



# Maturitní práce

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

**Autor: Marek Szeles**

**Škola: Gymnázium Jana Nerudy, Praha 1**

**Vedoucí práce: Mgr. Zdeněk Plášek**

## Prohlášení

*Prohlašuji, že jsem svou maturitní 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 ..... dne .....

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 2015/2016 za pomoci webové stránky a soustavy databází.

### *Klíčová slova:*

Semináře, Registrační systém, Analytika, Web, 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 2015/2016. The system is created in the form of a webpage, using a system of databases to store data.

### *Keywords:*

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

## Obsah

1. Úvod.....	5
1.1. Řešení ve školním roce 2014/2015 .....	7
2. Zvolené řešení pro nadcházející školní rok 2015/2016 .....	9
2.1. Představení použitých nástrojů .....	11
2.1.1. Html .....	11
2.1.2. CSS .....	11
2.1.3. PHP .....	11
2.1.4. MySQL.....	11
3. Popis systému a proces jeho vzniku .....	16
3.1. Postup vzniku systému.....	16
3.2. Funkčnost systému.....	17
3.2.1. Řadový uživatel - student.....	18
3.2.2. Administrátor – učitel zpracovávající semináře.....	18
3.3. Programování .....	19
4. Využití systému v praxi .....	23
4.1. Z pohledu uživatele - studenta .....	24
4.2. Z pohledu administrátora .....	25
4.2.1. Tabulka společných uživatelů .....	25
4.2.2. Správa uživatelů .....	26
4.2.3. Správa seminářů.....	27
4.3. Stav využívání .....	27
5. Závěr.....	28
6. Přílohy.....	28
6.1. Příloha č.1 – Kód systému.....	28

## 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 ŠVP<sup>1</sup>, 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 velká, 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.

---

<sup>1</sup> ŠVP – Školní vzdělávací program [online] Dostupné z: [http://cs.wikipedia.org/wiki/%C5%A0koln%C3%AD\\_vzd%C4%9Bl%C3%A1vac%C3%AD\\_program](http://cs.wikipedia.org/wiki/%C5%A0koln%C3%AD_vzd%C4%9Bl%C3%A1vac%C3%AD_program)

## Současný problém registrace seminářů na GJN

### Popis problému

Po zavedení seminářového<sup>1</sup> systému pro poslední dva ročníky<sup>2</sup> na Gymnáziu Jana Nerudy vznikl problém - Nebylo jak určit, o které semináře je dostatečný zájem na to, aby byly otevřeny. Řešení použité pro minulý rok bylo značně provizorní a nebylo uživatelsky příjemné ani pro studenty, ani pro zpracovávající profesory. Je tedy nutné vytvořit lepší systém registrací studentů na semináře.

### Klíčové otázky

#### Technické

- Jaké je nejlepší řešení zpracování registrací?
- Jaké jsou možnosti při kódování?

#### Obsahové

- Jak usnadnit a zefektivnit zapisování seminářů pro studenty?
- Jak co nejvíce zjednodušit analýzu "úspěšných" seminářů pro zpracovávajícího?

1. Systém povinně volitelných seminářů, kdy si studenti zcela volí v části rozvrhu, jaké předměty se chtějí učit. Zbytek rozvrhu určuje Rámcový vzdělávací plán. 2. Kvinta a sexta Merchant of Venice.pptx

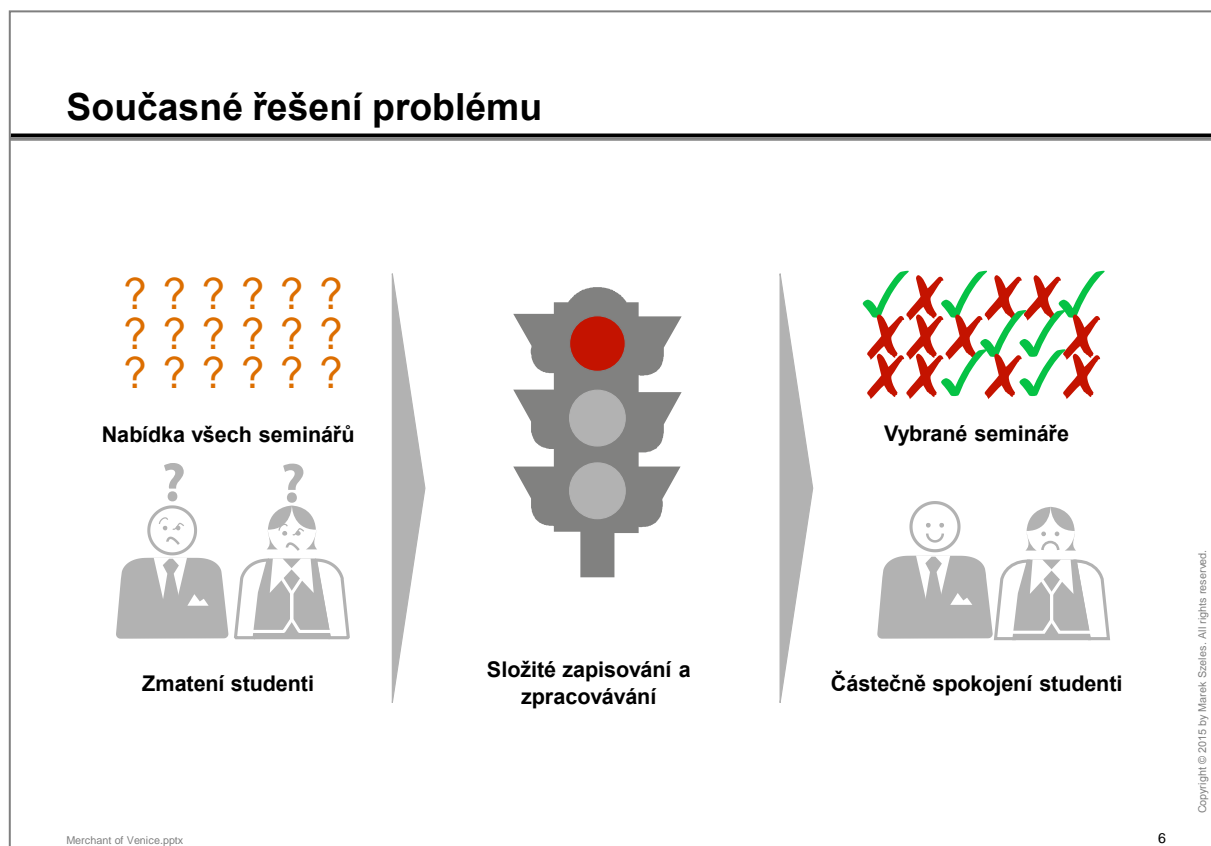
### *Obrázek 1: Popis problému a klíčové otázky*

Na grafickém rozložení výše je vidět přehledné popis problému a vybraný klíčové otázky pro tvorbu efektivního řešení.



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.



*Obrázek 3: Vizuální reprezentace současného řešení problému*

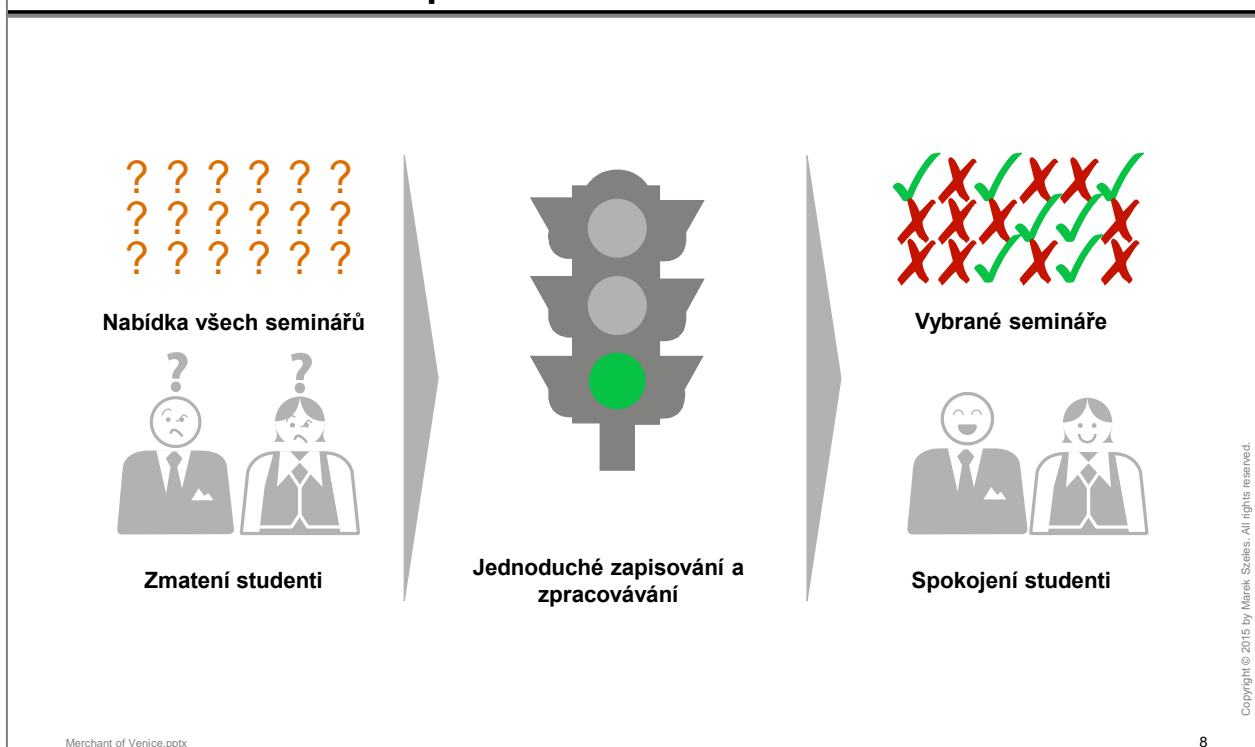


## 2. Zvolené řešení pro nadcházející školní rok 2015/2016

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?

### Požadované řešení problému

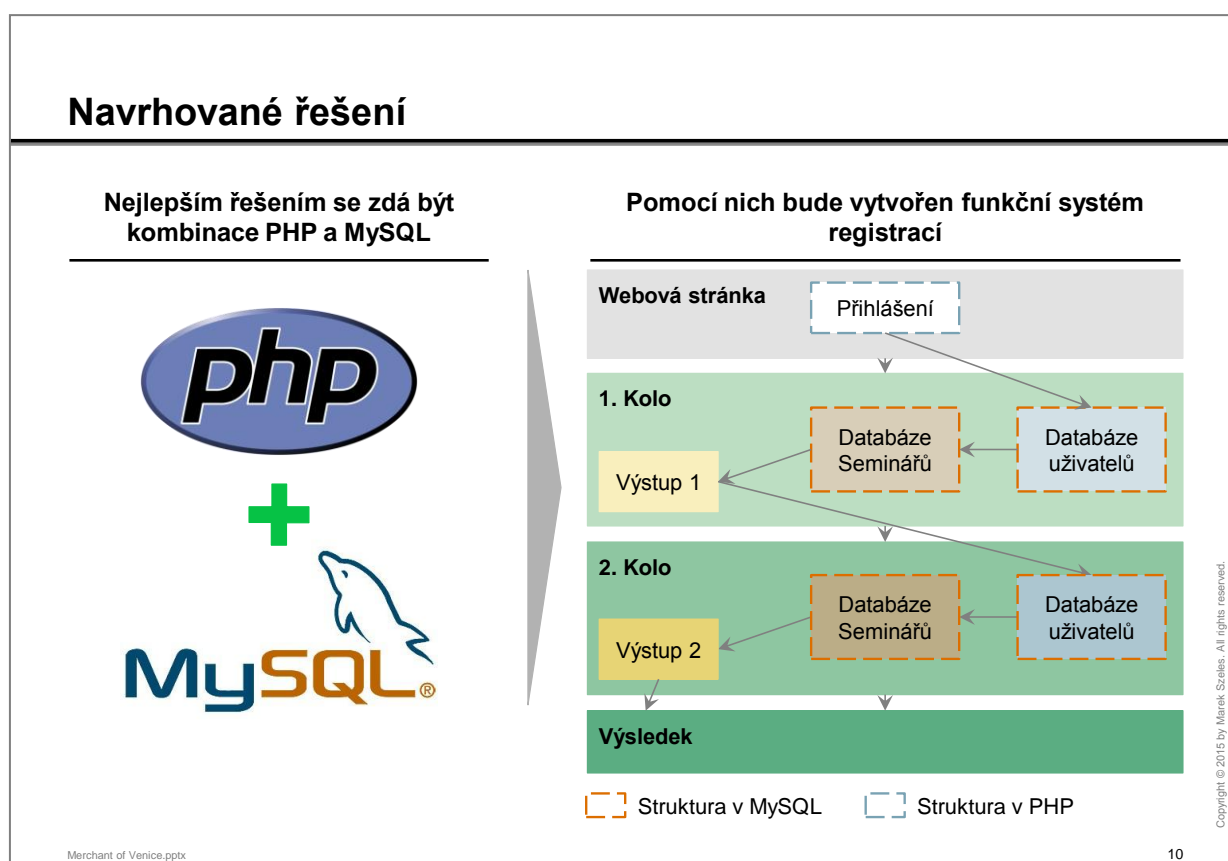


*Obrázek 4: Vizuální reprezentace ideálního nového řešení problému*

Na první set otázek již bylo zodpovězeno v předchozí kapitole, další otázky však musely být 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 zkompilovaného 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 5: Koncepční struktura systému*

## 2.1. Představení použitých nástrojů

V následující kapitole jsou obecně představeny jednotlivé softwarové nástroje využívané 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 Html<sup>2</sup>. 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.<sup>3</sup> 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. Přesto je podle indexování z roku 2013 zakomponován na zhruba 39% indexovaných webových stránkách. 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.<sup>4</sup>

### 2.1.4. MySQL

MySQL je světově nejrozšířenější Open-Source<sup>5</sup> databázový systém. Nabízí soustavu užitečných příkazů, pomocí kterých se dají velice efektivně zpracovávat velká množství dat ve formě tabulek, čemuž se souhrnně říká databáze.<sup>6</sup> Mezi

---

<sup>2</sup> 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.

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

<sup>4</sup> 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.

<sup>5</sup> Open-Source = s volně šiřitelnou licencí, a tedy i kódem

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

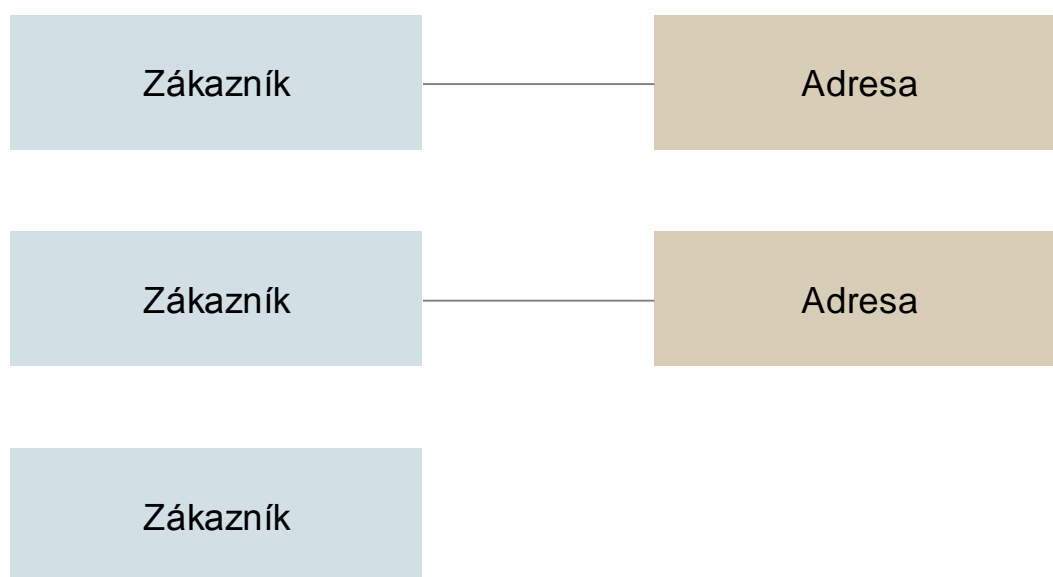
jednotlivými tabulkami mohou být pro zjednodušení, či kvůli bezpečnosti takzvané relace, neboli česky vztahy.<sup>7</sup>

Relace (database relationships) se rozdělují na několik druhů:

- **Jeden ku jednomu**

Relace jeden ku jednomu vyjadřuje jednoduchý vztah, kdy je k jednomu záznamu v jedné tabulce přiřazen právě jeden záznam v druhé tabulce.

Například pokud máme databázi v internetovém obchodu, můžeme mít jednu tabulku obsahující jména zákazníků a druhou tabulku obsahující adresy. Adresy bychom pak k uživatelům jednoduše připojili jako na následujícím schématu:



*Obrázek 6: Ilustrační schéma – jeden ku jednomu*

Je důležité si povšimnout, že není nutné, aby mezi záznamy nějaká relace byla, mohou tedy existovat záznamy zákazníků bez přiřazených adres, a naopak.

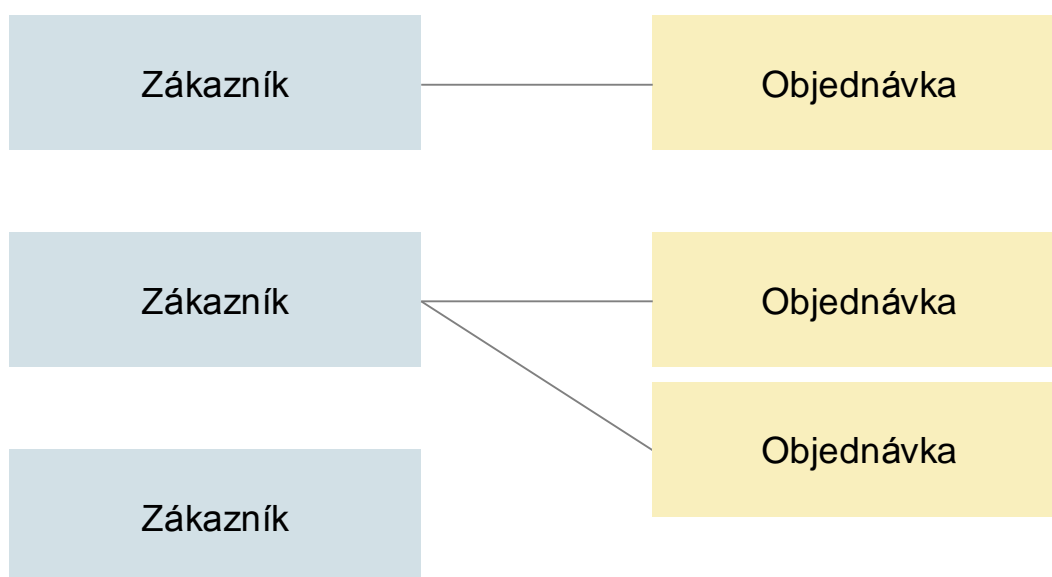
---

<sup>7</sup> 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>

- **Jeden ku mnoha/mnoho ku jednomu**

Jedná se o nejčastěji používanou formu relace. V rámci našeho příkladu by se jednalo například o to, že zákazník může mít k sobě přiřazeno více objednávek. Mezi další příklady mohou patřit případy, kdy objednávky obsahují více, než jen jeden kus zboží, nebo když má zboží popisky uložené v několika jazycích.

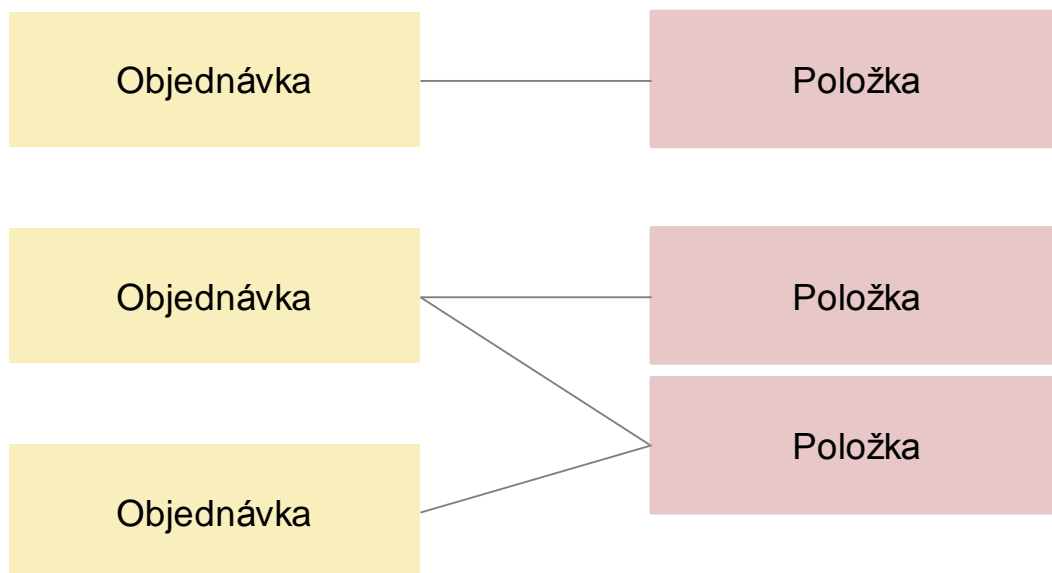
Vizualizace této relace je vidět na dalším schématu, na příkladu vztahu zákazník – objednávka, tedy jeden zákazník může mít více objednávek najednou, ale jedna objednávka koresponduje vždy jen k jednomu ze zákazníků.



*Obrázek 7: Ilustrační schéma – jeden ku mnoha*

- **Mnoho ku mnoha**

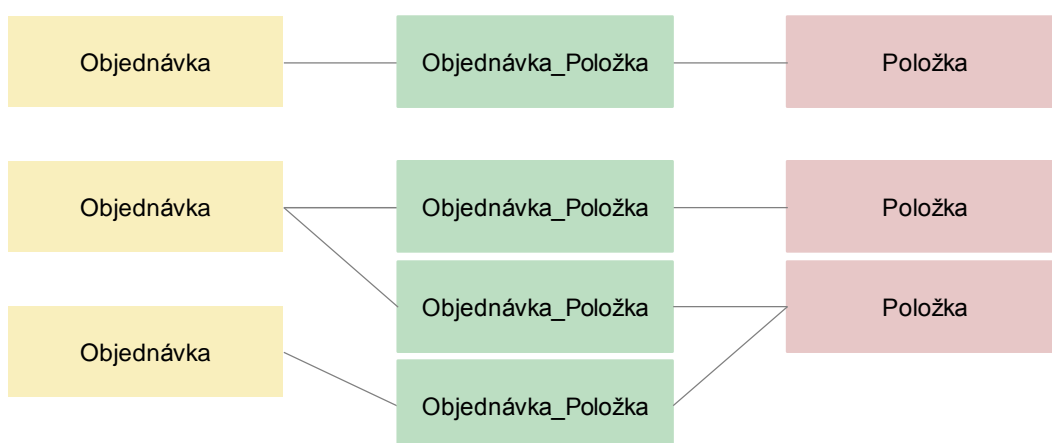
V některých případech je nutné mít možnost na obou koncích relace použít více položek tabulky. Kupříkladu objednávka může obsahovat více položek a zároveň každá položka může korespondovat k více objednávkám, viz schéma:



*Obrázek 8: Ilustrační schéma – mnoho ku mnoha*

V tomto případě je nutné vytvořit další, pomocnou výpočetní tabulku, ke které budou mít tabulky objednávek a položek relaci mnoho ku jednomu a jedno ku mnoha. Jediným účelem této tabulky je zprostředkovávat relaci mnoho ku mnoha.

Tabulka se většinou pojmenovává „NázevTabulky1\_NázevTabulky2“. Pokud bychom ji zanesli do schématu výše, vypadalo by celé rozhraní následovně:

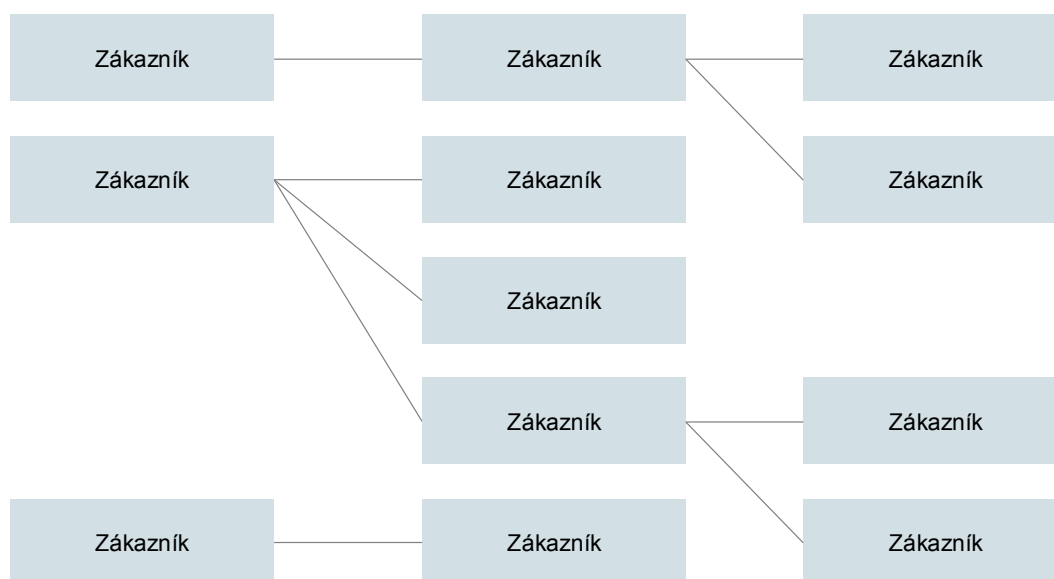


*Obrázek 9: Ilustrační schéma – mnoho ku mnoha s pomocnou tabulkou*

- **Sebeodkazující relace**

Tato ne až tak častá relace se používá, pokud je nutné, aby měla tabulka vztah sama se sebou. Pokud použijeme znovu příklad obchodu, pak je tato relace aplikovatelná zejména v případě, že obchod má referenční program, kdy zákazníci doporučují obchod dalším lidem, kteří se stávají novými zákazníky, a referent za to dostane určitý bonus.

V zásadě je tato relace podobná relaci „jedna ku mnoha“, jelikož jeden zákazník může obchod doporučit více lidem. Jediný rozdíl je nakonec v tom, že všechny položky jsou ze stejné tabulky. Na schématu by situace mohla vypadat například následovně:



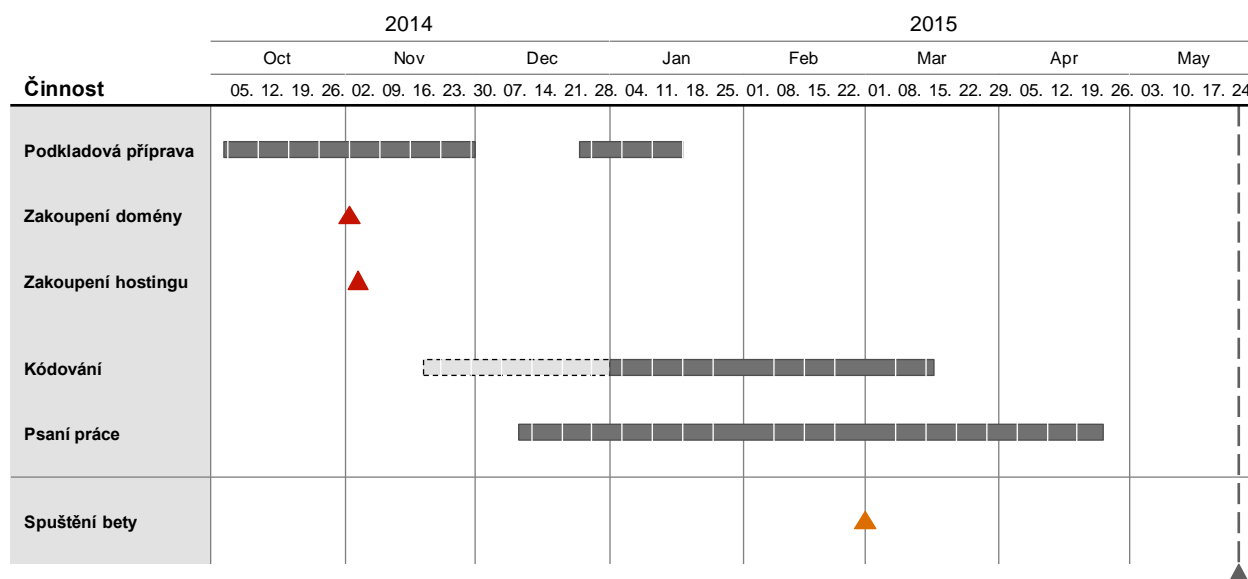
*Obrázek 10: Ilustrační schéma – sebeodkazující relace*

### 3. Popis systému a proces jeho vzniku

Po vysvětlení všech teoretických konceptů využitých při vytváření programu se následující praktické části zabývám vznikem programu jako takového.

#### 3.1. Postup vzniku systému

Jakmile bylo téma práce rozhodnuto, bylo nutné vytvořit si časový plán, podle kterého bude program i práce vznikat. Finální časové rozložení vzniku práce je vidět z následujícího harmonogramu:



Obrázek 11: Harmonogram práce

Nejprve, vzhledem k relativně velikému nedostatku znalostí a zkušeností nutných k vytvoření systému, bylo již od začátku října nutné se ve volném čase vzdělávat o možnostech programování zejména v MySQL a PHP, jelikož právě v těchto programovacích jazycích jsem měl největší znalostní mezeru. Inspirovat jsem se rozhodl ve své minulé tvorbě, konkrétně v dnes již nepoužívaném Informačním systému, na kterém jsem pracoval pro Českou středoškolskou unii. Tato možnost měla krom jiného výhodu i v tom, že bylo možné si všechny své poznatky ihned v praxi testovat a empiricky ověřit.

Nedlouho poté jsem na začátku listopadu zakoupil i doménu a webhosting u společnosti Wedos, se kterou mám již předchozí zkušenosti, na který jsem následně již nahrál základní, zatím nefunkční verzi systému, která se stala kostrou další práce.

Jak je na schématu vidět dále, s intenzivním programováním projektu mělo být započato v polovině listopadu. Tomu se tak bohužel z vážných důvodů naprostého vytížení kvůli pololetním písemkám nestalo a před polovinou prosince jsem pouze začal psát teoretickou část práce. Brzy se také projevila nutnost dalšího dodatečného studia funkcí, které budou potřebné, jelikož se změnila plánovaná struktura systému.

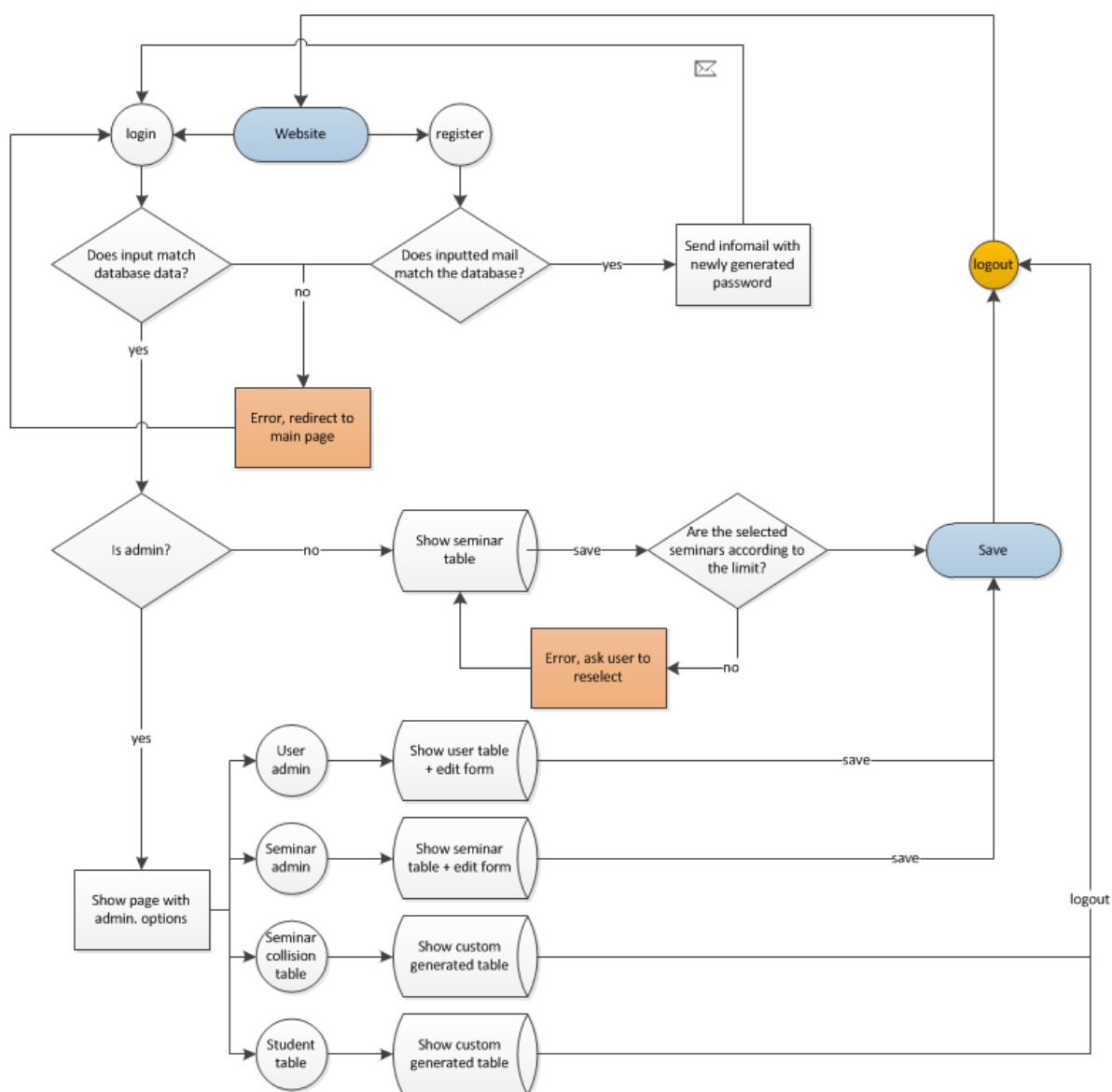


S programováním samotného systému se tedy započalo až začátkem ledna s tím, že koncem února se plánovalo testování funkční betaverze systému. Tento termín byl po intenzivním kódování dodržen a posléze byly v systému opraveny všechny chyby nalezené při testování, plus byly přidány další funkce pro administrátora.

Nakonec bylo již nutné pouze dopsat samotnou práci, což bylo opět ztíženo vlnou zakončovacích písemek, avšak nakonec se mi podařilo práci v této formě dokončit.

### 3.2. 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 12: Schéma architektury databáze

Jak je na schématu vidět, web je koncipovaný tak, že je na něm přihlašovací stránka, na které je možné se přihlásit, anebo registrovat. Aby nebylo možné vytvářet spoustu nežádaných 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 administrátor, který se přidá do databáze mimo systém). Pokud je uživatel zaveden v databázi, po zadání své e-mailové adresy v registraci se 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 pouze řadový uživatel, tedy student.

### **3.2.1. Řadový uživatel - student**

Řadový 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.

### **3.2.2. 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 vypsané semináře, které si uživatel zapsal.

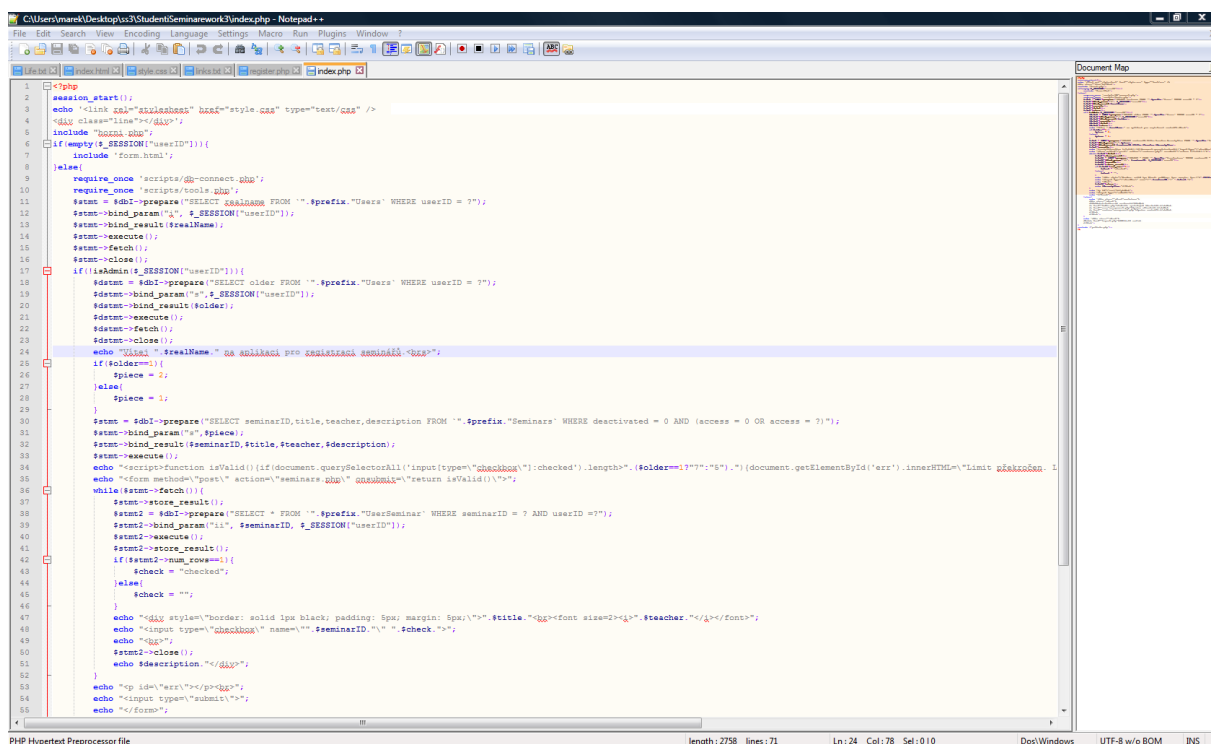
V administraci seminářů je možné vytvářet semináře pomocí formuláře, kde se zadává název, vyučující, možnost výběru pro jednotlivé ročníky a popis semináře. Ten lze při zadávání i formátovat pomocí html. Pod formulářem je tabulka, kde jsou vypsané všechny zadané semináře. Kromě vypsaných vlastností semináře je možné semináře i smazat. Na konci stránky je možné deaktivovat semináře, které mají počet přihlášených nižší, než je zadané číslo. Tím se seminář nesmaže, ale pouze

znevíditelní všem běžným uživatelům. Tímto se simuluje původní záměr mít dvoukolový výběr seminářů a zároveň se umožňuje vzácný případ, kdy se studenti i v druhém kole mohou dohodnout, že chtějí otevřít původně uzavřený seminář z prvního kola – jelikož aktivita semináře lze snadno změnit klepnutím v tabulce.

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.

### 3.3. Programování

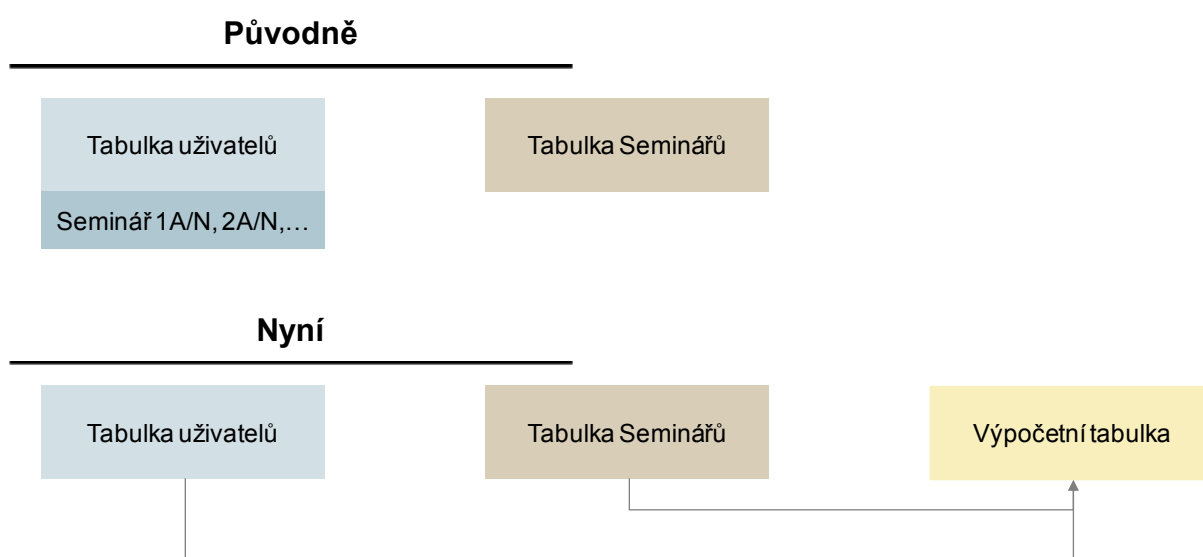
Většinová část systému je výpočetního rázu a tak i většina kódu byla napsána v jazyku PHP. K naprogramování celého systému jsem používal freeware program Notepad++<sup>8</sup>, se kterým mám již letité zkušenosti. Celý program obsahuje všehovšudy okolo tisíce řádků kódu, a tak není příliš složitý.



Obrázek 13: Ukázka z programu Notepad++, kód hlavní stránky

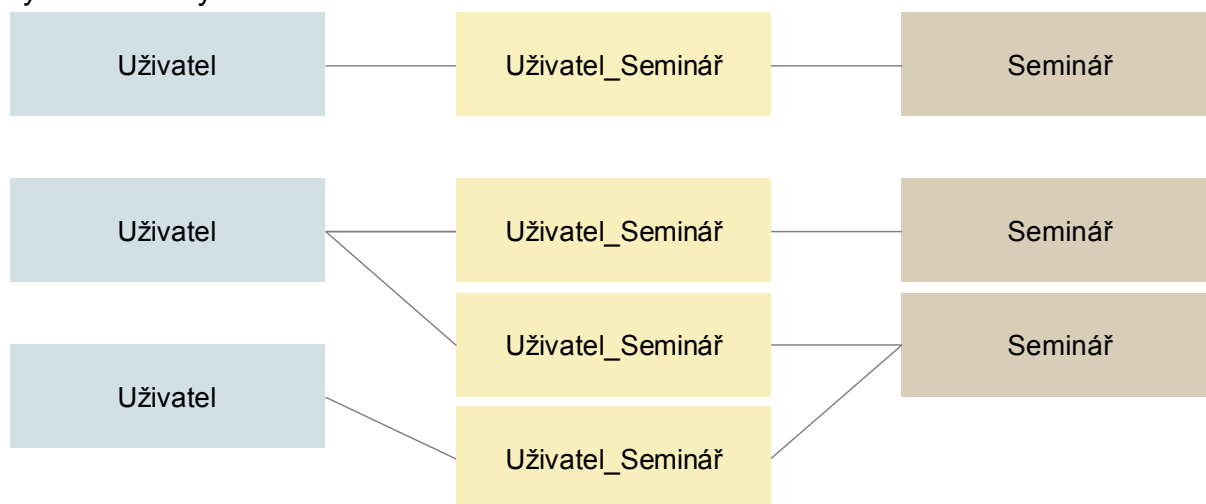
<sup>8</sup> Webové stránky programu: <http://notepad-plus-plus.org/>

Původně jsem zamýšlel vytvořit v databázi dvě tabulky – jednu pro studenty a jednu pro semináře. Jak je vidět na obrázku níže, tabulka studentů byla zamýšlena tak, že by měla pole pro každý seminář ve formě bool, a tedy student by byl na seminář přihlášen podle údaje v tomto poli, které by bylo buď ano, nebo ne. Po konzultaci s jedním mým známým, který se databázemi zabývá však brzy vyšlo najevo, že toto není ideální řešení. Databáze by pak nebyla vůbec flexibilní a například přidávání nových seminářů by bylo velice složité, jelikož by se do tabulky musela přidávat nová pole. Poradil mi tedy využít odlišnou architekturu, kde se využívá relace mnoha ku mnoha a třetí, výpočetní tabulky, která slouží ke zpracovávání záznamů, tedy dvojic student-seminář.



*Obrázek 14: Porovnání původně zamýšlené a použité struktury databáze*

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 15: Relace použité v databázi*

Další pálivou otázkou systému byla samozřejmě bezpečnost a spolehlivost. Ačkoliv je velice nepravděpodobné, že by se do systému někdo pokoušel dostat<sup>9</sup>, rozhodl jsem se nenechat nic náhodě a alespoň nejzákladnějším možným chybám a dírám 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 hashovací<sup>10</sup> funkcí<sup>11</sup>, 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.

Další je opatření proti velmi časté metodě hackerů, které se říká SQL injekce<sup>12</sup>. 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

<sup>9</sup> 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

<sup>10</sup> 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](http://cs.wikipedia.org/wiki/Ha%C5%A1ovac%C3%AD_funkce)

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

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

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 vypsal všechny e-mailové adresy i s jejich hesly, což je samozřejmě nepřijatelné. Bylo tedy nutné tomuto pomocí kódu zabránit.

Zabránit se takovémuto 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:

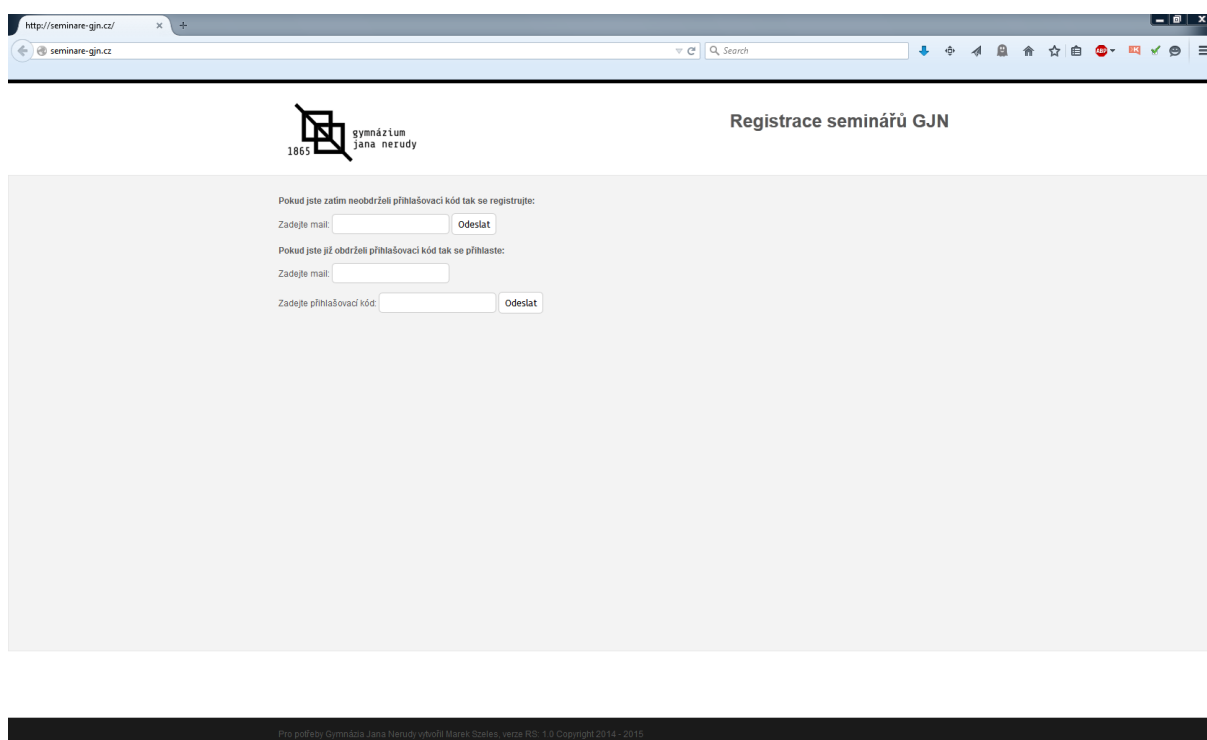
```
1 <?php
2 require_once 'scripts/db-connect.php';
3 session_start();
4 echo "<link rel='stylesheet' href='style.css' type='text/css' />";
5 <div class="line"></div>;
6 include "header.php";?>
7 <div class="container">
8 <div class="login">
9 <?php
10 $mail = $_POST["mail"];
11 $code = $_POST["code"];
12 $stmt = $db->prepare("SELECT userID FROM ".$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> <= break='/'>Pokud ne, klikněte sem</a> </div>";
20 $SESSION["userID"]=$userID;
21 }else{
22 echo "Emailová adresa či kód nesouhlasí-";
23 echo $userID;
24 }
25 ?>
26 </div>
27 </div>
28 <?php
29 echo "<meta http-equiv='refresh' content='2;url='".$url.">";
30 include ("footer.php");
31 ?>
```

Obrázek 16: Kód přihlašovací stránky, žlutě je zvýrazněna ochrana před SQL injekcí

## 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 registrace), tak řadového uživatele (studenta vybírajícího si seminář).

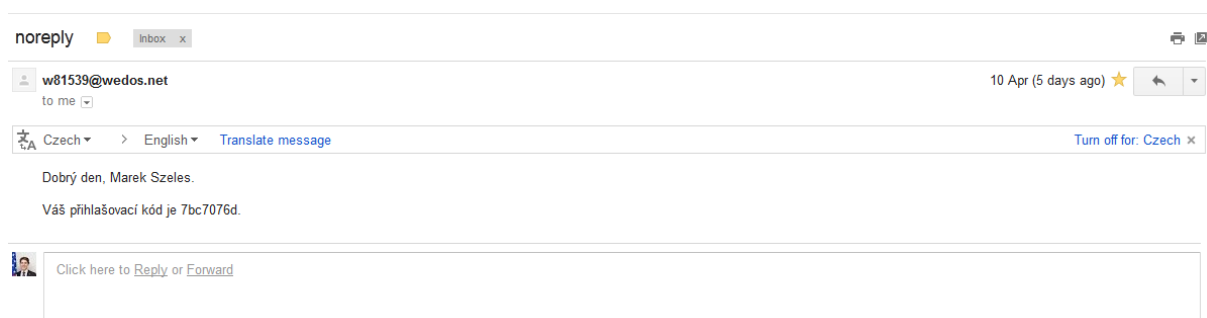
Po příchodu na stránku na webu <http://www.seminare-gjn.cz> se všem zobrazí uvítací stránka s registracemi a přihlášením:



The screenshot shows a web browser window with the address bar displaying <http://seminare-gjn.cz/>. The page header includes the logo of Gymnázium Jana Nerudy (GJN) and the title "Registrace seminářů GJN". The main content area contains two registration forms. The first form is for new users, asking for an email address and a password, with a "Odeslat" (Send) button. The second form is for existing users, asking for an email address and a confirmation code, also with a "Odeslat" button. The footer of the page contains a small copyright notice: "Pro potřeby Gymnázia Jana Nerudy vytvořil Marek Štepl, verze 1.0 Copyright 2014 - 2015".

*Obrázek 17: Úvodní stránka webu*

Pokud je daný mail zanesen do databáze jako uživatel, po jeho zadání se na něj odešle náhodně vygenerované heslo. Odeslaný E-mail vypadá následovně:



The screenshot shows an email interface with the sender "noreply" and the email address "w81539@wedos.net". The email is dated "10 Apr (5 days ago)". The subject line is "Dobrý den, Marek Szeles." and the body text is "Váš přihlašovací kód je 7bc7076d." There is a "Click here to Reply or Forward" link at the bottom.

*Obrázek 18: Email zaslaný registrovaným uživateli*

Následně se může pomocí hesla uživatel přihlásit. Po přihlášení se již rozhraní liší pro studenty a administrátory.

#### 4.1. Z pohledu uživatele - studenta

Po úspěšném přihlášení se studentovi zobrazí možnost výběru seminářů:

http://seminare-gjn.cz/

gymnázium  
jana nerudy  
1863

Registrace seminářů GJN

Vítej, Josef! Dostál na aplikaci pro registraci seminářů.

Zahrádkaření  
Pepaz ☐  
Naučte se pěstovat smrk.

Dražba Traitorů  
Ing. "Zetor" ☐  
Tohle je fakt prdel.

Přírodopis  
Samostudium ☐

Submit Query

[Odhlásit se](#)

Pro pořadí Gymnázia Jana Nerudy vytvořil Marek Štepl, verze RS: 1.0 Copyright 2014 - 2015

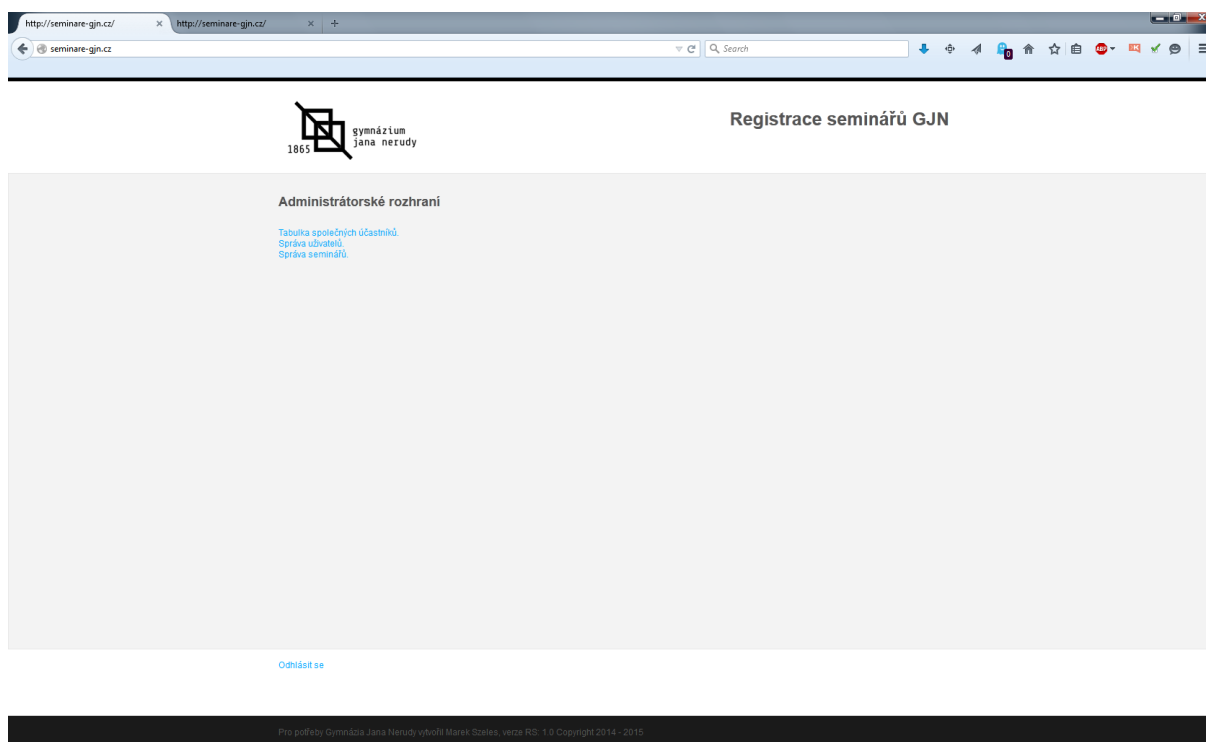
*Obrázek 19: Výběr seminářů pro studenty*

Následně po vybrání seminářů uživatel klepne na tlačítko „odeslat“, vše se uloží a on se může odhlásit. V opačném případě systém uživatele upozorní a on musí výběr zadat znovu.



## 4.2. Z pohledu administrátora

Po přihlášení se administrátorovi zobrazí menu s dalšími možnostmi:



Obrázek 20: Menu administrátora

### 4.2.1. Tabulka společných uživatelů

V této sekci administrátor vidí výpis seminářů seřazený podle abecedy, a pokud mají semináře společného studenta:

	???	Dražba Traktorů	Křivkologie :*	Poslední seminář	Přírodopis	Seminář z anglického jazyka	Zahrádkaření
???	X	0	0	0	0	0	0
Dražba Traktorů	0	X	0	0	1	1	5
Křivkologie :*	0	0	X	0	0	0	0
Poslední seminář	0	0	0	X	0	0	0
Přírodopis	0	1	0	0	X	0	0
Seminář z anglického jazyka	0	1	0	0	0	X	1
Zahrádkaření	0	5	0	0	0	1	X

[Návrat na hlavní stránku](#)

Obrázek 21: Tabulka společných uživatelů seminářů

#### 4.2.2. Správa uživatelů

V této sekci je možné administrovat uživatele, nahoře je formulář k vytvoření nových uživatelů (je možné vytvářet i nové administrátory) a pod tím jsou zobrazováni stávající uživatelé v tabulce s vlastnostmi.

##### Přidat uživatele:

Zadejte jméno:

Zadejte mail:

Ročník:  ▼

Admin: ☐

Vítej na aplikaci pro registraci seminářů.

Jméno	Email	Ročník	Typ	Počet seminářů	
<a href="#">Ondřej Ježil</a>	firma.ondra@gmail.com	-	Admin	2	<a href="#">Smazat</a>
<a href="#">Marek Szeles</a>	marek.szeles@gmail.com	-	Admin	2	<a href="#">Smazat</a>
<a href="#">Jan Pavlát</a>	john.le@seznam.cz	Třet'ák	Uživatel	2	<a href="#">Smazat</a>
<a href="#">Lukáš Trojan</a>	unvoidcz@gmail.com	Třet'ák	Uživatel	0	<a href="#">Smazat</a>
<a href="#">Zdeněk Plášek</a>	plasek@gjn.cz	-	Admin	0	<a href="#">Smazat</a>
<a href="#">Marián Poppr</a>	marianpoppr@centrum.cz	Třet'ák	Uživatel	2	<a href="#">Smazat</a>
<a href="#">Igor Vujovič</a>	vujovic1@seznam.cz	Čtvrt'ák	Uživatel	3	<a href="#">Smazat</a>
<a href="#">Josef Dostál</a>	plavit@seznam.cz	Třet'ák	Uživatel	2	<a href="#">Smazat</a>

[Návrat na hlavní stránku](#)

*Obrázek 22: Administrace uživatelů*

Po klepnutí na jméno uživatele se zobrazí jeho vybrané semináře:

Semináře uživatele Igor Vujovič:

Zahrádkaření

Dražba Traktorů

Seminář z anglického jazyka

[Návrat na hlavní stránku](#)

*Obrázek 23: Vybrané semináře zvoleného uživatele*

### 4.2.3. Správa seminářů

V této sekci je možné administrovat semináře, nahoře je formulář na vytvoření nového, pod ním tabulka zobrazující stávající semináře. Při zadávání popisu semináře je možné text formátovat pomocí html. Na spodku stránky je možné deaktivovat semináře s nižším počtem účastníků, než je zadané číslo. Tímto způsobem je simulován dvoukolový výběr seminářů:

Přidat uživatele:

Zadejte název:

Zadejte vyučujícího:

Ročník:

Zadejte popis:

Vítej na aplikaci pro registraci seminářů.

Název	Vyučující	Popis	Aktivita	Počet přihlášených	Otevřené pro	-
Zahrádkaření	Pepa2	Naučíte se pěstovat smrky.	<a href="#">Aktivní</a>	5	Oba ročníky	<a href="#">Smazat</a>
Dražba Traktorů	Ing. "Zetor"	Naučný seminář	<a href="#">Aktivní</a>	6	Oba ročníky	<a href="#">Smazat</a>
Seminář z anglického jazyka	Smith W., Grunshaw L.	• reálie anglicky mluvících zemí • anglická, americká a australská literatura a film – četba, rozbor, interpretace, diskuse • práce s britským denním tiskem, periodiky • diskuse na aktuální témata, argumentace, vyjadřování a obhajování názoru • debata, public speaking • prezentace seminárních prací a projektů • zdokonalování psaného projevu (eseje, tvůrčí psaní) • role-play, problem-solving a další interaktivní cvičení	<a href="#">Aktivní</a>	1	Čtvrtý ročník	<a href="#">Smazat</a>
Křivkologie :*	Houbová Marta	Geometrie	<a href="#">Neaktivní</a>	0	Oba ročníky	<a href="#">Smazat</a>
Přírodopis	Samostudium	...	<a href="#">Aktivní</a>	1	Čtvrtý ročník	<a href="#">Smazat</a>
???			<a href="#">Neaktivní</a>	0	Oba ročníky	<a href="#">Smazat</a>
Poslední seminář			<a href="#">Neaktivní</a>	0	Oba ročníky	<a href="#">Smazat</a>

Deaktivovat semináře pod limit:

Zadejte limit:

Obrázek 24: Administrace seminářů

### 4.3. Stav využívání

Po úspěšném otestování systému v únoru 2015, 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 – Kód systému

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.