

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## INFORMAČNÍ SYSTÉM ŠKOLICÍHO STŘEDISKA

BAKALÁŘSKÁ PRÁCE

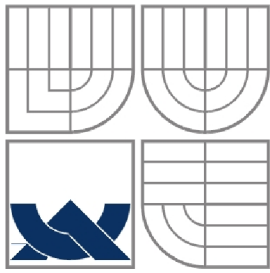
BACHELOR'S THESIS

AUTOR PRÁCE

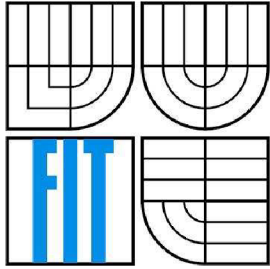
AUTHOR

MARTIN LACKO

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# INFORMAČNÍ SYSTÉM ŠKOLICÍHO STŘEDISKA

LEARNING CENTER INFORMATION SYSTEM

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

MARTIN LACKO

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. ŠÁRKA KVĚTOŇOVÁ, Ph.D.

BRNO 2009

## Zadání bakalářské práce

Řešitel: **Lacko Martin**

Obor: Informační technologie

Téma: **Informační systém školícího střediska**

Kategorie: Databáze

### Pokyny:

1. Seznamte se s dostupnými nástroji a jazyky pro tvorbu webových aplikací, zejména s databází MySQL, HTML, PHP, ASP.NET atp.
2. Proveďte podrobnou analýzu požadavků na IS školícího střediska. Vytvořte vhodné modely tohoto systému.
3. Proveďte detailní návrh webové aplikace.
4. Implementujte tento systém včetně zákaznické části. Na vhodně zvoleném vzorku dat demonstруйте možnosti použití vytvořené aplikace. Diskutujte rovněž možná propojení s externími systémy.
5. Zhodnoťte dosažené výsledky, zaměřte se na možnosti dalšího rozšíření.

### Literatura:

- Williams, H. E., Lane, D.: PHP a MySQL - Vytváříme webové databázové aplikace. Computer Press, 2002, 552 s. ISBN 8072267604
- Kosek, J.: HTML, tvorba dokonalých www stránek. Praha: Grada Publishing, 1998, 291 s. ISBN 80-7169-608-0
- PHP: Hypertext Preprocessor. Dostupné na: <http://www.php.net>
- DeLisle, M.: PHPMyAdmin - efektivní správa MySQL. Brno: Zoner Press, 270 s. ISBN 8086815099
- Ullman, L.: PHP a MySQL. Computer Press, 2004, 536 s. ISBN 8025100634

Při obhajobě semestrální části projektu je požadováno:

- Body 1 - 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Květoňová Šárka, Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2008

Datum odevzdání: 20. května 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií  
Ústav informačních systémů  
612 66 Brno, Božetěchova 2



---

doc. Dr. Ing. Dušan Kolář  
vedoucí ústavu

## **Abstrakt**

Bakalářská práce se zabývá problematikou návrhu a implementace informačního systému školicího střediska. Systém by měl poskytovat jednoduché a přehledné rozhraní pro běžné uživatele a zaměstnance školicího střediska. Systém je postaven na technologiích ASP.NET, C# a Microsoft SQL Server.

## **Abstract**

The bachelor's thesis is concerned with dilemma of suggestion and implementation of learning center information system. The system will allow simple and useful interface for regular users and employees of learning center. The system was implemented by following technologies ASP.NET, C# and Microsoft SQL Server.

## **Klíčová slova**

Školicí středisko, Informační systém, ASP.NET, databáze, Microsoft SQL Server, .NET Framework, C#.

## **Keywords**

Learning center, Information system, ASP.NET, database, Microsoft SQL Server, .NET Framework, C#.

## **Citace**

Lacko Martin: Informační systém školicího střediska, bakalářská práce, Brno, FIT VUT v Brně, 2009

# Informační systém školicího střediska

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Šárky Květoňové, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Martin Lacko  
9.5.2008

## Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Šárce Květoňové, Ph.D. za odborné vedení při vytváření této práce.

© Martin Lacko, 2009.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

Obsah .....	1
1 Úvod.....	2
2 Prehľad použitých technológií .....	3
2.1 .NET Framework .....	3
2.2 Jazyk C# .....	5
2.3 ASP.NET .....	5
2.4 Microsoft SQL Server.....	6
2.5 Visual Studio .....	6
3 Návrh informačného systému .....	7
3.1 Účel a cieľ práce .....	7
3.2 Analýza požiadaviek.....	7
3.3 Typy užívateľov prístupujúcich k systému.....	8
3.4 Modely systému.....	9
3.4.1 Diagramy prípadov užitia .....	9
3.4.2 Popis diagramu prípadu použitia lektorskej časti .....	10
3.4.3 Popis diagramu prípadu užitia zákazníckej časti .....	12
3.4.4 ER diagram .....	13
3.4.5 Popis databázových tabuliek.....	13
3.5 Štruktúra aplikácie .....	16
4 Implementácia.....	18
4.1 Vytvorenie databázovej štruktúry .....	18
4.2 Implementované časti systému .....	19
4.2.1 Vzhľad systému .....	19
4.2.2 Verejne prístupná časť .....	19
4.2.3 Lektorská časť systému .....	23
5 Záver .....	35
Zoznam príloh.....	37

# 1 Úvod

Rýchly rozvoj informačných technológií a informatizácia spoločnosti prináša so sebou množstvo zmien a inovácií, ktoré ovplyvňujú náš každodenný život. Veľké množstvo informácií, ktoré vznikajú je potrebné nejakým spôsobom efektívne spracovávať, ukladať a vyhodnocovať. Informačné systémy v spojení s internetom s ktorými prichádzame v dnešnej dobe do kontaktu takmer každý deň a ich úlohou je, aby boli prospešné celej spoločnosti, či už sú to informačné systémy bánk, štátnych inštitúcií, alebo malých firiem, ktoré šetria najmä čas, finančné prostriedky a ľudské zdroje. Čoraz väčší dôraz a nároky sú kladené na užívateľskú prívetivosť, aby tieto systémy boli jednoducho použiteľné pre každého jednotlivca a z množstva informácií ponúkali užívateľovi len to podstatné a dôležité.

Cieľom je vytvoriť informačný systém pre školiace stredisko poskytujúci počítačové kurzy. Jedná sa o webovú aplikáciu slúžiacu samotným zamestnancom strediska na evidenciu aktuálnych školení, počte prihlásených zákazníkov, sledovanie rozvrhu v učebniach, ale aj pre zákazníkov, ktorí sa môžu samostatne prihlasovať do jednotlivých kurzov, sledovať vypísaný rozvrhu hodín a tým odbremeniť zamestnancov od tejto časovo náročnej agendy.

Táto bakalárska práca sa bude zaoberať popisom mnou zvolených postupov pri tvorbe informačného systému. Prvá kapitola je úvodná a druhá bude venovaná predstaveniu použitých technológií, hlavne predstaveniu platformy ASP.NET, databázového serveru Microsoft SQL Server a vývojovému prostrediu Visual Studio.

Tretia kapitola sa zaoberá návrhom informačného systému od formulovania počiatočných požiadavkov o účele až po cieľa aplikácie. Kapitola popisuje detailnú analýzu požiadaviek a modely navrhnutého systému.

Štvrtá kapitola sa venuje popisu implementácie samotného systému kde popisujem implementované časti a ich funkčnosť. Tento popis je doplnený o praktické ukážky z mnou implementovaného informačného systému.

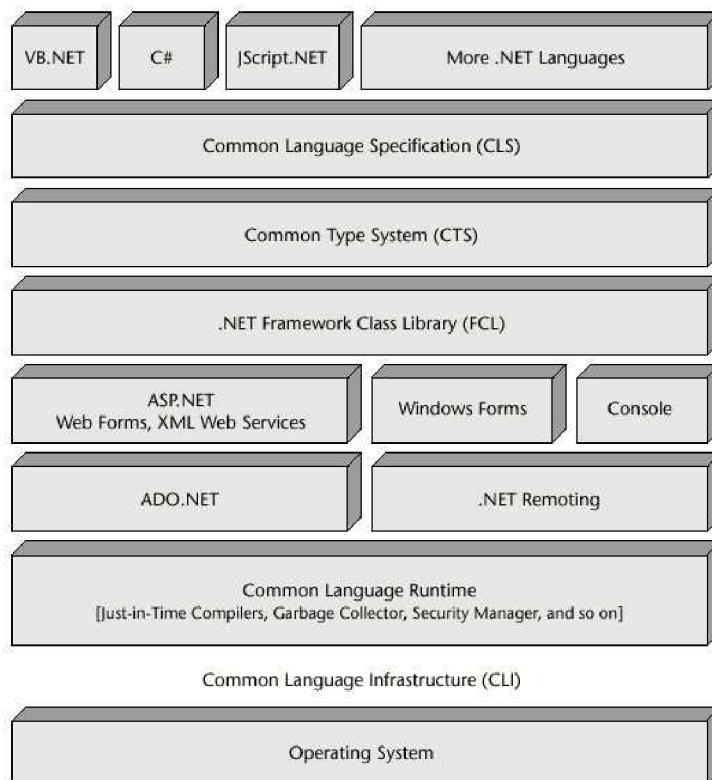
Piata a zároveň záverečná kapitola je zhrnutím celej bakalárskej práce, postupom prác a dosiahnutým cieľom. Zároveň v nej diskutujem o možnostiach ďalších rozšírení a prepojení s externými systémami.

## 2 Prehľad použitých technológií

V tejto kapitole sa budeme venovať technológiám použitým pri vývoji informačného systému a bližšie ich predstavíme a zhrnieme základné vlastnosti. Ako prvú predstavíme technológiu firmy Microsoft .NET Framework, programovací jazyk C#, platformu ASP.NET, SQL Server a vývojové prostredie Visual Studio.

### 2.1 .NET Framework

Základným pilierom ASP.NET je technologická platforma Microsoft .NET Framework, ktorá zapúzdruje sadu objektov používaných na strane serveru i klienta. Používa sa na vývoj aplikácií pre Windows a web.



Obr. 2.1: Štruktúra .NET Framework [2]



Výhody technológie .NET:

- **Objektovo orientované programovanie** - platforma .NET Framework a jazyk C# sú už od začiatku založené na objektovo orientovaných princípoch.
- **Dobrý návrh** – knižnica základných tried je už v základe navrhnutá vysoko intuitívnym spôsobom.
- **Jazyková nezávislosť** – v technológii .NET sa všetky jazyky, tj. Visual Basic .NET, C#, J# kompilujú do spoločného sprostredkovacieho jazyka IL (Intermediate language). To znamená, že jazyky môžu spolupracovať spôsobom, ktorý predtým nebol k dispozícii.
- **Lepšia podpora dynamických webových stránok** - .NET ponúka integrovanú podporu webových stránok pomocou novej technológie ASP.NET.
- **Účinnjší prístup k dátam** – sada komponentov .NET súhrne označovaná ako ADO.NET zaisťuje efektívny prístup k relačným databázam a rôznym zdrojom dát.
- **Zdieľanie kódu** – technológia .NET celkom mení koncepciu zdieľania kódu medzi aplikáciami. Zavádza zostavenie (assembly), ktorá nahradzuje tradičnú knižnicu DLL.
- **Zlepšené zabezpečenie**

Jadrom platformy .NET Framework je jeho behové prostredie, ktoré sa označuje ako modul CLR (Common Language Runtime) alebo behový systém .NET. Kód spustený pod kontrolou modulu CLR sa často označuje ako riadený kód (Managed code). Pred spustením v module CLR však ľubovoľný zdrojový kód (v jazyku C#, alebo v niektorom inom) vyžaduje kompiláciu. Kompilácia v prostredí .NET prebieha v dvoch krokoch:

- Kompilácia zdrojového kódu do jazyka IL.
- Kompilácia jazyka IL do kódu špecifického pre cieľovú platformu pomocou modulu CLR.

Sprostredkujúci jazyk IL spoločnosti Microsoft je založený na rovnakej myšlienke ako bajtový kód jazyka Java. Jedná sa o nízkoúrovňový jazyk s jednoduchou syntaxou (ktorá nepracuje s textom, ale číselnými kódmi). Je možné ho veľmi rýchlo preložiť do natívneho strojového kódu. Miesto kompilácie v jednom priechode (znamenalo by to dlhý čas spustenia) kompilátor JIT (just-in-time) jednoducho kompiluje každú časť kódu v okamžiku jeho volania. Vďaka tomu je možné očakávať, že spustenie riadeného kódu jazyka IL bude takmer rovnako rýchle ako spustenie natívneho strojového kódu. Modul CLR obsahuje okrem iného taktiež Garbage Collection, správu pamäte, výnimiek a vlákien [3].

## 2.2 Jazyk C#

Spoločne s platformou .NET sa predstavuje jazyk C#. Je to relatívne nový objektovo orientovaný jazyk významný z dvoch hľadísk:

- Špeciálne navrhnutý firmou Microsoft k použitiu s platformou .NET Framework.
- Jedná sa o jazyk založený na modernej metodike objektovo orientovaného návrhu.

Podrobnejšie informácie možno nájsť v literatúre [3].

## 2.3 ASP.NET

ASP.NET je jednou z najdynamickejšie rozvíjajúcich sa technológií pre vývoj webových aplikácií. Je to plne objektová platforma, základným pilierom je platforma .NET Framework. Jednou z veľkých výhod ASP.NET je možnosť písať kód vo viacerých programovacích jazykoch (CLR), ako to bolo spomenuté v jednej z predchádzajúcich kapitol. Aplikácie v ASP.NET sú predkompilované, čím sa zvyšuje ich rýchlosť oproti ostatným jazykom ako napr. PHP a ASP, ktoré patria do rodiny skriptovacích a pri každom spustení sa musia parsovať. Vytvárané aplikačné súbory majú príponu .aspx a Microsoft u nich aplikoval model oddelenia aplikačnej a prezentačnej vrstvy, to znamená, že oddelil kód od vzhľadu aplikácie a tým pádom sa docielilo zvýšenie prehľadnosti výsledného kódu. Samotný vzhľad je definovaný v súbore s príponou .aspx a kód je umiestnený v súbore s príponou .aspx.cs pokiaľ je použitý jazyk C#.

Hlavné vlastnosti ASP.NET:

- Serverové ovládacie prvky – bohatá kolekcia prvkov ako napr. Button a Textbox, ktoré sú implementované ako objekty u ktorých je možné meniť a nastavovať ich vlastnosti a zachytávať rôzne udalosti.
- Správa členstva a rolí – je to rada rozhraní API, ovládacích prvkov a poskytovateľov, ktoré nám dovoľujú riadiť správu členstva a rolí užívateľov aplikácie. Všetky nástroje získame v už zmienenom API, alebo v administračnom nástroji Web Site Administration Tool.
- Hlavné stránky – vďaka Master Pages môžeme v našich aplikáciách využívať vizuálnu dedičnosť. Umožňuje navrhnuť šablónu a tú potom využívať v celej aplikácii.
- Témy – Jednoduchým spôsobom je možné zaistiť konzistentný vzhľad a fungovanie webového sídla.
- Navigácia – Súčasťou ASP.NET sú navigačné serverové ovládacie prvky, ktoré umožňujú vytvárať mapu sídla (sitemap), ktorý sa ukladá v XML súbore, ktorý môžeme použiť ako zdroj dát napr. ovládacom prvku TreeView.

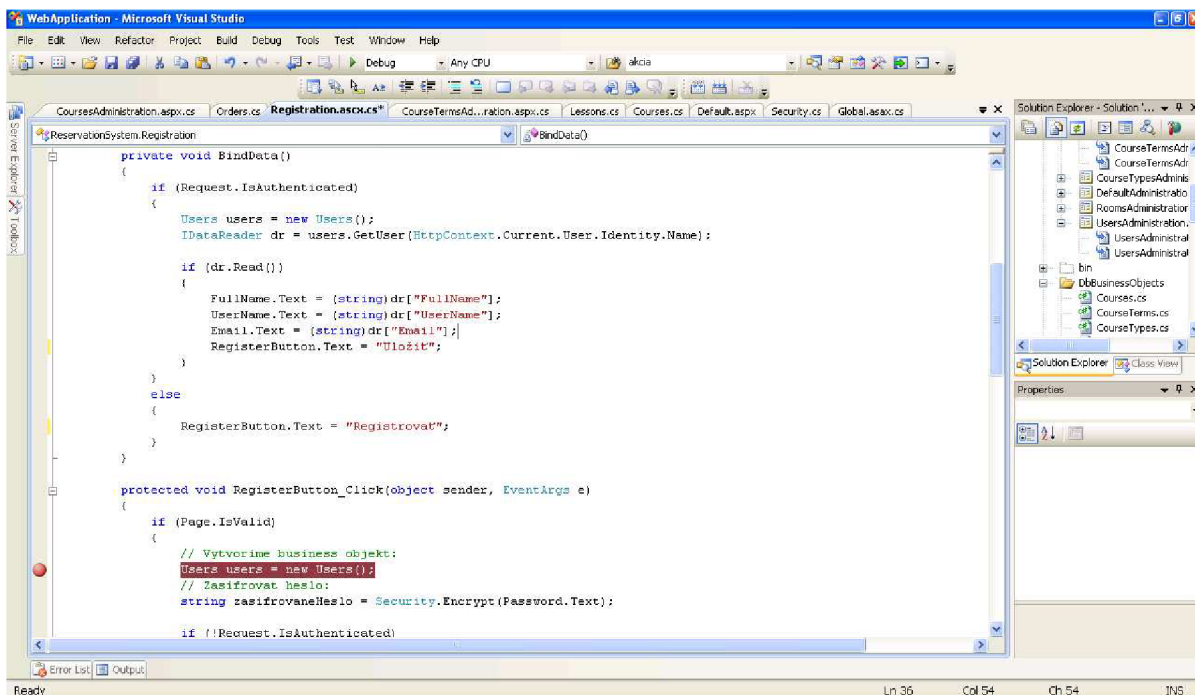
Podrobnejšie informácie možno nájsť v literatúre [4].

## 2.4 Microsoft SQL Server

SQL Server je relačný databázový systém spoločnosti Microsoft. Jedná sa o veľmi robustnú platformu typu klient – server. Ako dotazovací jazyk používa Transact - SQL, ktorý je rozšírením štandardu ANSI/ISO. V súčasnosti je aktuálna verzia nesie označenie 2008. Existujú viaceré edície od komerčných až po voľne šíriteľnú verziu označovanú ako Microsoft SQL Server 2008 Express, ktorá má však viacero obmedzení ako napr. veľkosť databázy, obmedzenie na počet procesorov a maximálnu veľkosť operačnej pamäte. Voliteľnou súčasťou inštalácie je aj Microsoft SQL Management Studio [8] – jedná sa o prepracovaný nástroj na správu nastavení MS SQL, vytváranie databáz, tabuliek, vytváranie dotazov a mnoho ďalších funkcionalít. V IS bola použitá školská verzia prístupná študentom cez MSDN vo verzii Enterprise.

## 2.5 Visual Studio

Produkt Visual Studiu je produktom spoločnosti Microsoft, v súčasnej dobe v najaktuálnejšej verzii 2008 patrí medzi najkvalitnejšie vývojové prostredia pre vývoj konzolových aplikácií, aplikácií s grafickým rozhraním, webových stránok, webových aplikácií a služieb v ASP.NET. Visual Studio dovoľuje vytvoriť akúkoľvek triedu, komponent, alebo aplikáciu pre .NET. Súčasťou je prepracovaný editor kódu s podporou technológie IntelliSense – automatického dokončovania, podpora pre Refactoring a chýbať nesmie taktiež debugger pracujúci na úrovni stroja, tak na úrovni kódu [5].



Obr. 2.2: Ukážka užívateľského rozhrania

# 3 Návrh informačného systému

V nasledujúcich kapitolách bakalárskej práce sa budeme venovať popisu postupu pri analýze a formulovaní požiadaviek pre informačný systém školiaceho strediska.

## 3.1 Účel a cieľ práce

Účelom bakalárskej práce je vytvorenie informačného systému školiaceho strediska. Stredisko poskytuje počítačové školenia z rôznych oblastí. Implementácia musí zahŕňať zákaznícku a lektorskú časť. Z pohľadu lektorskej časti by mal byť kladený dôraz najmä na zjednodušené a prehľadné grafické rezervovanie vyučovacích kapacít strediska – vyučovacích miestností a naplánovanie ich využitia z dlhodobého časového hľadiska. Zákaznícka časť musí umožniť prihlasovanie sa do otvorených kurzov a poskytnúť zákazníkovi požadované informácie.

## 3.2 Analýza požiadaviek

K informačnému systému budú pristupovať jednak samotní zamestnanci – lektori, ale aj zákazníci. Preto pri jeho tvorbe musíme počítať aj so zákazníckou časťou, ktorej verejná časť bude voľne prístupná a na tejto základnej stránke si potenciálni zákazníci môžu prezrieť jednak zoznam všetkých ponúkaných kurzov a zároveň aktuálnu ponuku organizovaných kurzov, ich náplň, vstupné požiadavky, časovú náročnosť, aktuálny počet voľných miest a samozrejme v neposlednom rade aj cenu. Tieto kurzy sú delené do skupín podľa ich zamerania, sú to napr.:

- Kancelárske aplikácie
  - Microsoft Office 2007
  - Openoffice
- Programovacie jazyky
  - Pokročilé programovanie v jazyku Java
  - Programovanie v jazyku C#
- Databázové systémy
  - Microsoft SQL Server 2008
  - MySql

Otvorenie každého kurzu je možné v určitom časovom období a tak je možné naplánovať otváranie jednotlivých kurzov.

Ak sa zákazník rozhodne využiť služby strediska a prihlásiť sa do jedného z kurzov, musí najprv prejsť procesom povinnej registrácie, kde vyplní základné údaje o svojej osobe, prípadne o spoločnosti, ak sa jedná o firemného zákazníka. Po úspešnej registrácii a prihlásení sa do zákazníckej časti je užívateľovi umožnené prihlásenie – zápis do kurzu, kde si môže zobrazit' rozvrh hodín s informáciami:

- Deň a hodinu konania vyučovania.
- Miestnosť.
- Obsadenosť učební, meno školiteľa a ďalšie informácie.

Zamestnanci strediska majú v informačnom systéme možnosť vykonávať viaceré akcie ako napr. :

- Pridávanie nového kurzu pre určité obdobie.
- Správa školiacich hodín pre vybrané školenie
- Odosielanie e-mailov o zrušení vyučovacej hodiny s následnou možnosťou sledovania, či účastník potvrdil prečítanie správy, alebo nie.
- Grafické sledovanie obsadenia učební školiaceho strediska za dané obdobie.

### 3.3 Typy užívateľov prístupujúcich k systému

V prístupe k systému rozlišujeme štyri typy užívateľov a to neregistrovaný užívateľ, študent, lektor a administrátor.

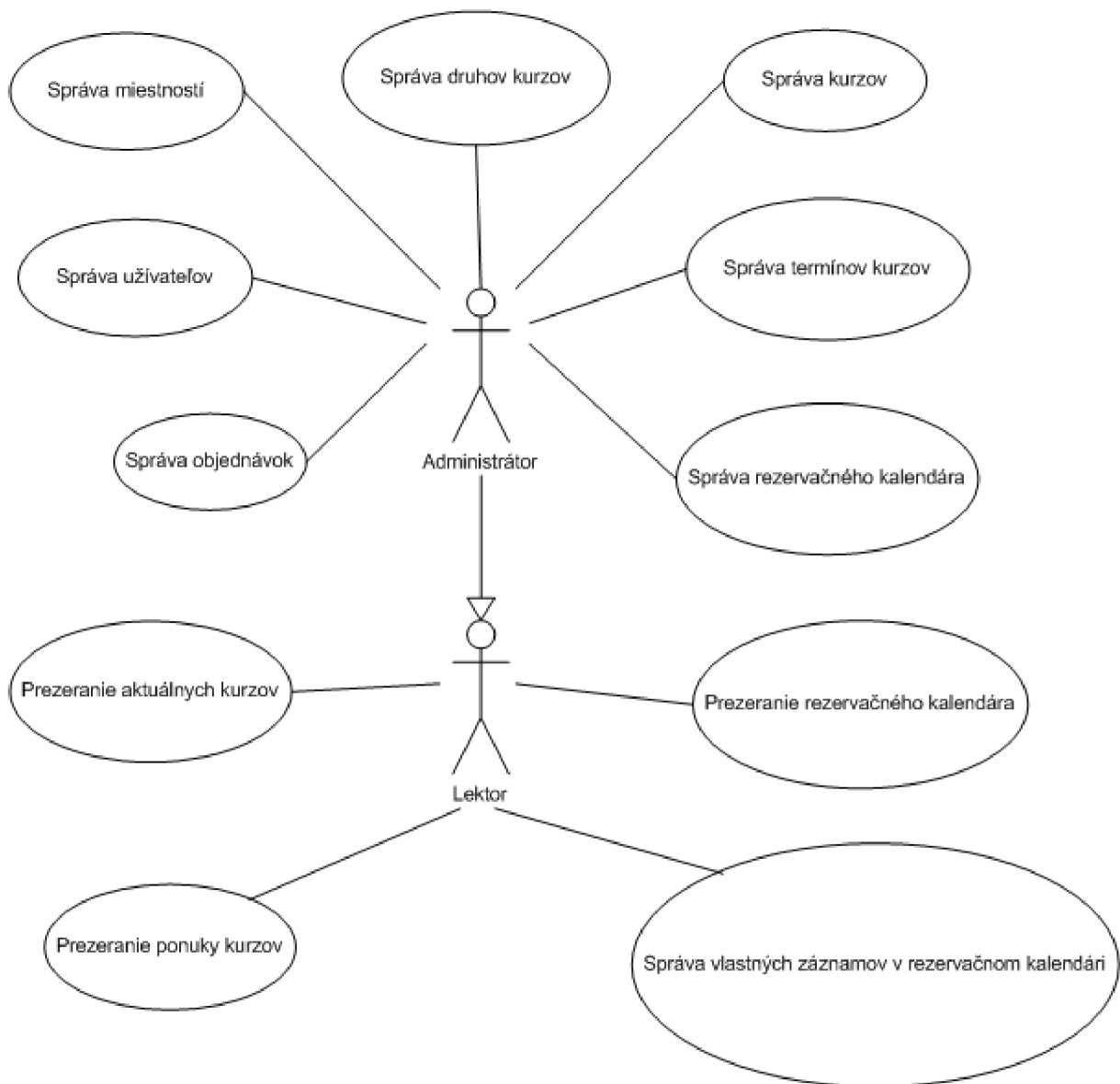
- **Administrátor** – jeho úlohou je správa systému do ktorej patrí správa užívateľov, pridávanie nových typov kurzov do ponuky, otváranie nových v danom časovom období. Má prístup k všetkým častiam systému takže by to mal byť manažér školiaceho strediska s prehľadom o fungovaní firmy, pretože on organizuje a otvára všetky školenia.
- **Lektor** – Jedná sa o zamestnanca, prípadne externého pracovníka školiaceho strediska, ktorého hlavnou úlohou je naplánovať si rozvrh hodín v kurze, kde bol pridelený ako lektor.
- **Študent** – Užívateľskú rolu študent získa návštevník stránok strediska po zaregistrovaní, tým pádom má možnosť zaregistrovať sa do otvorených kurzov a prezerat' si rozvrh hodín.
- **Neregistrovaný užívateľ** – jedná sa o návštevníka stránok strediska, ktorému je umožnené prezerat' si ponuku kurzov a samozrejme prihlásiť prípadne zaregistrovať sa do systému.

## 3.4 Modely systému

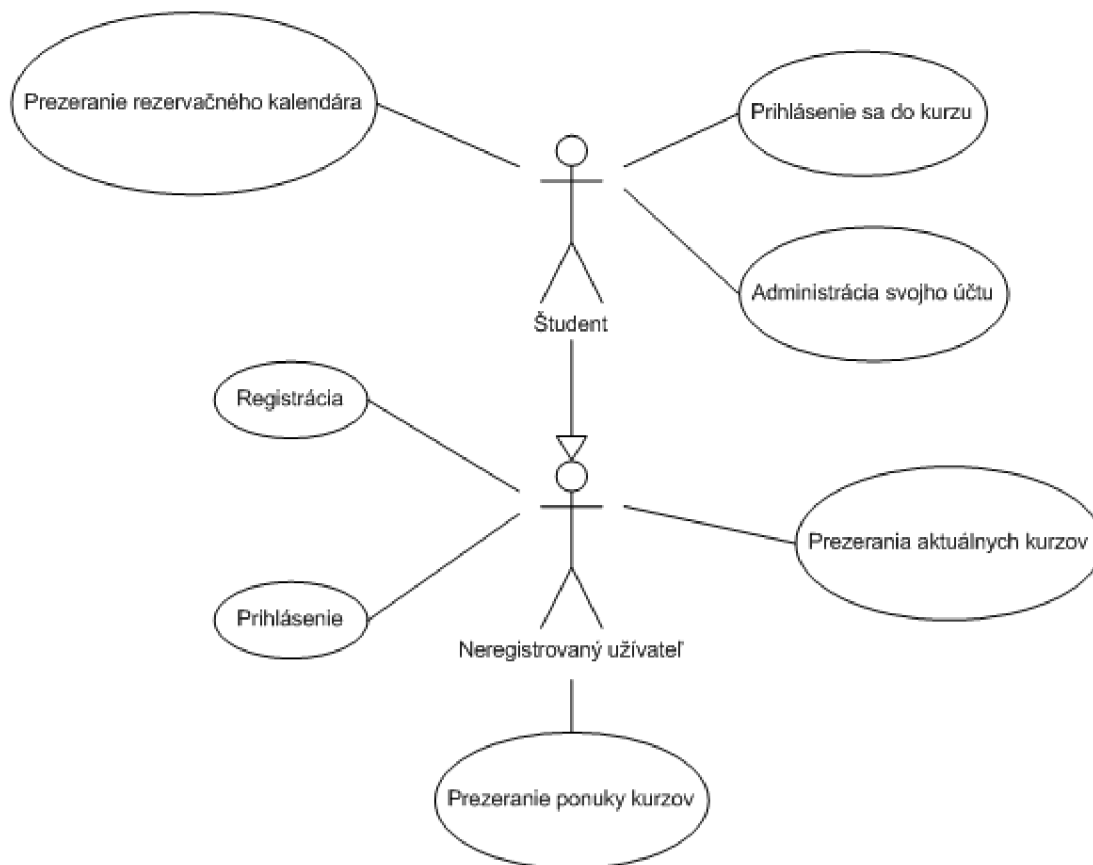
V procese návrhu systému som zostavil dva modely systému v jazyku UML, ktoré určujú aké požiadavky má spĺňať systém, aké akcie môžu vykonávať osoby prístupujúce do systému s pridelenými rôznymi užívateľskými rolami.

### 3.4.1 Diagramy prípadov užitia

Pre vytvorenie diagramov prípadov použitia bol použitý program Microsoft Visio 2007 [6]. Prvý uvedený je prípad použitia lektorskej časti systému, kde vystupujú užívateľské roly Administrátor a lektor. Ako druhý je uvedený prípad použitia zákazníckej časti systému, ako užívateľské roly sú tam zastúpené Študent a neregistrovaný užívateľ.



Obr. 3.1: Diagram prípadov použitia – Administrátor a lektor



Obr. 3.2: Diagram prípadov použitia – Študent a neregistrovaný užívateľ

## 3.4.2 Popis diagramu prípadu použitia lektorskej časti

### 3.4.2.1 Správa miestností

Správa miestností v systéme predstavuje databázu miestností – vyučovacích kapacít ktorými školiace stredisko disponuje a v ktorých prebieha výuka kurzov. Pod správou sa rozumie pridávanie nových záznamov, editáciu existujúcich a prípadné vymazanie záznamov z databázy. Každá miestnosť má svoj jedinečný identifikátor.

### 3.4.2.2 Správa druhov kurzov

Správa druhov kurzov predstavuje oblasti – typy do ktorých jednotlivé vypísané kurzy môžeme zaradiť. Ide napr. oblasť kancelárskych aplikácií, Internetu, alebo oblasť programovacích jazykov. Operácie, ktoré je možné s dátami prevádzať sú pridávanie, editáciu a mazanie existujúcich záznamov.

### **3.4.2.3 Správa kurzov**

Ide o evidenciu jednotlivých kurzov, ktoré má školiace stredisko v ponuke. Každý kurz patrí do určitej skupiny – druhu kurzu vid'. 3.4.2.1. Záznam o kurze nesie viaceré informácie ako je osnova kurzu, cena, rozsah školenia v dňoch a ďalšie. Správa zahŕňa pridávanie, mazanie a editáciu existujúcich záznamov.

### **3.4.2.4 Správa termínov kurzov**

Správa termínov kurzov umožňuje riadiacemu pracovníkovi - administrátorovi systému otváranie nových kurzov v danom časovom období. Jeden kurz trvajúci ktorého vyučovanie trvá napr. jeden týždeň môže byť otváraný v dvojtýždňových intervaloch. Týmto je možné dopredu naplánovať otváranie kurzov a umožniť prihlasovanie zákazníkov dopredu. Každý termín charakterizuje názov otváraného kurzu, dátum začiatku a ukončenia kurzu. Záznamy je možné pridávať, editovať a mazať.

### **3.4.2.5 Správa kalendára vyučovacích hodín**

Súčasťou systému je rezervačný kalendár miestností, kde si lektori môžu rezervovať kapacitu učební pre nimi vyučované školenia. Administrátor disponuje právami vidieť, pridávať a mazať všetky rezervácie vykonané ním. Súčasťou rezervačného záznamu sú údaje o vlastníkovi rezervácie, miestnosti, voľnej kapacite a ďalších údajoch.

### **3.4.2.6 Správa objednávok**

Predstavuje sledovanie obsadenosti jednotlivých termínov – koľko prihlásených zákazníkov je v systéme zaevidovaných a sledovanie či bol uhradený poplatok za dané školenie.

### **3.4.2.7 Správa užívateľov**

Správa užívateľov umožňuje pridávanie nových užívateľov do systému, či už ide o zamestnancov, alebo zákazníkov strediska. Administrátor má právo vidieť a editovať údaje o všetkých užívateľoch. Dôležitou vlastnosťou v správe je pridelovanie užívateľov do rolí, ktoré im povoľujú, prípadne zamedzujú prístup do jednotlivých sekcií systému.

### **3.4.2.8 Prezeranie aktuálnych kurzov**

Na hlavnej stránke systému je možnosť prezerat' si všetky aktuálne otvorené kurzy do ktorých je možné sa registrovať. Nové termíny kurzov vytvára užívateľ strediska s priradenou rolou Administrátor.

### **3.4.2.9 Prezeranie rezervačného kalendára**

Užívateľovi je umožnené prezeranie rezervačného kalendára, kde vidí grafickú obsadenosť učební, kedy a kde sa konajú jednotlivé vyučovacie hodiny kurzov a v ktorých termínoch je ešte voľná vyučovacia kapacita učební.



#### **3.4.2.10 Správa vlastných záznamov v rezervačnom kalendári**

Lektori školiaceho strediska, vidia v rezervačnom kalendári všetky záznamy, ale majú právo na editáciu len vlastných rezervačných záznamov. Pod správou rozumieme pridávanie a vymazanie rezervácie z kalendára, čím sa uvoľní kapacita miestnosti a je prístupná ostatným lektorom, ktorí ju môžu využiť.

### **3.4.3 Popis diagramu prípadu užitia zákazníkovej časti**

#### **3.4.3.1 Prihlásenie sa do kurzu**

Prihlásenie sa do kurzu reprezentuje možnosť záväzného prihlásenia sa zákazníka do otvoreného kurzu. Predpokladom úspešného prihlásenia je registrácia do systému a zadanie základných osobných údajov.

#### **3.4.3.2 Administrácia svojho účtu**

Zaregistrovaný zákazník má možnosť zmeny svojich registračných údajov, ide najmä o zmenu hesla ďalších údajov.

#### **3.4.3.3 Prihlásenie**

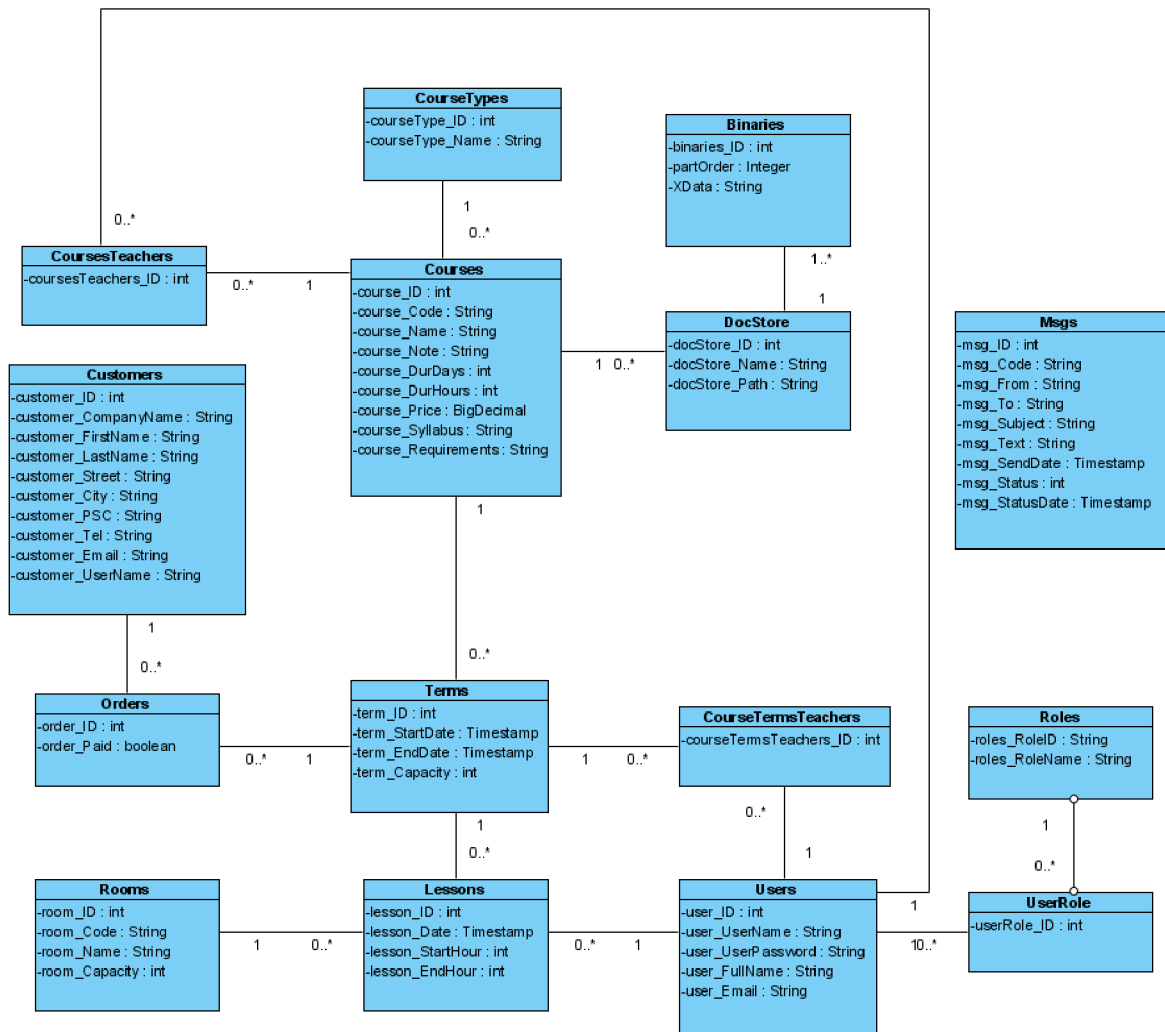
Prihlásenie reprezentuje overenie platnosti prihlasovacích údajov – mena a hesla. Overovanie hesla v databáze je šifrované.

#### **3.4.3.4 Registrácia**

Registrácia umožní zákazníkovi vytvoriť si pre prístup do systému a prihlasovanie ku kurzom platné prihlasovacie údaje.

### 3.4.4 ER diagram

ER diagram mnou navrhnutého systému.



Obr. 3.3 ER diagram

### 3.4.5 Popis databázových tabuliek

#### 3.4.5.1 Users

Tabuľka v ktorej sa ukladajú záznamy o užívateľoch prístupujúcich do systému, jednak zamestnancov strediska a taktiež aj zákazníkov. Každý užívateľ má svoj jednoznačný identifikátor `User_ID` a prihlasovacie meno `User_UserName`. Heslo v databáze je uložené v `User_UserPassword` zašifrované v podobe hash kódu vytvoreného šifrovacím algoritmom MD5.

### **3.4.5.2 Roles**

Tabuľka v ktorej sú uložené typy užívateľských rolí, v tomto systéme sú to nasledujúce tri:

- Administrátor
- Lektor
- Študent

### **3.4.5.3 UserRole**

Definuje priradenie rolí k danému užívateľovi, ktorých môže mať pridelených viacero súčasne. Administrátor môže byť napr. súčasne aj lektorom.

### **3.4.5.4 CourseTypes**

Tabuľka pre ukladanie záznamov o druhoch kurzov ponúkaných školiacim strediskom. Záznam je charakterizovaný jednoznačným identifikátorom a názvom skupiny kurzov.

### **3.4.5.5 Courses**

Údaje o samotných kurzoch sú uložené v tabuľke Courses. Každý kurz má uvedený jednoznačný identifikátor Course\_ID, kód kurzu Course\_Code, osnovu kurzu Course\_Syllabus, cenu Course\_Price a ďalšie údaje.

### **3.4.5.6 CoursesTeachers**

Každý kurz v ponuke strediska má pridelených Lektorov, ktorý majú potrebné znalosti, aby mohli daný kurz vyučovať. Záznam v tejto tabuľke odkazuje jedným cudzím kľúčom do tabuľky užívateľov Users a druhým do tabuľky kurzov Courses, tak máme zadané jednoznačné priradenie lektora ku kurzu.

### **3.4.5.7 Terms**

Termíny kurzov sa ukladajú do tabuľky Terms, každý kurz je charakterizovaný dátumom kedy začína, končí a kapacitou kurzu – koľko zákazníkov sa môže do kurzu prihlásiť.

### **3.4.5.8 CourseTermsTeachers**

Každý kurz má pridelených svojich špecializovaných lektorov z ktorých len niektorý môžu vyučovať práve otvorený termín kurzu. Pri otvorení nového termínu kurzu si vyberáme z užívateľov, ktorý sú pridelený ku kurzu v tabuľke CoursesTeachers. Takže táto tabuľka uchováva záznamy o tom, ktorý zo špecializovaných lektorov pre daný kurz vyučuje otvorený termín kurzu.

#### **3.4.5.9 Lessons**

Tabuľka Lessons slúži pre ukladanie záznamov o vyučovacích hodinách v termínoch kurzov, ktoré reprezentuje tabuľka Terms. Vyučovacia hodina má jednoznačný identifikátor Lesson\_ID, identifikátor miestnosti v ktorej sa koná vyučovanie Lesson\_RoomID, identifikátor prideleného Lektora Lesson\_LectorID, dátum a čas konania vyučovania.

#### **3.4.5.10 DocStore**

Ku každému kurzu je možné pripojiť dokumenty potrebné k výuke a tieto si zákazníci môžu dopredu stiahnuť a preštudovať. Tabuľka obsahuje cudzí kľúč DocStore\_CourseID, ktorý odkazuje do tabuľky Courses, tým je možné jednoznačne priradiť dokumenty k danému kurzu.

#### **3.4.5.11 Binaries**

V tejto databázovej tabuľke sú ukladané súbory, ktoré sa pridávajú ku kurzom. Pripojený súbor sa ukladá priamo do databázy. Ide o dokumenty potrebné k štúdiu.

#### **3.4.5.12 Rooms**

Tabuľka Rooms ukladá záznamy o vyučovacích miestnostiach, ktorými disponuje školiace stredisko. Hlavnou charakteristikou každej miestnosti je jej identifikátor Room\_Code a kapacita Room\_Capacity.

#### **3.4.5.13 Orders**

Po záväznej objednávke zákazníka sa do tabuľky Orders zapíše záznam o tom, ktorý zákazník zaslal objednávku Order\_CustomerID, do akého kurzu sa zapísal Order\_TermID. Posledným údajom na objednávke je príznak uhradenia požadovanej čiastky za školenie Order\_Paid.

#### **3.4.5.14 Customers**

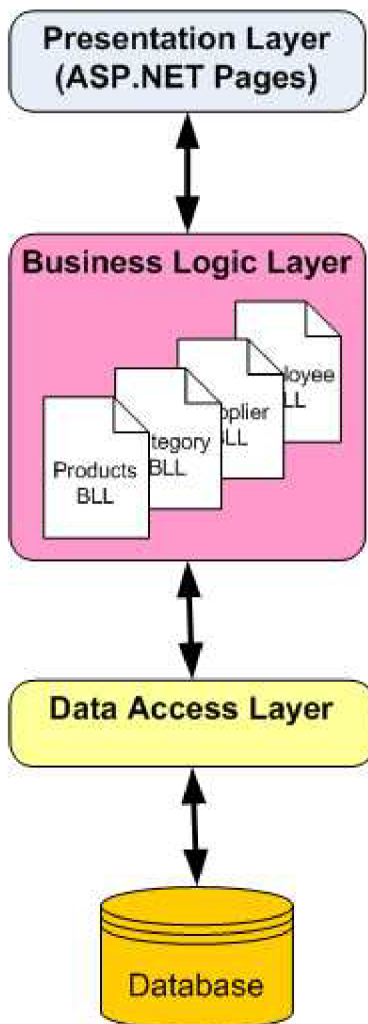
Tabuľka Customers uchováva osobné údaje zaregistrovaných zákazníkov, ktoré je potrebné vyplniť pred odoslaním záväznej objednávky. Je relačne previazaná s tabuľkou Users pomocou cudzieho kľúča Customer\_UserName, ktorý ukazuje do tabuľky užívateľov systému v ktorej sú uložené prihlasovacie údaje každej osoby prístupujúcej do systému.

#### **3.4.5.15 Msgs**

Po zrušení rezervácie vyučovacej hodiny v kalendári sa všetkým zákazníkom, ktorým sa zrušilo vyučovanie rozpošlú e-maily s oznamom o tejto zmene v rozvrhu. Každý vygenerovaný e-mail má jednoznačný identifikátor Msg\_Code, údaje o odosielateľovi a adresátovi a príznak prečítania správy Message\_Read, ktorého hodnota sa mení po potvrdení o oboznámení sa so zmenou v rozvrhu.

## 3.5 Štruktúra aplikácie

V poslednej dobe sa v oblasti vývoja software sa stal viacúrovňový návrh známy. Zjednodušene povedané rozdeľuje funkčné prvky, komponenty a kód projektu do samostatných úrovní. Na obecnej úrovni existujú štyri.



Obr. 3.5.1: Viacvrstvová architektúra aplikácie [7]

- **Dátové úložisko – Databáza**  
Uchováva dáta. Všeobecne môže ísť o relačnú databázu, XML súbor, alebo textový súbor.
- **Vrstva prístupu k dátam (Data Access Layer - DAL)**  
Kód, ktorý zabezpečuje získavanie a manipuláciu zdrojových dát uložených v dátovom úložisku.
- **Vrstva aplikačnej logiky (Business Logic Layer - BLL)**  
Kód, ktorý berie dáta získané dátovou úrovňou a ponúka ju klientom v abstraktnejšej a intuitívnejšej forme, pričom skrýva podrobnosti na nižšej úrovni napr. schéma uloženia dát a pridáva všetku validačnú logiku, ktorá zaisťuje bezpečný a konzistentný vstup dát.

- **Prezentačná vrstva (užívateľské rozhranie - UI)**

Kód, ktorý definuje, čo užívateľ vidí na obrazovke, vrátane formulárov dát a navigačných menu.

Prevzaté z [4].

V tomto systéme je ako dátové úložisko používaná relačná databáza Microsoft SQL Server 2008, ktorej štruktúra a popis tabuliek boli opísané v predchádzajúcich kapitolách. Vo vrstve prístupu k dátam DAL existujú dva odlišné návrhové modely pre prístup k databáze. Používané SQL dotazy sú uložené buď v uložených procedúrach. Pričom na ich spustenie sa používa objekt SqlCommand, ktorý ako parameter používa meno uloženej procedúry. Druhý model ukladá SQL dotazy vo forme textu, ktorý sa potom spúšťa štandardne pomocou SqlCommand. V tomto informačnom systéme sú použité uložené procedúry, ktorých hlavnou výhodou je rýchlosť a bezpečnosť v prístupe k dátam. Prístup k určitej uloženej procedúre povolíme len určitému webovému SQL účtu [4].

Vytvárať a spravovať uložené procedúry môžeme pomocou nástroja Microsoft SQL Server Management Studio [8], kde v strome nasledujeme cestu Databases – ReservationSystem (meno našej databázy) – Programmability – Stored Procedures. Ukážka kódu uloženej procedúry, ktorá reprezentuje vrstvu DAL:

```
USE [ReservationSystem]
GO
/***** Object:  StoredProcedure [dbo].[COURSESGetCourse]    Script Date:
05/16/2009 17:17:15 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[COURSESGetCourse]
    @CourseId int
AS
BEGIN
    SELECT c.Course_ID as CourseId,
           c.Course_Name as CourseName,
           c.Course_Code as CourseCode,
           c.Course_Note as CourseNote,
           c.Course_Price as CoursePrice,
           c.Course_DurDays as CourseDurDays,
           c.Course_DurHours as CourseDurHours,
           c.Course_CourseTypeId as CourseTypeId,
           c.Course_Syllabus as CourseSyllabus,
           c.Course_Requirements as CourseRequirements
    FROM dbo.Courses c
    WHERE c.Course_ID = @CourseId

END
```

Vrstvu aplikačnej logiky BLL zabezpečujú triedy, ktoré sú umiestnené v adresári DbBusinessObjects. Tieto triedy obsahujú metódy pre prácu s jednotlivými objektmi ako napr. trieda Courses.cs obsahuje metódy: `GetCourse()`, `AddCourse()`, `GetLectorsForCourse()` a ďalšie.

Adresár DbBusinessObjecst obsahuje tieto triedy:

- **Courses.cs**
- **CourseTerms.cs**
- **CourseTypes.cs**
- **Customers.cs**
- **DocStore.cs**
- **Lesson.cs**
- **Lessons.cs**
- **Msgs.cs**
- **Orders.cs**
- **Room.cs**
- **Rooms.cs**
- **Users.cs**

## 4 Implementácia

Nasledujúca kapitola sa zaoberá detailným popisom implementácie systému a implementovanými funkciami pri ktorých nechýba podrobný popis ich funkčnosti. V úvode vysvetlím inštaláciu webového serveru IIS a vytvorenie databázovej štruktúry. Ďalší obsahom tejto kapitoly bude popis administrátorskej a lektorskej časti systému.

Ako webový server bol použitý server firmy Microsoft Internet Information Services (IIS) [9]. Inštaláciu tejto služby môžeme previesť na operačných systémoch vyrábaných taktiež spoločnosťou Microsoft, ide napr. o Microsoft Windows XP, Vista alebo serverové verzie Windows Server 2003 prípadne Windows Server 2008. V operačnom systéme taktiež nesmie chýbať nainštalovaný .NET Framework minimálne vo verzii 2.0 (viď. kapitola 2.1 .NET Framework).

### 4.1 Vytvorenie databázovej štruktúry

Ako databázová platforma bol použitý MS SQL Server 2008 (podrobnejšie informácie viď. Kapitola 2.4 SQL Server). Súčasťou súborov so zdrojovým kódom je backup súbor s kompletnou zálohou databázy informačného systému, ktorá obsahuje 15 tabuliek, ktoré sú naplnené údajmi testovacími

dátami, ale taktiež údajmi potrebnými pre fungovanie systému. Ich štruktúra, vzťahy a atribúty sú vytvorené podľa ER diagramu (viď. kapitola 3.4.4 ER diagram). Databáza ďalej obsahuje uložené procedúry slúžiace na manipuláciu s dátami. Tento backup súbor sa nachádza v hlavnom adresári so súborami so zdrojovým kódom, má názov `ReservationSystem.bak`. Tento súbor je nutné použiť pri obnove databázy – Restore database pomocou nástroja Microsoft SQL Server Management Studio.

## 4.2 Implementované časti systému

V tejto kapitole budú popisované obidva implementované časti systému a ich funkcionality. Najprv to bude verejná časť systému a následne časť lektorská určená zamestnancom.

### 4.2.1 Vzhľad systému

Na obrázku 4.2.21 je ukážka z hlavnej stránky systému.



The screenshot shows the main page of a reservation system. At the top left is a computer icon and the title "Rezervačný systém školiaceho strediska". At the top right are links for "Prihlásiť" and "Registrovať". Below the title is a section titled "Prehľad termínov ponúkaných kurzov:". There are two course listings, each with a small 3D map icon. The first listing is for "Adobe Photoshop (APH)" with details: Termín: 18. mája 2009 - 20. mája 2009, Kapacita: 25, Už prihlásených: 1, Voľných miest: 24, Cena: 5 000,00 Sk, Doba trvania kurzu: 3 dní (celkom 24 hodín). The second listing is for "Základy programovania v jazyku Java (JVA)" with details: Termín: 18. mája 2009 - 20. mája 2009, Kapacita: 20, Už prihlásených: 2, Voľných miest: 18, Cena: 4 200,00 Sk, Doba trvania kurzu: 3 dní (celkom 24 hodín).

Obr. 4.2.21 Hlavná stránka

### 4.2.2 Verejne prístupná časť

Táto sekcia je verejne prístupná a určená zákazníkom – záujemcom o školenia ponúkané školiacim strediskom.

#### 4.2.2.1 Registrácia a zmena registračných údajov

Každý návštevník stránok má možnosť zaregistrovať sa do systému a využiť jeho ďalšiu funkcionality, alebo len anonymne prezerat' stránky. Zaregistrovaný užívateľ môže prezerat' zoznam všetkých ponúkaných kurzov a samozrejme ich aktuálne vypísané termíny. Pri termínoch je zobrazené množstvo potrebných informácií, najmä cena kurzu, doba trvania v dňoch a hodinách, počas ktorých vyučovanie trvá. Registrovaný užívateľ vidí podrobnosti kurzu – jeho vyučovaciu osnovu, poznámky a vstupné požiadavky, ktoré musí spĺňať, aby mohol daný kurz úspešne



absolvovať. Registrácia je samozrejme nutným krokom pred samotným prihlásením sa do vypísaných termínov.

---

## Registračné údaje

Užívateľské meno*:	<input type="text"/>
Heslo*:	<input type="password"/>
Potvrdenie hesla*:	<input type="password"/>
Meno*:	<input type="text"/>
E-mail*:	<input type="text"/>

---

### Rezervačný systém školiaceho strediska

Obr. 4.2.1 – Registrácia nového užívateľa

Všetky údaje v registračnom formulári sú povinné, zákazník si vyberá svoje užívateľské meno, ktoré sme byť v systéme jednoznačné preto sa pri registrácii kontroluje či už neexistuje v databáze rovnaké. Táto skutočnosť je následne užívateľovi oznámená. Heslo je do databázy uložené v zašifrovanom tvare pomocou hešovacieho algoritmu MD5. Každé z polí je opatrené validačnými prvkami patriacimi medzi štandardné serverové ovládacie prvky v ASP.NET. Zhodu zadaných hesiel kontroluje validátor `<asp:CompareValidator>`.

Zmenu registračných údajov reprezentuje totožná obrazovka, kde si svoje registračné údaje môže zmeniť.

#### 4.2.2.2 Prihlásenie a odhlásenie

Zaregistrovaný užívateľ má možnosť prihlásiť sa do systému po kliknutí na odkaz „Prihlásiť“ na hlavnej stránke. Zobrazí sa prihlasovací formulár:

---

Užívateľské meno:

Heslo:

Uložiť heslo

**Prihlásiť**

---

**Rezervačný systém školiaceho strediska**

Obr. 4.2.2 – Prihlásenie do systému

Po vyplnení prihlasovacích údajov a kliknutí na prihlásiť sa správnosť zadaných údajov overí v databáze. Heslo sa najprv zašifruje hešovacím algoritmom MD5 a porovnáva sa hešom uloženým v databáze. Ak sa jedná o platné prihlásenie pošleme užívateľovi autentifikačné cookie.

Odhlásenie užívateľa prebehne zavolaním metódy `FormsAuthentication.SignOut()`, ktorá zabezpečí odstránenie autentifikačného tiketu z cookie.

#### 4.2.2.3 Prihlásenie do kurzu

Po zaregistrovaní má užívateľ možnosť prezrieť si podrobnejšie informácie o aktuálnych termínoch kurzov, ale predtým ešte musí zadať svoje podrobné zákaznícke údaje, ako je presná adresa, telefónne číslo, email a prípadne meno firmy, ak sa jedná o firemného zákazníka. Po vyplnení registračných údajov zákazníka je následne užívateľovi v systéme pridaná rola „Študent“, ktorá zabezpečí oprávnenie prezerat' spomínané detaily vypísaných kurzov. Vid' obr. 4.2.3.

### Prehľad termínov ponúkaných kurzov:

#### Adobe Photoshop (APH)

**Termín:** 18. mája 2009 - 20. mája 2009  
**Kapacita:** 25  
**Už prihlásených:** 1  
**Voľných miest:** 24  
**Cena:** 5 000,00 Sk  
**Doba trvania kurzu:** 3 dni (celkom 24 hodín)



Obr. 4.2.3 Prehľad termínov ponúkaných kurzov

Po kliknutí na podrobnosti kurzu sa dostaneme na obrazovku:

## Podrobnosti kurzu

### MS Exchange (EXCH)

**Termín:** 19. mája 2009 - 22. mája 2009  
**Kapacita:** 15  
**Už prihlásených:** 0  
**Voľných miest:** 15  
**Cena:** 13 500,00 Sk  
**Doba trvania kurzu:** 5 dní (celkom 40 hodín)

#### Osnova

Poštovní server Exchange je silným nástrojom pro poštovní služby. Díky jeho výborné spolupráci s prostředím Windows umožňuje kvalitní správu. Naučíte se nejen obsluhu a filozofii tohoto serveru, ale také návrhu kompletního prostředí pro poštovní služby od středních až po velké firmy. Bude si moci vyzkoušet různé situace, které v praxi mohou nastat a proniknete rychle do správy tohoto rozsáhlého systému včetně zálohování a správy databáze poštovních schránek.

#### Požiadavky

Znalosť OS Windows Server.

#### Poznámky

Kurz venovaný správě poštovního serveru od instalace po konfiguraci a zabezpečení

[Závazne objednať kurz](#)

#### Rezervačný systém školiaceho strediska

Obr. 4.2.4 Zobrazenie podrobností kurzu s možnosťou prihlásenia

V detaile má zákazník možnosť vidieť aktuálny počet prihlásených, počet voľných miest a iné. Po kliknutí na tlačidlo „Závazne objednať kurz“ je do systému vygenerovaná záväzná objednávka a dôjde k presmerovaniu na „Rozvrh hodín“, kde si môže prezerat' vypísané vyučovacie hodiny v kurzoch, kde je prihlásený. U zapísaných kurzov je možné stiahnuť si zverejnené dokumenty potrebné k vyučovaniu.

#### Poznámky

Zpracování fotografií v prostředí profesionálního nástroje Adobe Photoshop


Název dokumentu	Název súboru	
Učebnica	eng_ap1130ag_radius.pdf	<a href="#">Stiahnuť</a>

Obr. 4.2.5 Dokumenty na stiahnutie

## Rozvrh hodín

máj 2009						
po	ut	st	št	pi	so	ne
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Deň	Miestnosť	1. hodina 7:00 - 7:50	2. hodina 8:00 - 8:50	3. hodina 9:00 - 9:50
PO 18. mája 2009	UG1 <sub>25</sub>			
	UKA <sub>40</sub>			
	UP1 <sub>20</sub>	<div style="border: 1px solid red; padding: 5px;"> <p><b>JVA</b> 2/20</p> <p>Kurz: JVA - Základy programovania v jazyku Java            Termín kurzu: 18. 5. 2009 - 20. 5. 2009            Kapacita termínu: 20            Prihlásených študentov: 2            Miestnosť: UP1 - Učebňa programovacích jazykov 1            Kapacita miestnosti: 20            Vypísal lektor: Skúšobný lektor 1            Začiatok lekcie: 18. 5. 2009 - 1. hodina            Koniec lekcie: 18. 5. 2009 - 1. hodina</p> </div>		
	UP2 <sub>30</sub>		<div style="border: 1px solid red; padding: 5px;"> <p><b>APH</b> 1/25 </p> </div>	

Obr. 4.2.6 Detail vyučovacej hodiny v rozvrhu hodín

Každý záznam vyučovacej hodiny v rozvrhu hodín nesie informácie o tom, kto danú hodinu vypísal – ktorý lektor, o aký kurz sa jedná a počet voľných miest. Zobrazovanie rozvrhu sa ovláda kalendárom vľavo hore. Zobrazuje sa stále týždenný rozvrh hodín vo všetkých miestnostiach, takže stále je to od prvého dňa týždňa – Pondelok až Piatok. Pokiaľ klikneme v kalendári napr. na sobotu, čo je 6. Deň v týždni, musíme spraviť korekciu, aby sa kalendár načítal stále od začiatku týždňa. To je zabezpečené nasledovne:

```
DateTime startDate = Date.AddDays(-((int)(Date.DayOfWeek - 1)))
```

























Pokiaľ v kalendári označíme nedeľu, `DayOfWeek` vráti hodnotu 0, takže vo výsledku sa dátum nastaví pondelok nasledujúceho týždňa.

### 4.2.3 Lektorská časť systému

Lektorská časť systému slúži pre zamestnancov školiaceho strediska u ktorých rozlišujeme dva typy rolí, ktoré môže mať užívateľ pridelené. Sú to roly Administrátor a Lektor. Rola Administrátor reprezentuje v systéme zamestnanca školiaceho strediska, ktorý ma určitú riadiacu funkciu, spravuje a riadi teda chod celého školiaceho strediska. Má prístup ku všetkým častiam systému. Rolu Lektor má pridelenú užívateľ určený ako školiteľ pre jednotlivé kurzy, ten si organizuje a plánuje vyučovacie hodiny v kurzoch kde je pridelený.

### 4.2.3.1 Rezervačný kalendár

Jednou z hlavných funkcionalít mnou navrhnutého informačného systému je rezervačný kalendár. Kalendár zobrazuje týždenný rozvrh hodín v učebniach, ktorými disponuje školiace stredisko. Počet stĺpcov kalendára je pevne daný a obsahuje, každý stĺpec reprezentuje jednu vyučovaciu hodinu, ktorá trvá 50 minút. Každú celú hodinu začína jedna vyučovacia jednotka, teda napr. 8.00 – 8.50. Kalendár zobrazuje 12 vyučovacích jednotiek, počnúc 7.00 až 18.50. Riadky tabuľky tvoria pracovné dni týždňa, teda pondelok až piatok, pričom každý z riadkov je zložený z ďalších riadkov reprezentujúcich miestnosti – učebne, kde je možné organizovať vyučovanie. Každá z miestností má svoje označenie a taktiež je zobrazená jej kapacita, viď obr. 4.2.7.

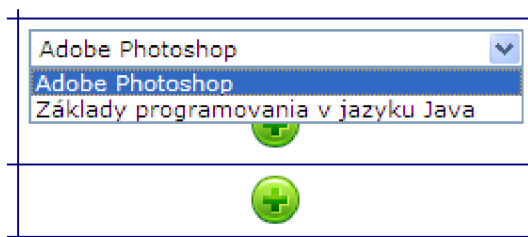
Deň	Miestnosť	1. hodina 7:00 - 7:50	2. hodina 8:00 - 8:50	3. hodina 9:00 - 9:50	4. hodina 10:00 - 10:50	5. hodina 11:00 - 11:50	6. hodina 12:00 - 12:50
PO 25. mája 2009	UG1 <sub>25</sub>						
	UKA <sub>40</sub>						
	UP1 <sub>20</sub>						
	UP2 <sub>30</sub>						

Obr. 4.2.7 Ukážka rezervačného kalendára

Na obrázku 4.2.7 som uviedol ukážku prázdneho rezervačného kalendára. Prázdne vyučovacie jednotky sú vyobrazené ikonou znamienka „+“ zelenej farby. Samotnú rezerváciu vykonáme kliknutím na túto ikonu, ktorá spôsobí zavolanie metódy `GetCurrentCourseTerms`, tá spustí kód uloženej procedúry `COURSETERMSGetCurrentCourseTerms`. Procedúra z databázy vráti aktuálne termíny kurzov, pre ktoré platí:

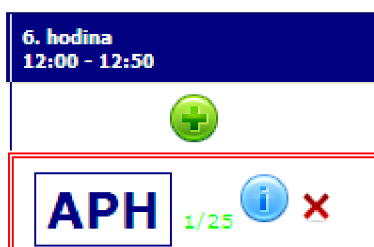
- Aktuálny dátum je medzi dátumom začiatku a dátumom ukončenia kurzu.
- Aktuálne prihlásený Lektor je medzi lektormi, ktorí sú oprávnený vyučovať tento kurz.
- Kapacita miestnosti vybratej kliknutí na vyučovaciu jednotku v kalendári je  $\geq$  ako je kapacita kurzu. Každý účastník kurzu „musí mať svoje miesto“.

Termínmi, ktoré vyhovujú popísanej podmienke naplníme `DropDownList`. Z neho následne lektor vyberie ku ktorému termínu chce rezervovať vyučovaciu hodinu. Viď. obr 4.2.8.



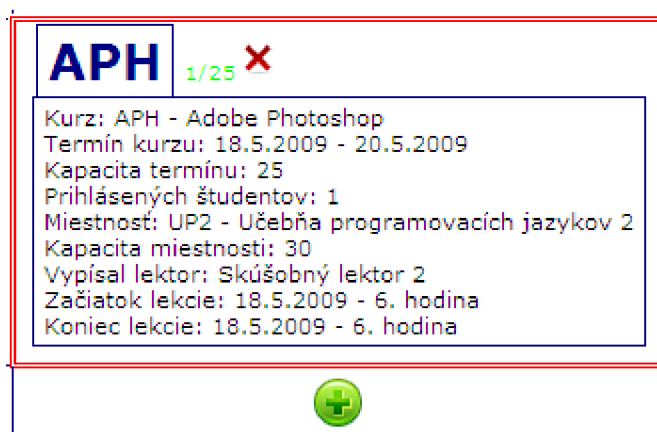
Obr. 4.2.8 Rezervácia vyučovacej hodiny

Rezerváciu vyučovacej hodiny dokončíme stlačením ikony s piktogramom „+“. Po dokončení procesu registrácie je v rezervačnom kalendári zobrazená aj novo pridaná rezervácia. Manipulovať s rezerváciami je možné len tom, kto ju vytvoril, teda vlastníkovi. Vlastník má následne možnosť svoju rezerváciu zmazať vid'. obr. 4.2.9.



Obr. 4.2.9 Rezervácia

Pri samotnej rezervácii je uvedený kód kurzu, ktorému rezervácia patri, ďalej je možné zobrazit' podrobnosti a prípadne ju môže vlastník zmazať vid'. obr. 4.2.10.



Obr. 4.2.10 Detail rezervácie

V každodennom živote sa taktiež stáva, že nami naplánované skutočnosti sa veľa krát menia a tak je to napr. aj so zmenou v rozvrhu. Lektor môže svoju rezerváciu učebne zrušiť a naplánovať ju v iný čas. Zákazníkov zapísaných do daného kurzu je však nutné o tejto neočakávanej skutočnosti

oboznámiť. Pri procese mazania rezervácie sa na pozadí spúšťa proces, ktorý všetkým zákazníkom prihlásených do kurzu, ktorého vyučovacia hodina sa ruší, automaticky odošle email s oboznámením.

### Zrušenie lekcie

Martin Lacko [martin.lacko@umakov.cz]

Odesláno: štv 14.5.2009 11:48

Komu: martin.lacko@umakov.sk

---

Oznamujeme vám, že sa ruší vyučovanie.

**Dátum:** 20. mája 2009

**Hodina:** 2

**Dôvod:** Lekcia zrušená.

**Lektor:** Skúšobný lektor 2

---

Potvrďte prosím, prečítanie tohto oznámenia kliknutím na odkaz [TU](#).

Obr. 4.2.11 Ukážka automaticky generovaného emailu

Každá z odoslaných emailových správ je uložená v databázovej tabuľke `Msgs` a má charakteristický jedinečný identifikátor vytvorený pomocou metódy `Guid guid = Guid.NewGuid()`. `GUID` [10] je skratkou z `Globally unique identifier`. Ide vlastne o 128 bitový integer (16 bajtov), ktorý je možné použiť všade tam, kde je potrebné používať určitý jedinečný identifikátor. Toto vygenerované `GUID` sa spojí s adresou `www` serveru kde je tento systém umiestnený. Adresa `www` servera je zadaná v konfiguračnom súbore `Web.config` v sekcii `<appSettings>` pod kľúčom `ConfirmPage`, ktorý používajú všetky webové aplikácie vytvorené v `ASP.NET`.

```
<add key="ConfirmPage" value="http://212.5.194.176:9999/MailConfirm.aspx" />
```

Spojením adresy serveru a identifikátora `GUID` vznikne hypertextový odkaz ako napr.

<http://212.5.194.176:5000/MailConfirm.aspx?code=20568ce2-902d-499e-baf5-1918719e7f7f>, ten sa

vloží do tela vygenerovaného emailu. Tým je možné priamo z emailovej správy potvrdiť prečítanie

doručeného oznámenia. Potvrdzovacia stránka `MailConfirm.aspx` si z parametra prečíta `GUID`

správy, ktorej prečítanie sa má potvrdiť. Následne je spustená uložená procedúra, ktorej úlohou je

vyhľadať v databáze email s daným `GUID`, ak je nájdený zmení sa jeho stav správy `Msg_Status` na

prečítaný, ten je reprezentovaný číselnou hodnotou 20. Potvrdzovacia stránka po úspešnom potvrdení

a modifikácii príznaku prečítania zobrazí poďakovanie za vykonané potvrdenie. V administračnej

časti informačného systému je možné zobraziť tabuľku `Msgs` so záznamami a v stĺpci stav vidieť,

ktorý príjemca potvrdil prečítanie správy. Administračné rozhranie popíšem v nasledujúcich

kapitolách.

### 4.2.3.2 Administračná stránka

K administračnému rozhraniu systému má prístup len užívateľ s rolou Administrátor. Administrácia zahŕňa všetky operácie nad tabuľkami systému, ktoré patria k administračnému rozhraniu. Administračná stránka je umiestnená v súbore DefaultAdministration.aspx. Na zobrazovanie tabuliek v jednotlivých administračných podstránkach je použitý serverový ovládací prvok GridView, ktorý je súčasťou .NET Framework od verzie 2.0.

#### Administračná stránka



Obr. 4.2.12 Ukážka rozhrania administračnej stránky

### 4.2.3.3 Administrácia užívateľov

Administrácia užívateľov zahŕňa možnosť zmeny údajov, pridávanie, alebo vymazanie zo systému. Jednou z dôležitých funkcií tejto administrácie je pridelovanie alebo odoberanie rolí, čiže záznamov z tabuľky UserRole. Ukážka administračného rozhrania vid' obr. 4.2.13.

#### Administrácia užívateľov

Užívateľské meno	Celé Meno	E-mail	Rola		
bajzcert	Tomáš Bajzecer	xbajze00@stud.fit.vutbr.cz	Lektor;		
dolezalz	Zdeněk Doležal	xdolez39@stud.fit.vutbr.cz	Lektor;		
lackoa	Andrej Lacko	andrej.lacko@post.sk	Študent;		
lackom	Martin Lacko	xlacko01@stud.fit.vutbr.cz	Administrátor; Lektor;		
lektor1	Skúšobný lektor 1	xlacko01@stud.fit.vutbr.cz	Lektor;		
lektor2	Skúšobný lektor 2	xlacko01@stud.fit.vutbr.cz	Lektor;		
malyp	Peter Malý	martin.lacko@umakov.sk	Študent;		

Užívateľské meno:

Heslo:   Nežadovať znovu heslo

Potvrdenie hesla:

Meno:

E-mail:

Rola:  Administrátor  Lektor  Študent

[Uložiť](#) [Storno](#)

[Späť na úvod administrácie](#)











Obr. 4.2.13 Ukážka administrácie užívateľov



#### 4.2.3.4 Administrácia miestností

Administrácia miestností obsahuje prvok GridView, ktorého dátový zdroj je naviazaný na získavanie dát z databázovej tabuľky Rooms. Každá miestnosť má atribút Capacity, čo je kapacita danej miestnosti. Táto kapacita sa používa napr. pri pridávaní rezervácie do rezervačného kalendára, podľa nej sa rozhodne či poskytuje dostatočnú kapacitu pre zvolený termín kurzu vid'. kapitola 4.2.3.1.

#### Administrácia miestností

Kód	Názov miestnosti	Kapacita		
UG1	Učebňa grafiky	25		
UKA	Učebňa kancelárskych aplikácií	40		
UP1	Učebňa programovacích jazykov 1	20		
UP2	Učebňa programovacích jazykov 2	30		

Kód:

Názov:

Kapacita:

[Uložíť](#) [Storno](#)

---















[Späť na úvod administrácie](#)

Obr. 4.2.14 Ukážka administrácie miestností

#### 4.2.3.5 Administrácia druhov kurzov

Administrácia druhov kurzov je umiestnená v súbore CourseTypesAdministration.aspx. Vykonávajú sa v nej operácie nad tabuľkou CourseTypes. Typy kurzov predstavujú kategórie do ktorých zaradzujeme novo pridávané kurzy v ponuke školiaceho strediska.

#### Administrácia druhov kurzov

Druh kurzu		
Databázové systémy		
Digitálne fotografie a video		
Internet		
Operačné systémy		
Počítačové siete		
Programovacie jazyky		

Druh kurzu:

[Uložíť](#) [Storno](#)

---

[Späť na úvod administrácie](#)

Obr. 4.2.15 Ukážka administrácie typov kurzov

#### 4.2.3.6 Administrácia kurzov

Administrácia kurzov je umiestnená v súbore CourseAdministration.aspx. Stránka slúži na správu kurzov, ktoré sa v mojom systéme ponúkajú. Pri vytváraní štruktúry, aké informácie má kurz v sebe obsahovať som sa inšpiroval u reálne na trhu pôsobiacich školiacich centier a stredísk. Každý kurz je samozrejme charakterizovaný svojim konkrétnym názvom a kódom kurzu, ktorý slúži najmä pre potreby strediska ako identifikátor daného kurzu a mal by byť pokiaľ to je možné jednoznačný. Pri tvorbe nového kurzu musíme z DropDownList-u vybrať kategóriu do ktorej kurz patrí. Ten je plnený z databázovej tabuľky CourseTypes uloženou procedúrou. Ďalšími dôležitými atribútmi sú informácie o cene, dĺžke trvania v dňoch a hodinách denne. Doplňujúcimi informáciami je syllabus, čo je osnova kurzu, aká je jeho logická stavba, ktoré konkrétne veci sú obsahom školenia. Nie každý z ponúkaných kurzov je vhodný aj pre zákazníkov – laikov prípadne začiatočníkov. Niektoré kurzy požadujú určité vstupné znalosti – prerekvizity, aby mal kurz pre zákazníka svoj význam a porozumel vykladanému učivu.

Každý z kurzov môžu vyučovať len lektori odborne na to spôsobilí s príslušnými znalosťami a certifikáciami. Teda kurz má svojich pridelených lektorov, ktorých priradíme zo zoznamu všetkých užívateľov školiaceho strediska vyhovujúcich podmienke:

- Užívateľ má pridelenú rolu Lektor.
- Nie je už v danom kurze ako lektor pridelený – neexistuje o ňom záznam v tabuľke

CoursesTeachers s identifikátorom kurzu s ktorým práve pracujeme.

V dolnej časti editačného okna som umiestnil dva listboxy. Listbox na pravej strane označený ako „Ostatní lektori“ je plnený, ako som už uviedol vyššie, všetkými lektormi školiaceho strediska splňujúcimi hore uvedené podmienky. Dáta z databázy získava uložená procedúra `COURSESGetLectorsNotForCourse`. Listbox na ľavej strane označený ako „Vybraní lektori“ obsahuje lektorov už pridelených pre daný kurz, dáta z databázy sa pre neho získavajú pomocou procedúry `COURSESGetLectorsForCourse`. Lektorov medzi týmito dvoma Listboxami „prehadzujeme“ pomocou šípiek. Ukážka administračného rozhrania vid' obr. 4.2.16.

Ku každému z kurzov môžeme vytvoriť sklad dokumentov, ktorý tvoria slajdy, prednášky a iné dokumenty potrebné k absolvovaniu kurzu. Pre vstup do skladu dokumentov použijeme v editačnom okne kurzu hypertextový odkaz „Dokumenty pre kurz“. Na Stránke DocsAdministration.aspx môžeme pridávať, prípadne zmazať už uložené dokumenty vid' obr. 4.2.17. K uloženiu dokumentu potrebujeme zadať typ dokumentu o aký sa jedná. Môže ísť napr. o učebnicu, prednáškové slajdy atď. Ďalej musíme cez tlačidlo Browse nájsť požadovaný súbor. Po kliknutí na Ulož je tento súbor uložený priamo do databázy. Neukladá sa žiadna relatívna cesta, kde je súbor umiestnený v súborovom systéme, ale jeho úložisko je priamo v databáze. Výhodou takého to riešenia je ochrana pred zmenou miesta uložených súborov, čím dôjde k že relatívne cesty k súborom sa stanú neplatnými. Tak je súbor dostupný kedykoľvek to je potrebné. Nevýhodou takého to riešenia

sú na druhej strane zvýšené nároky diskovú kapacitu pretože databáza sa bude s pribúdajúcim množstvom dokumentov zväčšovať. Avšak v dnešnej dobe kde kapacita úložných zariadení je pomerne vysoká a dostačujúca, nemusí byť táto nevýhoda podstatná.

Spôsob ukladania samotných súborov BLOB (Binary Large Object) [11] do databázy je riešený neštandardným spôsobom, ktorý bude popísaný v nasledujúcich riadkoch. Microsoft SQL Server k ukladaniu BLOB – ov ponúka dátové typy Binary a VarBinary(). Z vlastných skúseností z praxe z ukladania a čítania dát z/do týchto dátových typov v iných aplikáciách pracujúcich s databázou MS SQL, kedy dochádzalo k určitým problémom, som zvolil teda iný spôsob, ktorý je jedným z možných prístupov k ukladaniu binárnych dát v databáze. Najprv sa do databázovej tabuľky DocStore vložia údaje o ukladanom dokumente a to: DocStore\_CourseID – ku ktorému kurzu sa daný dokument viaže, DocStore\_ID je jednoznačný identifikátor – primárny kľúč vkladaneho dokumentu generovaný automaticky, DocStore\_Name obsahuje názov typu dokumentu a DocStore\_Path obsahuje názov samotného dokumentu s príponou, tak ako je uložený v súborovom systéme. Ďalším krokom je samotné uloženie v databáze. Princíp ukladania je v rozložení poľa bajtov – vkladaneho súboru na hexadecimálne reťazce o dĺžke 30 000 znakov, ktoré sa ukladajú do tabuľky Binaries. Počet hexadecimálnych reťazcov je závislý od veľkosti ukladaného súboru. Každý vkladany reťazec – záznam v tabuľke Binaries obsahuje ID vloženého záznamu Binaries\_ID, ďalej Binaries\_DocID čo je identifikátor – cudzí kľúč odkazujúci do tabuľky DocStore, ku ktorému uloženému záznamu patrí daný rozdelený hexadecimálny reťazec. Ďalším dôležitým údajom je PartOrder, ktorý reprezentuje poradové číslo vkladaneho hexadecimálneho reťazca, čo je kľúčové pre spätné poskladanie reťazcov a ich prekódovanie na pole bajtov, potom je súbor pripravený na stiahnutie užívateľom. Samotný reťazec je uložený v stĺpci Xdata, ktorý je dátového typu Text.

Zdrojový kód metódy rozkladajúcej pole bajtov na hexadecimálne reťazce:

```
/// <summary>
/// Rozlozi pole bajtov na hexadecimálne reťazce po 30 000 znakov -
/// v poslednom reťazci moze byt znakov menej
/// </summary>
/// <param name="bytes">Bajty pre rozsekanie a prevod</param>
/// <returns>Pole hexadecimálnych reťazcov</returns>
private string[] HexastringsForBinaries(byte[] bytes)
{
    List<string> strings = new List<string>();
    const int segmentLength = 15000; int index = 0;
    while (index < bytes.Length)
    {
        int max = Math.Min(index + segmentLength, bytes.Length);
        StringBuilder sb = new StringBuilder(max);
        for (int i = index; i < max; i++)
        {
            sb.Append(bytes[i].ToString("X2"));
        }
        strings.Add(sb.ToString());
        index += segmentLength;
    }
    return strings.ToArray();
}
```

## Administrácia kurzov

Názov kurzu	Kód kurzu	Poznámka	Typ kurzu	Kód kurzu	Trvá dni	Trvá hodín	Cena		
Adobe Photoshop	APH	Zpracování fotografií v prostředí profesionálního nástroje Adobe Photoshop	Digitálne fotografie a video	APH	3	24	5 000,00 Sk		
MS Exchange	EXCH	Kurz věnovaný správě poštovního serveru od instalace po konfiguraci a zabezpečení	Počítačové siete	EXCH	5	40	13 500,00 Sk		
MS Outlook	OTL	Obsluha elektronické pošty a elektronické kanceláře v prostředí MS Outlook	Internet	OTL	1	8	1 150,00 Sk		
MS SQL Server 2008	MSSQL	Konfigurácia a správa databází v prostredí MS SQL Serveru 2008	Databázové systémy	MSSQL	3	24	8 000,00 Sk		
Unix, Linux pre začiatočníkov	UNX	Seznámení s prací na počítačích se systémem Unix. Absolvent je schopen samostatně pracovat v prostředí systému Unix, používat základní příkazy, psát a opravovat textové soubory	Operačné systémy	UNX	3	24	3 900,00 Sk		
Základy programovania v jazyku Java	JVA	Popis jazyka java. Základy práce s API.. Praktická cvičení.	Programovacie jazyky	JVA	3	24	4 200,00 Sk		

Názov kurzu:

Kód kurzu:

Poznámka:

Druh kurzu:

Trvá dni:

Trvá hodín:

Cena:  Kč

Syllabus:

Požiadavky:

Vybraní lektori:

Ostatní lektori:

[Dokumenty pre kurz](#)

[Uložiť](#) [Storno](#)

Obr. 4.2.16 Ukážka administrácie kurzov

## Administrácia dokumentov pre kurz

Názov dokumentu	Cesta		
Učebnica	eng_ap1130ag_radius.pdf		

Názov dokumentu:

Cesta k dokumentu:

[Uložiť](#) [Storno](#)

[Späť na úvod administrácie](#) [Späť na administráciu kurzu](#)

Obr. 4.2.17 Administrácia dokumentov pre kurz

#### 4.2.3.7 Administrácia termínov kurzov

Administrácia termínov kurzov je umiestnená v súbore CourseTermsAdministration.aspx. V prvom poli vyberáme kurz, ktorého termín vytvárame. Termín je ďalej charakteristický počiatočným a záverečným dňom, zabudnúť nesmieme na kapacitu pretože každý termín hoci aj toho istého školenia môže mať rozdielnu kapacitu. Práve táto hodnota sa v rezervačnom kalendári porovnáva v kapacitou miestnosti, na základe tohto porovnania je lektorovi v danom okienku kalendára ponúknuté kurzy.

Termín kurzu má samozrejme svojich lektorov, jedného prípadne viacerých. Týchto lektorov k termínu priradíme podobným spôsobom, ako lektorov ku kurzom. S tým rozdielom, že k termínu priradíme lektorov, ktorý sú priradený ku kurzu, ktorého termín vytvárame. Údaje o priradených lektoroch sa naplňajú do tabuľky CourseTermsTeachers. Ukážka administračného rozhrania vid'. obr. 4.2.18.

Ku každému z termínov sa prihlasujú zákazníci. Každé prihlásenie sa považuje za záväznú objednávku. Tieto záznamy sa pridávajú do tabuľky Orders. Každá objednávka obsahuje príznak, či zákazník uhradil sumu za objednané školenie alebo nie. Tento príznak je ukladaný do stĺpca Order\_Paid typu bit, ktorý nadobúda hodnotu True, alebo False. K zoznamu prihlásených zákazníkov sa dostaneme kliknutím na hypertextový odkaz „Prihlásení študenti“ v editačnom okne termínu kurzu. V tejto obrazovke vidíme zákazníkov prihlásených v danom kurze a informáciu, či uhradili sumu za školenie. K dispozícii je tlačidlo, ktoré vykoná potvrdenie platby. Táto funkcionality by mali vykonávať len zodpovedné osoby k tomu určené. Vid'. obr. 4.2.19.

## Administrácia termínov kurzov

Názov kurzu	Počiatkový deň	Záverečný deň	Kapacita		
Adobe Photoshop	18. mája 2009	20. mája 2009	25		
Základy programovania v jazyku Java	18. mája 2009	20. mája 2009	20		
MS Exchange	19. mája 2009	22. mája 2009	15		
MS Outlook	21. mája 2009	25. mája 2009	30		
MS SQL Server 2008	22. mája 2009	25. mája 2009	15		
Unix, Linux pre začiatočníkov	29. mája 2009	2. júna 2009	30		

Kurz:

Počiatkový deň:

po	ut	st	št	pi	so	ne
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Záverečný deň:

po	ut	st	št	pi	so	ne
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Kapacita:

Vybrani lektori

Skúšobný lektor 2

Ostatní lektori

Tomáš Bajzecer

[Prihlásení študenti](#)

[Uložiť](#) [Storno](#)

[Späť na úvod administrácie](#)

Obr. 4.2.18 Ukážka administrácie termínov kurzov

## Administrácia prihlásených študentov

Priezvisko	Meno	Ulica	Mesto	PSC	Telefón	Email	Stav platby	Potvrdenie platby
Lacko	Andrej	Bertotovce 105	Bertotovce	082 35	082 35	andrej.lacko@post.sk	Nezaplatené	<input type="button" value="Potvrdiť platbu"/>

[Späť na úvod administrácie](#) [Späť na administráciu termínov kurzov](#)

Obr. 4.2.19 Obrazovka s prihlásenými študentmi

### 4.2.3.8 Administrácia odoslanej pošty

Administrácia odoslanej pošty poskytuje administrátorom pohľad na tabuľku Msgs, kde majú možnosť sledovať stav odoslaných emailov, či došlo k potvrdeniu prečítania, alebo nie. Odoslanie bolo podrobne spomenuté v závere kapitoly 4.2.3.1. Ukážka rozhrania vid'. obr. 4.2.20.



## Administrácia odoslanej pošty

Príjemca	Predmet	Stav	Dátum zmeny
andrej.lacko@post.sk	Zrušenie lekcie	Odoslané	14.5.2009 11:35:00
andrej.lacko@post.sk	Zrušenie lekcie	Odoslané	14.5.2009 11:42:00
martin.lacko@umakov.sk	Zrušenie lekcie	Odoslané	14.5.2009 11:42:00
andrej.lacko@post.sk	Zrušenie lekcie	Odoslané	14.5.2009 11:48:00
martin.lacko@umakov.sk	Zrušenie lekcie	Potvrdené prijatie	14.5.2009 11:54:00

[Späť na úvod administrácie](#)

Obr. 4.2.20 Rozhranie administrácie pošty

## 5 Záver

Cieľom tejto bakalárskej práce bolo vytvoriť informačný systém pre školiace stredisko. Pri jeho tvorbe som kládol dôraz najmä na jednoduché užívateľské rozhranie a funkčnosť celého systému, ktorý má šetriť čas zákazníkov a poskytnúť im prehľad o svojich objednaných školeniach a vyučovacích hodinách. Zamestnancom školiaceho strediska má uľahčiť správu kurzov a najmä sprehľadniť, zjednodušiť prácu s rezerváciami vyučovacích hodín a zefektívniť využitie kapacít miestností a učební, ktorými stredisko disponuje.

Prvá časť práce sa venovala predstaveniu použitých technológií, hlavne sa jednalo o platformu ASP.NET, kde som predstavil jej základné rysy, princíp fungovania a nové možnosti, ktoré ponúka. Keďže som s touto platformou neprišiel do kontaktu počas štúdia ani v praxi, bolo jej štúdium časovo dosť náročné. Jedná sa o prepracovanú, rozsiahlu, ale o veľmi dobre navrhnutú platformu, ktorá ponúka programátorovi mocný nástroj na budovanie aplikácií, no na to aby ňou bola si vyžaduje rozsiahle znalosti.

Druhá časť sa venovala samotnému návrhu aplikácie počnúc oboznámením sa s požiadavkami na systém a vytvoreniu určitej neformálnej špecifikácie pri ktorej som sa inšpiroval webovými stránkami a informačnými systémami školiacich stredísk na slovenskom, ale aj českom trhu. Na základe toho som vytvoril vhodné modely navrhovaného systému. Pri implementácii som použil viacvrstvový model architektúry aplikácie, ktorý oddeľuje vrstvu prístupu k dátam od vrstvy aplikačnej logiky.

Tento systém pokrýva však len časť problematiky a požiadavkov na školiace stredisko a nie je ani jeho cieľom nahradiť iné komerčné na trhu zabehnuté ekonomické a účtovné systémy. Práve možným prepojením môjho systému s inými externými systémami sa môže dosiahnuť zvýšenie funkcionality, zautomatizovanie určitých procesov v rámci firmy čo vedie k uspokojeniu potrieb zákazníka. Práve v tejto oblasti vidím ďalší námet na rozvoj systému. Konkrétne ide o prepojenie evidencie úhrad za objednané kurzy databázy môjho systému s databázou externého ekonomického systému a ich synchronizácie. Samozrejme ďalšie možnosti na rozvoj vidím aj v systéme, ktorý je výsledkom tejto práce. Jednalo by sa o rozšírenie existujúceho rezervačného systému a to smerom k zákazníkovi. Cieľom by bolo umožniť rezervácie do jednotlivých termínov samotným zákazníkom, kedy si môžu vybrať ten, ktorý im najviac časovo vyhovuje.

Pred začatím práce na tomto informačnom systéme som mal len nejasné vízie a predstavy, čo všetko obnáša návrh a implementácia takýchto systémov navyše s daným časovým plánom, takže z tejto práce si odnášam cenné skúsenosti nielen z programátorskej oblasti, ale aj to že nie je potrebné podceňovať potrebu dobrého návrhu a analýzy. Tieto cenné skúsenosti sú pre mňa krokom vpred a získané znalosti budú v budúcnosti zúročené.



# Literatúra

- [1] Lacko, L., *ASP.NET a ADO.NET 2.0 Hotová řešení*. Brno, Computer Press, 2006, 1. Vydanie, ISBN 80-251-1028-1.
- [2] The .NET Framework architecture [online]. 2004. Dostupný na WWW:  
[http://www.codeguru.com/dbfiles/get\\_image.php?id=8245&lbl=IMAGE1.JPG&ds=20040921](http://www.codeguru.com/dbfiles/get_image.php?id=8245&lbl=IMAGE1.JPG&ds=20040921)
- [3] Nagel, Ch., Evjen, B., Glynn, J., Watson, K., Skinner, M., Jones, A., *C# 2005 Programujeme profesionálne*. Brno, Computer Press, 2006, 1. Vydanie, ISBN 80-251-1181-4.
- [4] Evjen, B., Hanselman, S., Sivakumar, S., Rader, D., Muhammad, F. *ASP.NET 2.0 Programujeme profesionálne*. Brno, Computer Press, 2006, 1. Vydanie, ISBN 80-251-1286-1.
- [5] Microsoft Visual Studio [online]. Dostupný na WWW:  
[http://cs.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://cs.wikipedia.org/wiki/Microsoft_Visual_Studio)
- [6] Microsoft Visio [online]. Dostupný na WWW:  
<http://www.microsoft.com/cze/office/programs/visio/highlights.mspix>
- [7] Creating a Business Logic Layer [online]. Dostupný na WWW:  
<http://www.asp.net/learn/data-access/tutorial-02-cs.aspx>
- [8] Introducing SQL Server Management Studio [online]. Dostupný na WWW:  
<http://msdn.microsoft.com/en-us/library/ms174173.aspx>
- [9] Internet Information Services [online]. Dostupný na WWW:  
[http://en.wikipedia.org/wiki/Internet\\_Information\\_Services](http://en.wikipedia.org/wiki/Internet_Information_Services)
- [10] Guid.NewGuid structure [online]. Dostupný na WWW:  
<http://msdn.microsoft.com/en-us/library/system.guid.newguid.aspx>
- [11] Binary large object BLOB [online]. Dostupný na WWW:  
[http://en.wikipedia.org/wiki/Binary\\_large\\_object](http://en.wikipedia.org/wiki/Binary_large_object)

# Zoznam príloh

Príloha 1. CD