

Semestrální práce

**Vytvoření registračního systému ke zpracování
příhlášek k seminářům na Gymnáziu Jana Nerudy**

Autor:	Marek Szeles
Škola:	ČVUT, Fakulta elektrotechnická
Studijní program:	Softwarové inženýrství a technologie
Školní rok:	2015/2016
Semestr:	Zimní
Předmět:	Prezentace (B6B04PRE)

Prohlášení

Prohlašuji, že jsem svou semestrální práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v kapitole použitá literatura, či v poznámkách pod čarou.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Praze dne 31.12.2015

podpis:

Marek Szeles

Anotace

Cílem této práce je vyřešit přetrvávající problém s vytvářením rozvrhů na Gymnáziu Jana Nerudy. V teoretické části je popsán problém, s různými stávajícími možnostmi řešení. Po vybrání vhodného řešení je v praktické části popsáno vytváření samotného systému registrací studentů, konkrétně na povinně-volitelné semináře ve školním roce 2016/2017 za pomoci webové stránky a soustavy databází.

Klíčová slova:

Semináře, Registrační systém, Analytika, Databáze, PHP, MySQL

Anotation

The aim of the paper is to solve the current issue of creating student timetables on the Jan Neruda Gymnasium in Prague. In the theoretical part the issue is described, with multiple possible solutions. After the correct approach is chosen, the creation of an administrative system is described in the practical part of the paper – specifically in the case of seminar registration for the academic year 2016/2017. The system is created in the form of a webpage, using a system of databases to store data.

Keywords:

Seminars, Registration system, Analytics, Databases, PHP, MySQL

Obsah

Obsah.....	4
1. Úvod.....	5
1.1. Řešení ve školním roce 2014/2015 a 2015/2016.....	6
2. Zvolené řešení pro nadcházející školní rok 2016/2017	7
2.1. Představení použitých nástrojů.....	9
2.1.1. Html	9
2.1.2. CSS.....	9
2.1.3. PHP	9
2.1.4. MySQL.....	9
3. Popis systému a proces jeho vzniku.....	10
3.1. Funkčnost systému	10
3.1.1. Řadový uživatel - student	11
3.1.2. Administrátor – učitel zpracovávající semináře.	11
3.2. Programování.....	12
3.2.1. Bezpečnostní prvky.....	13
3.2.2. Testování	15
4. Využití systému v praxi.....	15
4.1. Stav využívání	15
5. Závěr	15
6. Přílohy.....	16
6.1. Příloha č.1 – Zadání práce	16
6.2. Příloha č.2 – Vizualizace webu na různých přístrojích a prohlížečích.....	16
6.3. Příloha č.3 – Uživatelská příručka pro studenty.....	16
6.4. Příloha č.4 – Uživatelská příručka pro administrátory (učitele)	16
6.5. Příloha č.5 – Kód programu.....	16

1. Úvod

Na konci školního roku 2013/2014 se na Gymnáziu Jana Nerudy po dlouhodobém nátlaku ze strany studentů rozhodlo, že se změní struktura výuky v posledních dvou ročnících studia, tedy v kvintě a sextě. Nová koncepce organizace studia má připomínat spíše studium na vysoké škole – základ rozvrhu pro studenty tvoří pouze nutné předměty s definovanou hodinovou dotací v ŠVP¹, zbytek si pak student volí sám podle svých zájmů a předpokládaného zaměření při maturitě. V případě nastávajících pátých (předmaturitních) ročníků bylo nutné si vybrat pět povinně volitelných seminářů, v případě nastávajících maturantů bylo seminářů rovnou sedm. Bylo však zároveň jasné, že primární preferovaný výběr studentů nemůže být vždy z praktických důvodů dodržen. Mohlo se předpokládat, že pluralita studentstva bude veliká, a že na mnoho seminářů se přihlásí velice malý počet žáků, tudíž se tyto semináře nebudou moci otevřít. Naopak mohl také nastat problém opačný – na některé semináře se mohlo přihlásit studentů příliš, a seminář by musel být rozdělen. I po zvolení preferovaných seminářů však posléze byla dalším problémem následná koordinace vybraných seminářů podle časových možností lektorů a podle toho, zdali se některé semináře vybrané jednotlivými studenty nepřekrývají.

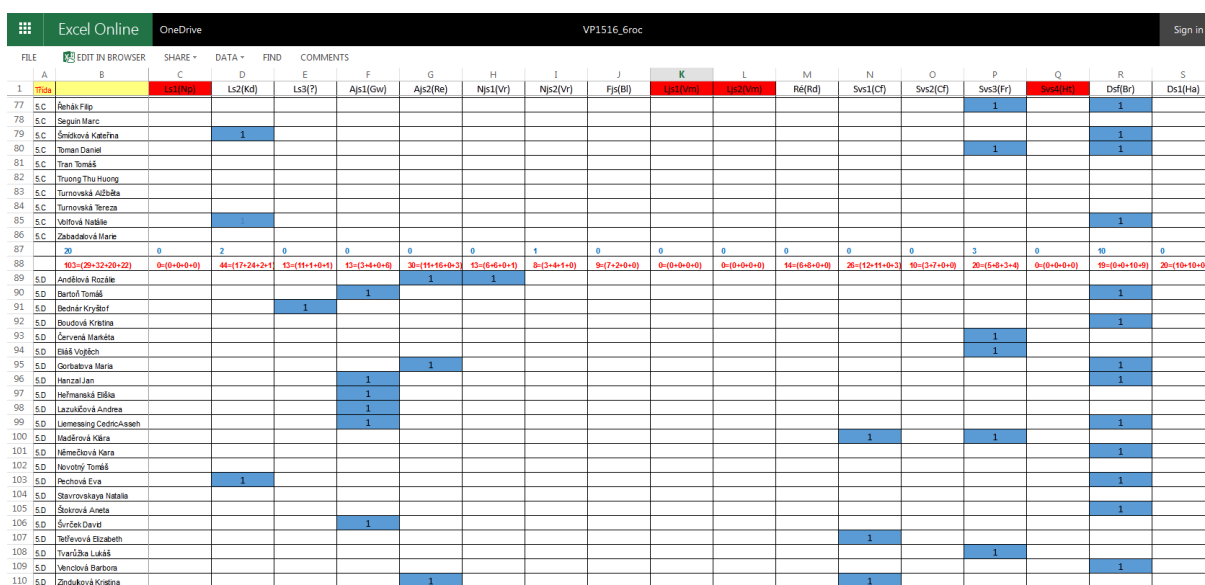
Největším problémem pro vedení školy však bylo získat samotné preferované semináře od studentů. Vzhledem k tomu, že nebyl k dispozici žádný specializovaný program, muselo se přistoupit k provizornímu řešení, které nebylo ideální a vzhledem k administrativní náročnosti se ani nepředpokládala jeho dlouhodobá udržitelnost. Hned posléze se tedy začala hledat jiná řešení.

Cílem této maturitní práce je popsat klíčové aspekty vzniklého problému, analyzovat starý systém registrací a posléze pomocí nabytých znalostí navrhnout nový systém určený primárně pro funkci registrace seminářů a opsat jeho vznik a funkčnost.

¹ ŠVP – Školní vzdělávací program [online] Dostupné z: http://cs.wikipedia.org/wiki/%C5%A0koln%C3%AD_vzd%C4%9Bl%C3%A1vac%C3%AD_program

1.1. Řešení ve školním roce 2014/2015 a 2015/2016

Vzhledem k tomu, že času od rozhodnutí o změně organizace studia do nutného rozdělení studentů bylo relativně málo, bylo využito primitivní a neefektivní, zato snadné a funkční řešení. Využilo se nových služeb sdílení dokumentů přes internet na službě Excel Online, kam se měli studenti postupně zapisovat do tabulky obsahující jejich jména a zkratky nabízených seminářů (Seznam seminářů s vysvětlením zkratk byl distribuován separátně). Přístup k tomuto dokumentu byl distribuován mailem a studenti měli v dané tabulce v řádku označeným svým jménem vyplnit v políčku ve sloupci pod vybranými semináři jedničku, a vyplněná pole posléze barevně označit.



Obrázek 1: Tabulka použitá pro zpracování volby seminářů studenty v minulých letech

Toto řešení mělo hned několik jasných nedostatků. Prvním z nich je bezpečnost – sdílený dokument byl otevřený každému, kdo na něj dostal odkaz. Díky tomuto opatření se sice přístup k dokumentu velice jednoduše šířil pomocí e-mailu, ale jakmile by se k němu dostal někdo nepovolaný, velice snadno by mohl celou tabulku smazat. Stejně tak jednotliví studenti viděli, které semináře mají zapsané jejich spolužáci a mohli je velice jednoduše upravit. S tím je svázána i velice špatná uživatelská přístupnost systému – v ohromné tabulce studenti lehko ztratili přehled, a tak mohli velice snadno něco omylem smazat, či svou volbu zapsat do špatného políčka, omylem si zvolit více seminářů, než je povoleno, a podobně. Vzhledem k tomu, že v tabulkách Excel online není možné omezit zadávaná data, možnosti různých chyb jsou doslova nekonečné.

V neposlední řadě nesmíme opomenout ani administrátorskou zátěž spojenou s provozem tabulkového systému. I přes své nedostatky byla tabulka složena ze složité soustavy vzorců a přesměrování, nemožnost omezit zadávaná data byla vynahrazována výpočty a následným

barevným formátováním buněk. Výroba takovéto tabulky a zapracování jmen žáků zabrala příslušnému profesorovi několik hodin neplacené práce navíc. K tomu připočtíme fakt, že ani konečný výstup nebyl ani zdaleka ideální. Výsledná tabulka sice byla schopná vyjádřit, který student si zvolil jaký seminář a jaké semináře by měly projít prvním kolem, ale obsahovala také spoustu zbytečných dat navíc a dále s nimi nepracovala. Nebyla kupříkladu schopna porovnat, zda mají dva semináře společného studenta, a podobně.

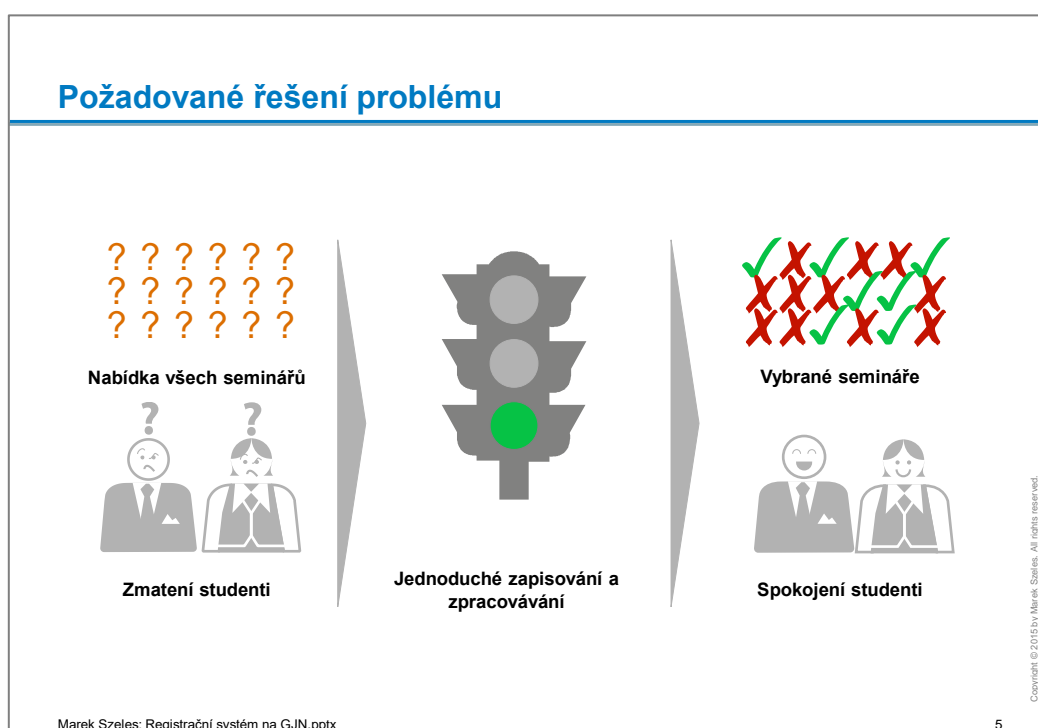
Z těchto, a i z dalších důvodů je zřejmé, že tento systém bylo nutné co nejrychleji nahradit něčím efektivnějším.

2. Zvolené řešení pro nadcházející školní rok 2016/2017

Po důkladné analýze stávající metody registrace bylo nutné si stanovit základní východiska pro tvorbu nového registračního systému, tedy:

- Jaké jsou konkrétní slabé a silné stránky stávajícího systému – Co je dobré zachovat a co nahradit?
- Jak usnadnit a zefektivnit zapisování seminářů pro studenty?
- Jak co nejvíce zjednodušit analýzu "úspěšných" seminářů pro zpracovávajícího?
- Jaké jsou technické možnosti při vytváření takového systému?

Na první skupinu otázek již bylo zodpovězeno v předchozí kapitole, další otázky však musely být

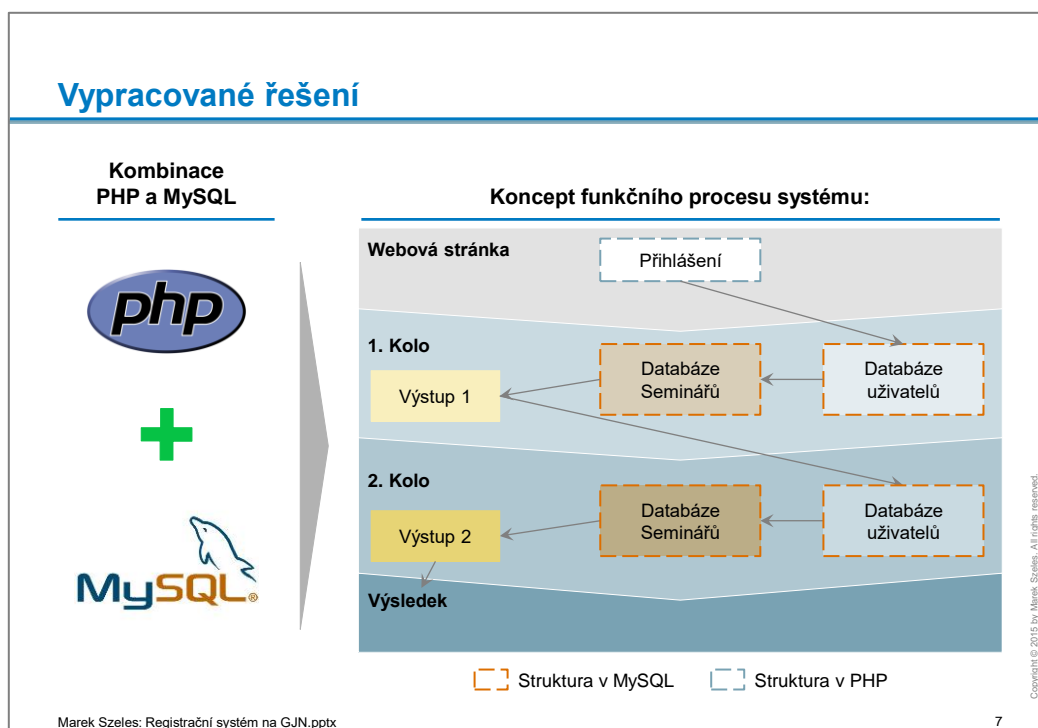


Obrázek 2: Vizuální reprezentace ideálního nového řešení problému

pečlivě zváženy. Pro začátek bylo vhodné zvážit, zda nevyužít znalosti programovacího jazyku Java, se kterým jsem již měl zkušenosti z předchozích projektů. To by však také znamenalo vývoj systému ve formě tzv. Appletu na internetové stránce, což je forma vhodná zejména na složitější programy na webové stránky, které podporují i určitou formu například základního fyzikálního modelování, či zobrazování trojrozměrných objektů. Cenou za to je vcelku robustní velikost zkompilevaného kódu, pomalost načítání a naprostá nekompatibilita s některými prohlížeči, zejména těmi na mobilních zařízeních. Vzhledem k tomu, že zmíněné prvky registrační systém nepotřebuje, bylo toto řešení brzy vyloučeno a pozornost se přesměrovala jinam.

Z důkladných rešerší na internetu zaměřených na to, jak obdobné systémy fungují, vyšlo najevo, že nejvhodnějším řešením bude nejspíše vytvoření veřejně přístupné webové stránky se systémem vytvořeným spojením skriptovacího jazyka PHP, který by zajišťoval výpočetní funkčnost a databázovým systémem MySQL, který by zajišťoval tvorbu a změny databází z proměnlivými daty – studenty, semináři, apod.

Jakmile byla zvolena použitá technologie, vytvořil jsem následně také koncepční plán struktury systému, aby odpovídal zadání. Toto koncepční schéma je vidět na následujícím obrázku:



Obrázek 3: Koncepční struktura systému

2.1. Představení použitých nástrojů

V následující kapitole jsou obecně představeny jednotlivé softwarové nástroje využité při vývoji systému pro registraci studentů. Popisy jsou stručné, zejména pro zlepšení orientace v terminologii v následující, konkrétní části popisu vzniku systému.

2.1.1. Html

Základem textového obsahu webu je značkovací jazyk zvaný HyperText Markup Language, zkráceně Html². Tento jazyk se běžně používá k vytváření „koster“ (wireframes) webových stránek a využívá ho v nějaké formě naprostá většina internetu.

2.1.2. CSS

CSS je kolekce metod pro grafickou úpravu webových stránek. Zkratka znamená Cascading Style Sheets, česky "kaskádové styly". Kaskádové, protože se na sebe mohou vrstvit definice stylu, ale platí jen ta poslední.³ CSS se nejčastěji používá ke grafické úpravě html stránek, například úpravě barvy fontu, pozadí, velikosti písma, a podobně.

2.1.3. PHP

PHP je serverový skriptovací jazyk určený nejen pro webové skriptování. Při použití PHP pro dynamické www stránky jsou skripty spouštěny na serveru a k uživateli je přenášen až výsledek jejich činnosti – proto je PHP velice účinný a efektivní při skriptování na webu. Interpret PHP skriptu je také možné volat pomocí příkazového řádku, nebo dotazovacích metod HTTP.⁴

2.1.4. MySQL

MySQL je světově nejrozšířenější Open-Source⁵ databázový systém. Nabízí soustavu užitečných příkazů, pomocí kterých se dají velice efektivně zpracovávat velká množství dat ve formě tabulek, čemuž se souhrnně říká databáze.⁶ Mezi jednotlivými tabulkami mohou být pro zjednodušení, či kvůli bezpečnosti takzvané relace, neboli česky vztahy.⁷

² MAHMOUD, Qusay H., Wlodek DOBOSIEWICZ a David SWAYNE. Redesigning introductory computer programming with HTML, JavaScript, and Java. *ACM SIGCSE Bulletin* [online]. 2004, vol. 36, issue 1, s. 120-1 [cit. 2015-03-04]. DOI: 10.1145/1028174.971344.

³ MEYER, Eric A. *Smashing CSS: professional techniques for modern layout*. Chichester: Wiley, 2011, xiii, 285s. Smashing magazine. ISBN 978-0-470-68416-0.

⁴ ULLMAN, Larry E. *PHP 6 and MySQL 5 for dynamic Web sites*. Berkeley, Calif.: Peachpit Press, 2008, xix, 620 s. ISBN 0-321-52599-x.

⁵ Open-Source = s volně šiřitelnou licencí, a tedy i kódem

⁶ ATKINSON, Leon. *Core MySQL*. Upper Saddle River, NJ: Prentice Hall, c2002, xxiv, 721 p. ISBN 0130661902.

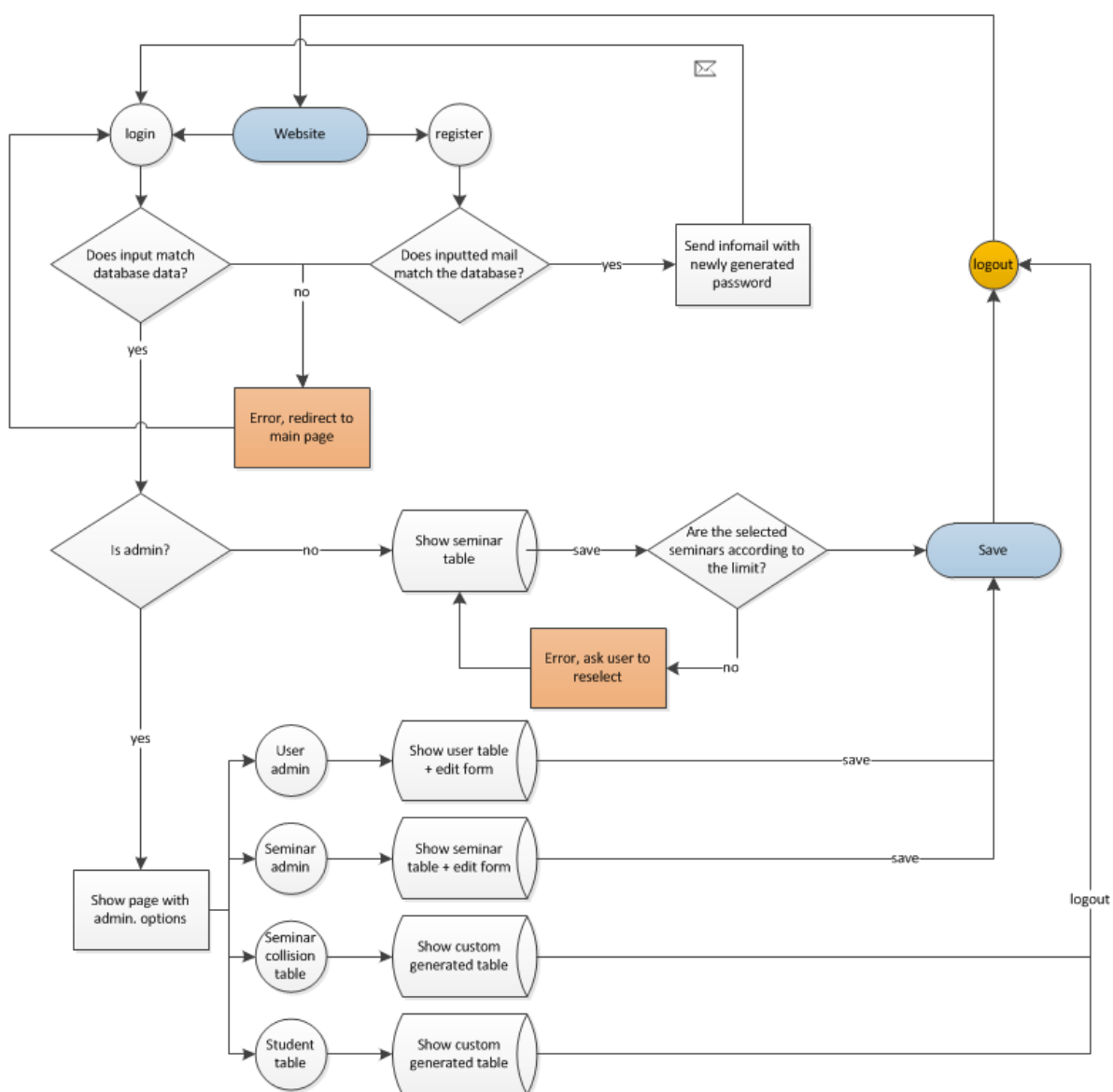
⁷ GUZEL, Burak. SQL for Beginners: Part 3 - Database Relationships. [online]. 2010 [cit. 2015-03-16]. Dostupné z: <http://code.tutsplus.com/articles/sql-for-beginners-part-3-database-relationships--net-8561>

3. Popis systému a proces jeho vzniku

Po vysvětlení všech teoretických konceptů využitých při vytváření programu v následující praktické části zabývám vznikem programu jako takového. Konkrétní zadání projektu v rámci předmětu ZWA je obsaženo v příloze č. 1.

3.1. Funkčnost systému

Po nastudování potřebných funkcí a technologií bylo nutné připravit strukturu funkčnosti webu a databáze, aby byl co nejjednodušší a zároveň uživatelsky příjemný. Vzniklá struktura je vizuálně znázorněna na následujícím zjednodušeném schématu:



Obrázek 4: Schéma architektury databáze

Jak je na schématu vidět, web je koncipovaný tak, že na webu je přihlašovací stránka, na které je možné se přihlásit, anebo registrovat. Aby nebylo možné vytvářet spoustu nežádáných uživatelů a jelikož víme, které uživatele chceme, aby se registrovali - tedy studenty GJN, je registrace podmíněna tím, že zadaný mail je již zaveden do databáze (na počátku je tedy nutné, aby existoval alespoň jeden admin, který se přidá do databáze mimo systém). Pokud je uživatel zaveden v databázi, po zadání své e-mailové adresy se registraci mu zobrazí náhodně vygenerované heslo a zároveň je mu toto heslo zasláno na zadanou e-mailovou adresu. Poté se může pomocí tohoto hesla přihlásit.

Pokud souhlasí kombinace e-mailu a hesla, uživatel je přihlášen a zobrazená nabídka závisí na tom, zda-li je uživatel administrátor, nebo pouhý řadový uživatel, tedy student.

3.1.1. Řadový uživatel - student

Řadoví uživatelé nemají po přihlášení tolik možností. Po úspěšném přihlášení se jím zobrazí seznam seminářů zobrazující vždy název, vyučujícího a popis semináře, podle toho, jaké jsou pro uživatele k dispozici (nabídka seminářů se liší pro jednotlivé ročníky).

Poté, co si řadový uživatel zvolí semináře, o které má zájem a klepne na tlačítko „odeslat“, systém rozhodne, zdali je jeho výběr platný – tedy jestli nemá vybráno seminářů více, než je pro jeho ročník povolený limit. V tom případě systém uživatele upozorní a navrátí ho na předchozí stránku.

Pokud výběr seminářů proběhne v pořádku, systém data uloží do databáze a uživatel se může odhlásit. Uživatelská příručka se nachází v příloze č. 3.

3.1.2. Administrátor – učitel zpracovávající semináře.

Administrátor má k dispozici mnohem více možností, a tak se mu po přihlášení na webu zobrazí celé menu. V něm si má možnost vybrat mezi administrací uživatelů, administrací seminářů, tabulkou kolizí seminářů a tabulkou kolizí studentů.

V administraci studentů je možné vytvářet a mazat studenty pomocí jednoduchého formuláře, který edituje záznamy v databázi. Pod formulářem se vyskytuje tabulka, ve které se zobrazují všichni uživatelé a jejich vlastnosti (jméno, mailová adresa, ročník, počet vybraných seminářů). Je také možné uživatele mazat. Po klepnutí na jednotlivé uživatele se zobrazí nová stránka, kde jsou vypsány semináře, které si uživatel zapsal.

V administraci seminářů je možné vytvářet semináře pomocí formuláře, kde se zadává název, vyučující, možnost výběru pro jednotlivé ročníky a popis semináře. Ten lze při zadávání i formátovat pomocí html. Pod formulářem je tabulka, kde jsou vypsány všechna zadané

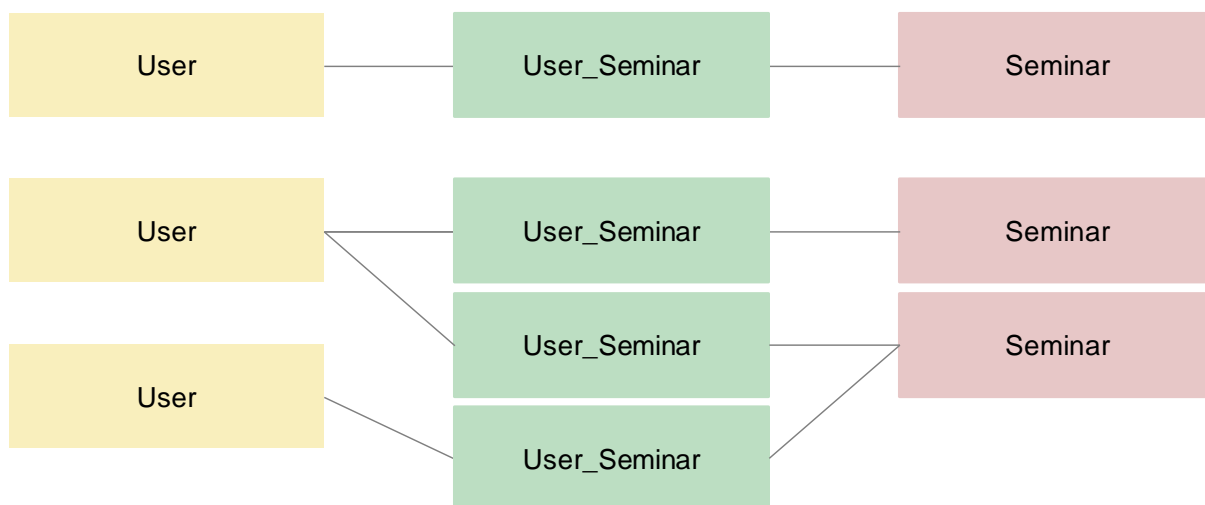
V tabulce kolizí seminářů a studentů se zobrazují dvě obdobné tabulky – v první je možné vidět které semináře mají kolik společných studentů a v druhé, kteří studenti mají společné semináře. Uživatelská příručka pro administrátory se nachází v příloze č. 4.

Většinová část systému je výpočetního rázu a tak i většina kódu byla napsána v jazyku PHP. K naprogramování celého systému jsem používal freeware program Notepad++⁸, se kterým mám již letité zkušenosti. Celý program obsahuje všehovšudy okolo tisíce řádků kódu, a tak není příliš složitý.



12

Vzhledem k tomu, že je semináře a uživatele nutné propojit relací mnoha ku mnoha, existují v databázi tři tabulky: „user“, „seminar“ a „user_seminar“. Poslední zmíněná je pomocná, pomocí které se pak sčítají zapsání na semináře. Schematicky by to mohlo být znázorněno následovně:



Obrázek 6: Relace použité v databázi

3.2.1. Bezpečnostní prvky

Další pálivou otázkou systému byla samozřejmě bezpečnost a spolehlivost. Ačkoliv je velice nepravděpodobné, že by se do systému někdo pokoušel dostat⁹, rozhodl jsem se nenechat nic náhodě a alespoň nejzákladnějším možným chybám a díram v kódu předejít.

Prvním takovým opatřením je samotný fakt, že je při přihlášení považováno nejen heslo, jak bylo původně plánováno, ale také e-mail. Vzhledem ke způsobu, jak je heslo generováno, tak to zajistí 100% spolehlivost. Heslo je totiž generováno hashovací¹⁰ funkcí¹¹, která používá pseudonáhodné číslo vygenerované počítačem. Je zde ale určitá pravděpodobnost, byť extrémně malá, že by systém vygeneroval dvěma uživatelům stejný hash. V tom případě by se systém snažil přihlásit dva uživatele naráz a to by buď ovlivnilo data, nebo by systém celý zkolaboval.

⁹ Toto usuzuji vzhledem k malému okruhu lidí, kteří systém znají a také vzhledem k tomu, že je systém navrhnut tak, že v systému není uložený jiný osobní údaj nežli e-mail uživatele, což není pro hackery lákavá kořist

¹⁰ Způsob šifrování, ze kterého ale nelze získat původní data, narozdíl od šifer. Více informací zde: http://cs.wikipedia.org/wiki/Ha%C5%A1ovac%C3%AD_funkce

¹¹ Použit byl hash Adler32 kvůli estetickým výstupům hesel, info o hashi zde: <http://en.wikipedia.org/wiki/Adler-32>

Další je opatření proti velmi časté metodě hackerů, které se říká SQL injekce¹². Jde ve své podstatě o to, že hacker místo e-mailu a hesla napíše útržek SQL kódu, kde konkrétně například napíše **"OR 1=1"** – uvozovkami tím uzavře uvozovky a pak nabídne systému funkci nebo a za tím tvrzení, že 1=1, což samozřejmě platí vždy. Výsledkem by pak bylo, že by systém vypsál všechny e-mailové adresy i s jejich hesly, což je samozřejmě nepřípustné. Bylo tedy nutné tomuto pomocí kódu zabránit.

Zabránit se takovémuo vniknutí dá několika způsoby, já jsem použil asi nejsnazší cestu, kdy se po zadání údajů aplikuje také formát dat, v našem případě tedy *string*. Potom už kompilér nebude data interpretovat jako SQL kód, ale právě jako *string*, neboli běžný text a potom je samozřejmé, že žádný uživatel s e-mailem a heslem **"OR 1=1"** neexistuje. Hackera to tedy po neúspěšném přihlášení vrátí na hlavní stránku. Řešení je zvýrazněno na následujícím kódu přihlašovací stránky:

```
1 <?php
2 require_once 'scripts/db-connect.php';
3 session_start();
4 echo '<link rel="stylesheet" href="style.css" type="text/css" />';
5 <div class="line"></div>;
6 include "header.php"?>
7 <div class="qhsah-container">
8 <div class="qhsah">
9 <?php
10 $mail = $_POST["mail"];
11 $code = $_POST["code"];
12 $stmt = $dbI->prepare("SELECT userID FROM `".$prefix."Users` WHERE mail = ? AND password = ?");
13 $stmt->bind_param("ss", $mail, $code);
14 $stmt->bind_result($userID);
15 $stmt->execute();
16 $stmt->fetch();
17 $stmt->store_result();
18 if ($userID!=null) {
19     echo "<div style='text-align: center;'> <h2>Přihlášení</h2>Přihlášení proběhlo úspěšně, nyní budete přesměrováni<br> <a
20     $_SESSION["userID"]=$userID;
21 }else{
22     echo "Emailová adresa či kód nesouhlasí.";
23     echo $userID;
24 }
25 ?>
26 </div>
27 </div>
28 <?php
29 echo "<meta http-equiv='refresh' content='2;url=\"/" . " . ">";
30 include ("paticka.php");
31 ?>
```

Obrázek 7: Kód přihlašovací stránky, zvýrazněna ochrana před SQL injekcí

¹² <http://www.veracode.com/security/sql-injection>

3.2.2. Testování

Aplikace byla testována z několika hledisek. Nejprve prošla bezpečnostním testováním rizikových prvků popsaných v předchozí kapitole.

Posléze byla otestována z grafického hlediska, aby vypadala dobře na různých zařízeních, jelikož rozměry obrazovek se liší a je nutné, aby stránka vypadala dobře jak na počítači, tak na mobilu. Testování proběhlo na 25ti různých zařízeních s různými operačními systémy a webovými prohlížeči. Výsledek testů (screenshoty zařízení) jsou zahrnuty v příloze č. 2.

Poslední fáze testování se zaměřila na obsah – do databáze bylo vyplněno několik seminářů a zhruba deset studentů bylo požádáno o otestování systému. Tímto byly odlazeny poslední nedostatky.

4. Využití systému v praxi

V následující části je popsáno a názorně zobrazeno praktické využití systému, tedy jak je možné systém používat z pohledu jak administrátora (učitele zpracovávajícího registraci), tak řadového uživatele (studenta vybírajícího si seminář).

4.1. Stav využívání

Po úspěšném otestování systému, včetně otestování zadávání seminářů, uživatelů i zkoušky samotnými studenty, se s „ostrým“ plným využitím nově vzniklého systému počítá od nadcházejícího školního roku.

5. Závěr

V rámci práce byl podrobně popsán problém s výběrem seminářů na Gymnáziu Jana Nerudy, nastíněna možná řešení a následně úspěšně vytvořen kvalitní registrační systém na bázi kombinace PHP a MySQL, který tento problém bude řešit v budoucích letech a i letos má potenciál výrazně pomoci při zpracovávání vybraných seminářů. Vznik tohoto systému je v práci zdokumentován, a to jak po teoretické, tak i po praktické stránce.

V teoretické části jsou popsány použité postupy a technologie nutné pro tvorbu systému. Je zde popsáno které technologie k čemu slouží a na příkladech je ukázáno jejich možné použití.

Následně je v části praktické popsán vývoj systému, jeho struktura a funkční rozhraní. V další části následuje také ukázka jeho využití a celkové zhodnocení ze strany školy, které počítá

s dalším využitím systému do budoucna, jelikož se systém jeví jako mnohem efektivnější řešení výběru seminářů, než jaké je na škole využíváno v současnosti.

6. Přílohy

6.1. Příloha č.1 – Zadání práce

Původní zadání práce v rámci předmětu ZWA

6.2. Příloha č.2 – Vizualizace webu na různých přístrojích a prohlížečích

6.3. Příloha č.3 – Uživatelská příručka pro studenty

6.4. Příloha č.4 – Uživatelská příručka pro administrátory (učitele)

6.5. Příloha č.5 – Kód programu

Vzhledem k povaze systému (několik propojených souborů) a objemu několika set řádků kódu je z úsporných důvodů celý kód systému dostupný jako příloha pouze v digitální podobě, v jednom komprimovaném souboru (.rar).