triedy ktorej dedia.

Definícia *Abstraktná trieda* je trieda reprezentujúca rodičovskú triedu, ktorá abstraktne určuje štruktúru triedy. Samotná implementácia je definovaná až v odvodených triedach.

3.1 Spoločné triedy pre všetky rozhrania

3.1.1 cAbstractDB class

cAbstractDB je abstraktná trieda, ktorá má za úlohu abstraktne definovať rozhranie na prácu s databázou. Štruktúra definuje základné metódy, ktorými sú: vytvorenie a zrušenie spojenia s databázou, vykonanie dotazu a získanie informácií z databázy. Implementácia metód sa nachádza v odvodených triedach, ktorými sú cMySQLDB a PostgreSQLDB. Náš systém sme majoritne programovali pre komunikáciu s MySQL databázou, ale definovanie triedy cPostgreSQL umožňuje ľahší prechod na PostgreSQL databázový systém.

3.1.2 cMySQLDB class

cMySQLDB je trieda dediaci štruktúru abstraktnej triedy cAbstractDB a zároveň implementujúca metódy na spoluprácu s MySQL serverom.

3.1.3 cPostgreSQLDB class

cPostgreSQLDB je trieda takisto dediaci štruktúru abstraktnej triedy cAbstractDB, ale implementácia metód sa orientuje na PostgreSQL databázu.

3.1.4 cApplication class

Trieda `cApplication` je definovaná ako hlavná trieda, ktorá rozhoduje o tom, ktorá stránka sa spustí. Systém má definované tri rozhrania (administrátorské, študentské, učiteľské), pre každé z nich potrebujeme iný riadiaci program, preto je trieda `cApplication` definovaná abstraktne. Z nej sú zdedené tri triedy (`cAdminApplication`, `cStudentApplication`, `cTeacherApplication`), pre každé rozhranie jedno. Jediná metóda, ktorá je v `cApplication` implementovaná, je *Run*. Tá má za úlohu spracovať údaje od užívateľa, abstraktnou funkciou *GetPageById* zistiť, ktorá stránka sa má zobrazit' a následne jej prenechať riadenie.

3.1.5 `cPage` class

Všetky rozhrania sa skladajú z niekoľkých sekcií – stránok, ktoré však majú niektoré črty spoločné. Preto sme definovali abstraktnú triedu `cPage`, ktorá reprezentuje ľubovoľnú stránku ľubovoľného rozhrania. Štruktúra každej stránky sa dá rozdeliť do 2 častí:

- spracovávanie údajov
- príprava dát potrebných na zobrazenie

Na základe toho sme definovali 2 abstraktné metódy: *Process* a *Draw*.

3.1.6 `cMenu` a `cMenuItem` class

Máloktorý rozsiahlejší systém by sa zaobišiel bez menu, ktoré ponúka užívateľovi prehľadnosť štruktúry programu. Na správu menu položiek sme si naprogramovali triedy `cMenu` a `cMenuItem`. Prvá trieda reprezentuje menu ako celok, druhá reprezentuje jednu konkrétnu položku v menu. Metódy oboch tried dokážu pridávať alebo vyhľadávať menu položky. Spôsob, akým sú triedy naprogramované, umožňuje vytvárať štruktúrované menu. Každá trieda reprezentuje spájaný zoznam ďalších menu položiek, ktoré obsahujú ďalší spájaný zoznam. Rádovo sa dokáže vytvoriť menu s ľubovoľnou hĺbkou, avšak menu učiteľského rozhrania má maximálne hĺbku dva, preto je aj Smarty šablóna určená na vykresľovanie menu ohraničená touto hĺbkou. Zväčšiť hĺbku menu nie je problém, ale je to zbytočné.

3.2 Triedy v učiteľskom rozhraní

3.2.1 cTeacherApplication class

Trieda `cTeacherApplication` má úlohu riadiacej jednotky v učiteľskom rozhraní a vznikla dedením z abstraktnej triedy `cApplication`. Pri inicializácii objektu tejto triedy sa nakonfiguruje šablónovací systém `Smarty`. `cTeacherApplication` ďalej implementuje metódu `GetPageById`, ktorá má na základe užívateľových požiadaviek vybrať stránku, ktorú užívateľ požaduje. Táto informácia je systému posielaná metódou `GET` protokolu `HTTP` tým, že definuje hodnotu parametra `site`. Metódu `Run`, ktorá bola v rodičovskej triede implementovaná, z funkčnej stránky nespĺňala požiadavky učiteľského rozhrania, preto bolo potrebné danú metódu preprogramovať. Pôvodná metóda vybrala stránku metódou `GetPageById` a následne sa zavolali metódy `Process` a `Draw` vybratej stránky. Avšak sekcie vyhľadávanie a štatistika vyžadujú, aby sa dala vykonať `Process` metóda jednej stránky, ale `Draw` metóda druhej stránky. Preto som naprogramoval všeobecný mechanizmus, ktorý toto umožňuje a preto bola potrebná nová implementácia `Run` metódy.

3.2.2 cTeacherPage class

Keďže si každé rozhranie vyžaduje rozdielnu funkcionálnu svojich stránok, bolo potrebné, aby si každý z nás zadefinoval vlastnú abstraktnú triedu definujúcu stránky svojho rozhrania. `cTeacherPage` je takouto triedou a definuje základné vlastnosti stránok učiteľského rozhrania, medzi nimi aj základnú funkcionálnu vykresľovacej metódy `Draw`, ktorá pokiaľ zistí nezrovnalosti medzi právami užívateľa a požadovanými entitami, zobrazí stránku informujúcu o odopretí prístupu. Ostatné metódy nachádzajúce sa v `cTeacherPage` môžeme rozdeliť do dvoch skupín:

- metódy zisťujúce práva užívateľa ku konkrétnym entitám (predmetom, skúškam, testom, otázkam, ...)
- metódy zisťujúce často potrebné údaje (názov aktuálneho predmetu, ...)

3.2.3 Menu triedy učiteľského rozhrania

Existencia viacerých rolí v systéme, ako sú učitelia a proktori, prináša aj rozdielnosť aktivít, ktoré môžu jednotlivé role vykonávať. Preto je potrebné udržiavať pre každú rolu iné menu. V učiteľskom rozhraní existujú štyri triedy definujúce rozdielne menu:

- *cMainTeacherMenu* – obsahuje zoznam sekcií pre učiteľa neviazaných na predmet
- *cCourseTeacherMenu* – obsahuje zoznam sekcií pre učiteľa viazaných na predmet
- *cMainProctorMenu* – obsahuje zoznam sekcií pre proktora
- *cTestingMenu* – obsahuje zoznam sekcií testovania

Každé zo spomenutých tried menu vzniklo odvodením z triedy *cMenu* a všetky rozširujú rodičovskú triedu o metódu *GenerateMenu*, ktorej zavolanie vygeneruje všetky položky v menu.

3.2.4 Triedy stránok učiteľského rozhrania

Najväčší počet tried patrí práve do tejto podkapitoly. Tieto triedy po funkčnej stránke definujú sekcie učiteľského rozhrania.

Patria sem triedy:

cCourseInformationPage, cCoursePage, cDemoTestPage, cEvaluationPage, cExamTermsPage, cChangePasswordPage, cInformationPage, cProctorsPage, cQuestionDBPage, cSearchPage, cStatisticsPage, cTestGeneratingPage, cTestImportPage, cTestingPage, cUserInformationPage

Všetky vznikli odvodením z rodičovskej triedy *cTeacherPage* a implementujú metódy *Process* a *Draw*, ktoré obsluhujú spracovávanie údajov od užívateľa a prípravu dát potrebných na grafický výstup. Trieda *cTeacherApplication* rozhoduje o tom, ktorá stránka má byť načítaná a ktorej má byť odovzdané riadenie. Väčšina stránok sa ďalej ešte delí na drobnejšie celky, podsekcie. O tom, ktorá podsekcia sa má zobrazit', rozhoduje už každá stránka sama. Informácia o tom, ktorá podsekcia sa má užívateľovi

zobraziť, sa posiela metódou GET protokolu HTTP tým, že definuje hodnotu parametra *action*.

Každá trieda má iné požiadavky na práva užívateľa. Niektoré stránky vyžadujú iba práva učiteľa, iné zase dovoľujú prístup aj proktorom. Ďalšie sú ďalej viazané na predmet, pri nich sa zisťuje patričnosť učiteľa/proktora k predmetu. Niekedy je potrebná ešte podrobnejšia analýza práv, napr. pri skúškach, testoch, keď sa zisťuje či sa skúška alebo test viaže k danému predmetu. Zisťovanie práv sa deje v obidvoch vyššie spomenutých metódach.

Kapitola 4

Bezpečnosť

Bezpečnosť je jednou z dôležitých črt programu a v našej práci je takmer nevyhnutná. Rozoberieme si ju zo štyroch rozdielnych uhl'ov pohľadu. V prvom z nich budeme brať používateľa za potencionálneho záškodníka, ktorý chce náš systém poškodiť alebo prinútiť ho správať sa inak. V druhom prípade rozoberieme, ako je chránený prístup k jednotlivým sekciám učiteľského rozhrania pre rôznych užívateľov. V treťom prípade sa budeme zaoberať uhádnuteľnosťou zahašovaného hesla. A nakoniec si povieme o tom ako funguje správa session.

4.1 Ochrana pred útokmi

4.1.1 SQL injection

Popis:

SQL injection je technika, ktorá využíva zraniteľnosť programov využívajúcich databázové systémy. Útok môže byť úspešný iba v prípade, že program nefiltruje vstup od užívateľa a ten je rovno posúvaný databázovému serveru. Keďže vstup nie je filtrovaný, je možné, aby užívateľ do formulára zadal príkazy, ktoré je databázový server schopný interpretovať.

Príklad:

```
sql = "SELECT * FROM order WHERE name = " + orderID + ";"
```

Ak však užívateľ zadá hodnotu premennej `orderID="1 OR 'a'='a"`, vyselektujú sa všetky hodnoty z tabuľky `order`. Alebo ak by zadal `orderID="1;DROP TABLE order"`, vymaže celú tabuľku `order` z databázy.

Dôsledky:

Ako je aj z predchádzajúceho príkladu vidieť, dôsledky útoku SQL injection sú rôzne a môžu byť až fatálneho rozmeru. Útočník môže získať informácie, na ktoré nemá práva, ba môže až vymazať informácie z databázy. Preto je veľmi dôležité uvedomiť si prítomnosť daného problému a začať ho riešiť.

Zabránenie útoku:

Riešenie ako zabrániť tomuto typu útoku vôbec nie je ťažké. Zakaždým ako posielame požiadavku databázovému serveru, filtrujeme vstup od užívateľa, aby neobsahoval špeciálne znaky a riadiace inštrukcie databázového systému. Funkcie na filtrovanie sú už väčšinou v databázovom systéme naprogramované, programátorovi teda stačí iba danú funkciu zavolať. Konkrétne v MySQL sa funkcia volá *mysql_real_escape_string*. Každý vstup posielaný databázovému serveru som touto funkciou aj odfiltroval.

4.1.1 HTML injection

Popis:

HTML injection, takisto ako aj SQL injection, je ďalším variantom útoku známeho ako *Cross site scripting* (XSS). Útočník chce vďaka neošetreným chybám v zabezpečení systému podstrčiť programu svoj vlastný kód, ktorý je následne vykonaný. HTML injection sa konkrétne orientuje na vkladanie špeciálnych HTML znakov.

Príklad:

Útočník vkladá na fóre príspevok typu: `"XYZ"` alebo

"".

Pokiaľ nie sú príspevky vo fóre odfiltrované od špeciálnych HTML znakov, tak v prvom prípade sa zobrazí v príspevku odkaz, v druhom sa zobrazí obrázok. Potom je už iba na útočníkovi/záškodníkovi, aký odkaz, resp. obrázok do fóra pridá.

Dôsledky:

Prinajmenšom sú dôsledky badateľné iba na výzore stránky, napr. rozhádzaný dizajn. V horšom prípade pribudnú príspevky s nevhodným obsahom. Toto je častokrát neprípustné najmä pre solídne firmy, ktorým by nevhodný obsah zhoršil reputáciu.

Zabránenie útoku:

Pri spracovávaní údajov zadávaných od užívateľa treba odfiltrovať špeciálne znaky (<, >, &), ktoré by mohli určovať HTML elementy. PHP už takúto funkciu má v sebe zabudovanú, stačí ju iba na príslušnom vstupe od užívateľa zavolať. Všetky vstupy od užívateľa, ktoré sa ukladajú do databázy, som ošetril funkciou *htmlspecialchars*, ktorá zabráni HTML injection.

4.2 Prístupové práva k sekciám programu

Každá sekcia učiteľského rozhrania si vyžaduje určité práva. Právami sa myslí napríklad, či je užívateľ prihlásený. Ak áno, či je prihlásený ako učiteľ alebo ako proktor. Niektoré sekcie sa viažu na predmet, vtedy sa overujú užívateľove práva spravovať daný predmet a podobne.

Práva si rozdelíme do niekoľkých skupín a bližšie si určíme k akým sekciám programu majú prístup:

➤ Neprihlásený užívateľ:

Nemá práva vstúpiť do žiadnej sekcie, vždy sa mu zobrazí len úvodná prihlasovacia stránka. Každá sekcia spracováajúca dáta, okrem sekcie prihlasovania, pokiaľ prijme údaje od neprihláseného užívateľa, ignoruje ich.

➤ Užívateľ prihlásený ako učiteľ:

Učiteľ má prístup ku všetkým sekciám, ktoré nie sú viazané na predmet. K tým, ktoré sa na predmet viažu, musí byť učiteľ v zozname učiteľov daného predmetu. Inak sa prístup odoprie. Učiteľ má prístup ku všetkým sekciám predmetu, ktorý vyučuje. Niekedy však potrebujeme previesť analýzu podrobnejšie a to v sekciách, ktoré podrobnejšie špecifikujú predmet. Napríklad v časti termínov skúšok sa kontroluje, či daný termín patrí prihlásenému učiteľovi a podobne. Učiteľ však môže mať na niektoré predmety práva iba ako proktor. V takom prípade má užívateľ prístup iba k sekcii testovanie daného predmetu.

➤ Užívateľ prihlásený ako proktor:

Proktor slúži ako dozor na skúške, preto zo sekcií viazaných na predmet má práva iba na testovanie. Takisto sa kontroluje, či má práva proktora daného predmetu. Zo sekcií neviazaných na predmet má prístup len k informačnej stránke a zmene hesla.

4.3 Bezpečnosť uloženia hesla v databáze

Autentifikácia užívateľa prebieha na základe overenia hesla. Keďže užívateľ nadobúda všetky práva viazané na svoju osobu len vďaka správnosti zadania hesla, dá sa heslo považovať za najcitlivejší údaj v databáze. Preto by mali byť heslá v databáze uložené spôsobom, ktorý by minimalizoval úspešnosť ich zistenia a zároveň by boli stále použiteľné pri autentifikácii. Z tohto dôvodu sme sa rozhodli neukladať samotné heslá, ale haše hesiel. Hašovacie funkcie sú funkcie, ktoré transformujú dáta na reťazec pevnej dĺžky. Ich vlastnosťou vhodnou na kryptovanie je, že sa spätne nedajú zistiť pôvodné dáta. Resp. dajú, ale iba použitím hrubej sily. Ukladaním hašov a nie samotných hesiel v databáze sme zvýšili utajenie hesla pri možnom úniku dát z databázy. Na internete sa ale dajú nájsť servery, ktoré obsahujú databázu hašov často používaných užívateľských hesiel. Preto aj tento systém môže mať svoje slabiny. Aby sme tomuto predišli, dalo by sa prinútiť užívateľov nezadávať krátke a ľahko uhádnuteľné heslá, ale tento spôsob by

bol pre užívateľov nepríjemný. Existuje však aj iné riešenie, ktoré dovoľuje užívateľom nastaviť si ľubovoľné heslo a zároveň znižuje rozlúštiteľnosť hašu. Pointa spočíva v tom, že sa v databáze neukladá haš hesla, ale haš hesla s náhodným 6 znakovým prefixom. Týmto sa nie len zťažuje uhádnuteľnosť zahašovaného hesla, ale sa aj zabezpečí, že dvaja užívatelia, ktorí používajú rovnaké heslo, budú mať rôzne haše. To je ako dôsledok rozdielného prefixu, ktorý má každý užívateľ iný. V našom systéme sme naimplementovali posledné spomínané riešenie. Hašovaciu funkciu sme použili MD5.

4.4 Správa session

Po prihlásení užívateľa do systému sa vygeneruje náhodný haš - SESSIONID, ktorý identifikuje užívateľa. Na základe neho si dokáže užívateľ ukladať informácie na serveri. Prednastavená správa session funguje tak, že sa tieto informácie ukladajú do súborov. My sme ju však preprogramovali tak, že sa ukladajú do databázy. Takto sme získali zabezpečenie, že sa k údajom dostane len človek, ktorý pozná prihlasovacie údaje do databázy. Implementácia správy session je uložená v súbore *session.php*.

Literatúra

[1] <http://linux.wku.edu/~lamonml/algor/sort/sort.html>