

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

System pro organizování školicích akcí a rekvalifikačních
kurzů



2013

Richard Nálepa

Anotace

Realizací tohoto projektu vznikne silně pomocný nástroj pro zadavatele, který ho bude využívat ke kompletní organizaci rekvalifikačních a doškolovacích kurzů. Aplikace bude vytvořena pro operační systém Windows a bude přesně zpracována dle požadavků administrativních pracovníků, kteří tuto aplikaci budou v praxi používat a během vývoje testovat. Chyby či nové návrhy budou konzultovány a dodatečně zapracovány do projektu tak, aby zadavatel byl s aplikací maximálně spokojen.

Děkuji vedoucímu práce Ing. Jiřímu Hronkovi za výborné vedení mé bakalářské práce a jeho věcné připomínky.

Obsah

1. Úvod	6
1.1. Zadání bakalářské práce	6
1.2. Důvod vzniku aplikace	6
1.3. Popis organizace kurzů a školení	6
1.4. Požadavky zadavatele na systém	7
2. Použitý hardware a software pro vývoj aplikace	8
2.1. Požadavky pro spuštění	8
3. Realizace aplikačního programu	9
4. Uživatelská část	10
4.1. Instalace	10
4.2. Odinstalace	10
4.3. První spuštění a ovládání programu	12
4.3.1. Tvorba rozvrhu	12
4.3.2. První spuštění	12
4.3.3. Ovládání programu	13
5. Programátorská část	27
5.1. Použité technologie	27
5.1.1. Platforma .NET	27
5.1.2. Technologie ADO.NET	27
5.1.3. MSSQL	28
5.2. USE CASE diagram aplikace	29
5.3. Databáze	30
5.3.1. Popis základních vazeb databáze	30
5.3.2. Seznam základních pohledů	31
5.3.3. ER diagram	31
5.4. Popis hlavních tříd a jejich metod	33
5.4.1. Třídy formulářů	33
5.4.2. Třídy	35
5.4.3. Tiskové třídy Crystal Report	35
5.4.4. Vybrané třídy a jejich metody	35
Závěr	42
Conclusions	43
Reference	44
A. Obsah příloženého CD	45

Seznam obrázků

1.	Instalace „Spuštění instalace“	11
2.	Odinstalace „Odstranění programu“	12
3.	První spuštění „Hlavní formulář“	13
4.	Ovládání programu „Nastavení počátečního datumu“	14
5.	Ovládání programu „Nastavení podle“	14
6.	Ovládání programu „Denní záznam“	15
7.	Ovládání programu „Záložky“	16
8.	Ovládání programu „Vytvoření nového předmětu“	16
9.	Ovládání programu „Nulová pozice“	16
10.	Ovládání programu „Vytvoření nového času“	17
11.	Ovládání programu „Vytvoření nového lektora“	18
12.	Ovládání programu „Přidání předmětu lektorovi“	18
13.	Ovládání programu „Vytvoření nového kurzu“	19
14.	Ovládání programu „Přidání předmětu kurzu“	19
15.	Ovládání programu „Přidání nového žáka“	20
16.	Ovládání programu „Přidání kurzu žákovi“	20
17.	Ovládání programu „Tvorba rozvrhu lektorům“	21
18.	Ovládání programu „Záložka vyhledávání“	22
19.	Ovládání programu „Vyhledání lektora v rozvrhu“	22
20.	Ovládání programu „Volby tisku“	23
21.	Ovládání programu „Rozvrh hodin žáka“	23
22.	Ovládání programu „Rozvrh hodin lektora“	24
23.	Ovládání programu „Archív žáků“	25
24.	Ovládání programu „Platby“	25
25.	Ovládání programu „Nová platba“	26
26.	Architektura ADO.NET	28
27.	USE CASE diagram aplikačního programu „Use Case diagram“	29
28.	ER model databáze „ER model“	32

1. Úvod

1.1. Zadání bakalářské práce

Úkolem bakalářské práce je vytvořit organizační nástroj dle potřeb zadavatele, který zajišťuje rekvalifikační kurzy akreditované Ministerstvem školství, mládeže a tělovýchovy ČR v oboru manikúry, pedikúry, nehtové modeláže. Dále organizuje i doškolovací kurzy v těchto oborech a postupně rozšiřuje svoji činnost na výuku péče o tělo komplexněji.

1.2. Důvod vzniku aplikace

Ve firmě zajišťující rekvalifikační, doškolovací a zájmové kurzy je zaměstnána moje přítelkyně, která dala hlavní impuls ke vzniku této aplikace. Její pracovní náplň spočívá v kompletní organizaci těchto kurzů. V minulosti byly veškeré záznamy vedeny papírovou formou, jakékoliv případné změny byly přelepovány, přepisovány, gumovány a poté opět ručně zpracovávány. Bylo tedy příhodné tuto činnost nějak zpřehlednit, zpřesnit, eliminovat lidské chyby a zajistit srozumitelnou databázi uchazečů, lektorů, nabízených kurzů, jejich úhrad apod. Jako provizorní řešení posloužil Microsoft Excel, který se ale postupně stal nepřehledným a neposkytoval potřebný komfort. Průběžně bylo nutné provádět úpravy či přidávat funkce, a tak jsem měl možnost se v této problematice podrobněji zorientovat a díky studiu na Univerzitě Palackého v Olomouci jsem začal mít představu o něčem specifitějším a začal pracovat na aplikaci MP_Kurzy.

1.3. Popis organizace kurzů a školení

- Školící středisko funguje již 15 let a ročně ho navštíví přibližně 150 uchazečů. Z toho dvě třetiny navštěvují rekvalifikační kurzy. Jednu třetinu tvoří zájemci o doškolovací a zájmové kurzy. Doškolovací kurzy jsou pořádány pro absolventy rekvalifikačních kurzů, kteří mají zájem specializovat se ve svém oboru. Zájmové kurzy jsou určeny i pro širokou veřejnost.
- V rámci rekvalifikačních kurzů, které zadavatel pořádá, jsou přísné požadavky na organizaci. Uchazeči jsou povinni absolvovat vyučovací předměty, jako je například psychologie, dermatologie, anatomie a mnohé další dle svého oboru. Množství vyučovacích hodin je rovněž rozdílné dle oboru a také s ohledem na dosavadní vzdělání absolventa.
- Rekvalifikační kurzy jsou zahajovány několikrát ročně a trvají 2 - 3 měsíce, s ohledem na množství přihlášených uchazečů a na frekvenci jejich docházky.

- Termín zahájení kurzu není pro všechny absolventy shodný a také termín ukončení se může u některých absolventů lišit.
- Vzhledem k odlišným nárokům na učební program jednotlivých oborů a s přihlédnutím na soukromé požadavky absolventů (pracovní povinnosti, rodina aj.) je potřeba rozvrh plánovat individuálně a zároveň mít možnost provádět změny zpětně.
- Školící středisko má k dispozici jednu učebnu s kapacitou třiceti míst. Tato místa se ale různě využívají, podle konkrétního předmětu (nárok na prostor pro nácvik praxe apod.).
- Absolventi ukončují kurz závěrečnou zkouškou a na základě této zkoušky získají osvědčení o rekvalifikaci, které je po splnění dalších zákonných podmínek opravňuje v daném oboru podnikat.

1.4. Požadavky zadavatele na systém

- Dodržovat tři vyučovací časy v jednom dni, možnost individuálního naplánování na každý den.
- Vytvářet vlastní databáze lektorů s možností editace.
- Vytvářet vlastní databáze předmětů s možností editace.
- Vytvářet vlastní databáze kurzů s možností editace.
- Vytvářet vlastní databáze žáků s možností editace.
- Možnost naplánovat a přehledně zobrazit všechny učební předměty a doškolovací akce v konkrétních časech a termínech, s přiřazenými lektory.
- Možnost zapisovat poznámky ke konkrétnímu vyučovacímu času a datu.
- Současně zaznamenávat a zobrazovat optimální naplnění učebny, aby nedošlo k překročení kapacity k danému datu a času při dynamickém plánování uchazečů na jednotlivé dny a vyučovací časy.
- Vytvářet učební plán pro jednotlivého lektora + export pro tisk.
- Vytvářet učební plán pro jednotlivého uchazeče + export pro tisk.
- Export pro tisk individuálně zvolených parametrů.
- Upozorňovat na státní svátky při plánování.

2. Použitý hardware a software pro vývoj aplikace

- Microsoft Windows XP Professional 32-bit, verze 2002 Service Pack 3
 - Celý vývoj aplikace byl programován na výše uvedené verzi operačního systému.
 - Počítač: Itell(R) Core(TM) Duo CPU T8100, 2.1GHz, 3Gb RAM
- Microsoft Visual Studio 2010 Version 10.0.30319.1 RTMRel SP, včetně Microsoft .NET Framework Version 10.0.30319.1 RTMRel
 - Zdrojový kód byl napsán prostřednictvím jazyka C#, což je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý společností Microsoft, určený převážně k programování aplikačních programů pro operační systémy Microsoft.
- MSSQL
 - K ukládání dat byla použita souborová databáze MSSQL.
- Uživatelská nápověda dostupná přímo v aplikačním programu byla napsána prostřednictvím programu HelpNDoc verze 2.9. Jedná se o volně dostupný software.

2.1. Požadavky pro spuštění

Aplikace byla testována na těchto konfiguracích:

- Microsoft Windows XP Professional 32-bit, verze 2002 Service Pack 3
- Počítač: Itell(R) Celeron(R) CPU 2.53GHz, 0.99Gb RAM

3. Realizace aplikačního programu

Po několika konzultacích se zadavatelem a seznámení s problematikou byl vytvořen podrobný USE CASE model a podle něj pak struktura databáze, která byla vytvořena v Microsoft SQL management studiu. Veškeré tabulky byly vytvořeny pomocí SQL příkazů, poněvadž v době vzniku databáze jsem neměl dostatečné znalosti s vytvářením diagramů pomocí Microsoft Visual Studia. Později byla databáze importována do projektu a byl vytvořen diagram pomocí grafického návrháře. Postupně, jak se projekt vyvíjel, vznikaly nové pohledy, které již byly vytvářeny přímo ve Visual Studiu. Zadavatel průběžně prováděl kontrolu všech naprogramovaných částí a následně proběhla fáze zkoušení v praxi.

Napřed byly naprogramovány jednotlivé formuláře pro zadávání údajů do databáze a následně formuláře pro editaci jednotlivých položek a s tím související kontroly a validace vstupních údajů. Některé z nich jsou pak podle regulárních výrazů předány konstruktoru třídy `Regex` ze jmenného prostoru `System.Text.RegularExpresion`. Následně je metodou `Match` této třídy vytvořena instance třídy `System.Text.RegularExpresion.Match` a vlastností `Success` lze zjistit, zda řetězec odpovídá podmínkám kontrolního regulárního výrazu.

Další fáze zahrnovala návržení hlavního formuláře. Ten byl po několika verzích přepracován do výsledné podoby. Nynější vzhled formuláře je výsledkem požadavků zadavatele, zobrazit současně veškeré potřebné informace. Vše s minimálním přepínáním mezi formuláři při tvorbě rozvrhů, a to i na úkor velkého množství komponent na jednom formuláři. Zásadní bylo vytvoření funkce, která plynule přepisuje dynamicky všechny komponenty hlavního formuláře, kterých je poměrně velké množství, například na základě události `OnMouseWheel()`. Bylo nutné provést časovou optimalizaci, poněvadž na hardwarově slabších počítačích bylo přepisování pomalé.

V další fázi jsem vytvořil rychlé a pohodlné vyhledávání na základě různých parametrů, které se dají navolit a volby uložit do souboru, aby při otevření aplikace zůstaly nastaveny.

V poslední fázi jsem vytvořil formuláře tiskových sestav a databázi plateb, které byly navrženy zadavatelem. Hlavním kritériem byl co nejjednodušší a ekonomicky nenáročný tisk.

Od samého vzniku byla aplikace testována v praxi a díky každodenní komunikaci s administrativní pracovníci byly počáteční chyby rychle eliminovány a nové požadavky pohotově řešeny. Aplikace již nyní uživateli plně nahrazuje veškeré dříve používané ruční metody a nabízí mnoho dalších funkcí, které se dají považovat za nadstandardní vzhledem k celé původní myšlence vzniku této aplikace. Naprogramovaná aplikace byla testována na počítačích s operačními systémy:

- Microsoft Windows XP Professional SP3 (32-bit)
- Microsoft Windows 7 Home Premium SP1 (32-bit)

U Microsoft Windows 7 Home Premium SP1 (32-bit) bylo zjištěno rozdílné chování komponenty MonthCalendar ve spolupráci s rollerem myši. U Microsoft Windows XP kliknutím na komponentu a následným pohybem rolleru myši reaguje aplikace standardně a obslouží událost OnMouseWheel(MouseEventArgs e) tohoto pohybu. U Microsoft Windows 7 je následný pohyb rolleru po kliknutí na komponentu MonthCalendar obsloužen jako pohyb v této komponentě, a to tak, že se posouvá v kalendáři po měsících. Jestliže chceme ve Windows 7 docílit stejného efektu při pohybu kolečka myši (tj. pohybu po jednom dni v hlavním formuláři), musíme kliknout na jinou komponentu, než je komponenta MonthCalendar.

4. Uživatelská část

4.1. Instalace

Instalace zahrnuje nainstalování aplikace a nainstalování objektové knihovny „Crystal Reports Basic Runtime for Visual Studio 2008“, která zajišťuje plnou funkčnost tiskových sestav.

Postup instalace:

- a) Z instalačního CD spustíme soubor setup.exe, který je umístěn v adresáři \MPKurzyInstal\ MPKurzyInstal\Release.
- b) Otevře se výchozí okno instalace, viz obr. 1.



Obrázek 1. Instalace „Spuštění instalace“

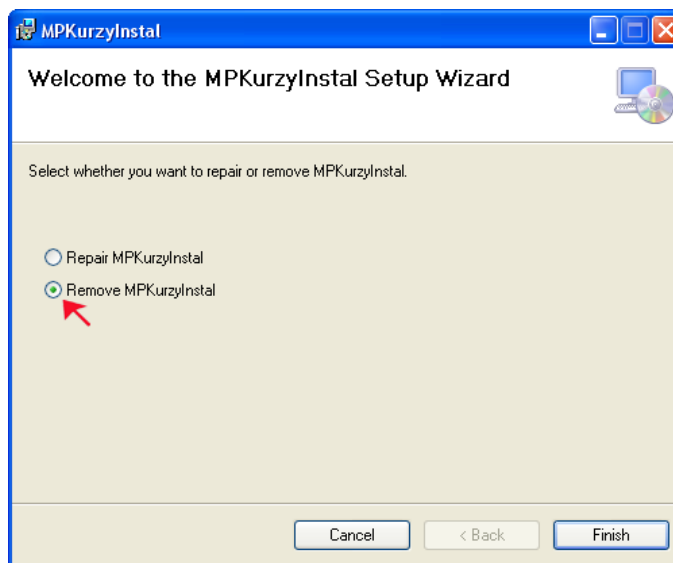
c) Pro nainstalování klikneme na tlačítko „Next“. Po otevření následujícího okna instalace nastavíme uživatele, cestu k uložení instalace a potvrdíme tlačítkem „Next“. V dalším okně potvrdíme navolené parametry opětovným kliknutím na tlačítko „Next“ a vyčkáme, než bude instalace dokončena. Po dokončení instalace klikneme na tlačítko „Close“.

d) Po dokončení instalace se na ploše vytvoří ikona s názvem MP_Kurzy, kterou lze nainstalovanou aplikaci spustit.

e) V případě, že na počítači není nainstalované prostředí Microsoft Visual Studio 2008 Professional SP a vyšší, je nutné nainstalovat balíček objektových knihoven „Crystal Reports Basic Runtime for Visual Studio 2008“, který je přiložen na instalačním CD v adresáři CRep.

4.2. Odinstalace

Odinstalací dojde k odebrání aplikačního programu včetně všech jeho součástí. Provedeme ji kliknutím na nabídku Start/Ovládací panely/Přidat nebo Odebrat programy. Odinstalaci lze také provést opětovným spuštěním instalačního souboru, výběrem „Remove MPKurzyInstal“ a dále kliknutím na tlačítko „Finish“, viz obr. 2.



Obrázek 2. Odinstalace „Odstranění programu“

4.3. První spuštění a ovládání programu

4.3.1. Tvorba rozvrhu

Tvorba rozvrhu je velice organizačně složitá činnost, která není možná bez lidského činitele. Je potřeba zohlednit časové požadavky lektorů a současně i uchazečů. Důležitý je rovněž celkový počet uchazečů, podle kterého se musí administrativní pracovník rozhodnout, kolikrát je nutné předměty naplánovat opakovaně, tak, aby jej všichni uchazeči zvládli absolvovat, a to s přihlédnutím na ekonomickou náročnost provozu (vytápění a osvětlení učebny, nasmlouvání lektorů a úklidové firmy). I po naplánování rozvrhu často dochází k nečekaným událostem (např. absence některých uchazečů z důvodu nemoci), kvůli kterým je třeba provádět změny také zpětně. Při optimálním uživatelském nastavení MP_Kurzy je program schopen dohlížet a upozorňovat na množství situací, které významně pomáhají administrativnímu pracovníkovi. Jedná se zejména o správné přiřazení předmětů ke kurzu, počtu kreditů k předmětu a určení koeficientu naplnění k předmětu. Program pak spolehlivě zobrazuje veškeré informace na hlavní straně, znázorňující diář, který byl vždy hlavním pracovním nástrojem administrativní pracovnice, a proto i program MP_Kurzy akceptuje zvyklou koncepci.

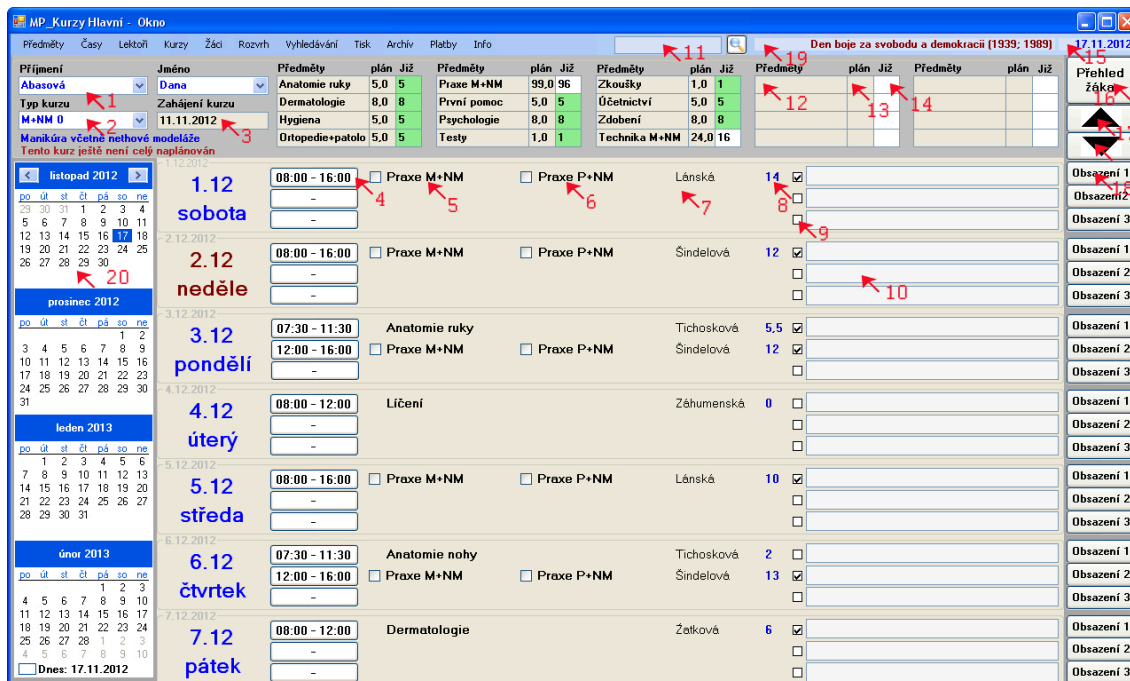
4.3.2. První spuštění

Vzhledem k tomu, že aplikace je už nějakou dobu používána v praxi, je databáze aplikace naplněna daty. Jména žáků a lektorů byla pozměněna z důvodu ochrany osobních údajů. Po otevření aplikace se otevře „Hlavní formulář“, který zachycuje veškeré potřebné informace, viz obr. 3.

4.3.3. Ovládání programu

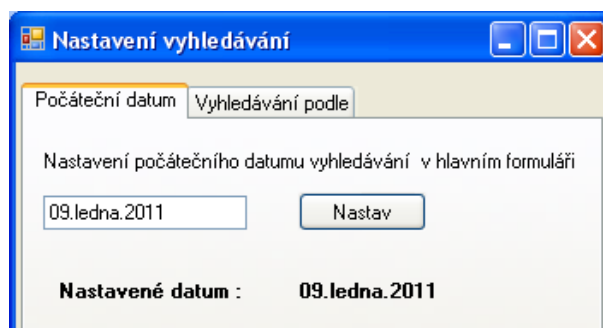
Aktivní dynamický formulář je rozdělen na tři části

- Horní část obsahuje informace o navoleném žákovi v ComboBoxu, viz popis č. 1 na obrázku č. 1 „Hlavní formulář“ (dále už jen číslo popisu), a o jeho aktuálním stavu naplánovaných hodin aktivně navoleného kurzu v ComboBoxu č. 2 včetně informace o celém názvu navoleného kurzu a pod ním, zda je tento kurz celý naplánován či nikoli. Pole č. 12, č. 13, č. 14 obsahují informace o předmětech a jejich počtu hodin, které musí žák absolvovat, a hodiny již naplánované. Kliknutím na tlačítko č. 16 „Přehled žáka“ se nám otevře sestava s informacemi o naplánovaných dnech k aktuálně navolenému kurzu v ComboBoxu č. 2. Sestava obsahuje datum, začátek vyučování, konec vyučování, počet kreditů a předmět, který bude v tento den vyučován. Sestavu je možné vytisknout. Pole č. 15 znázorňuje aktuální datum, č. 19 pak případný státní svátek k tomuto dni, nebo ke dni navolenému v kalendáři č. 20. Svátek se nemění, listujeme-li v kalendáři pomocí rolleru na myši.

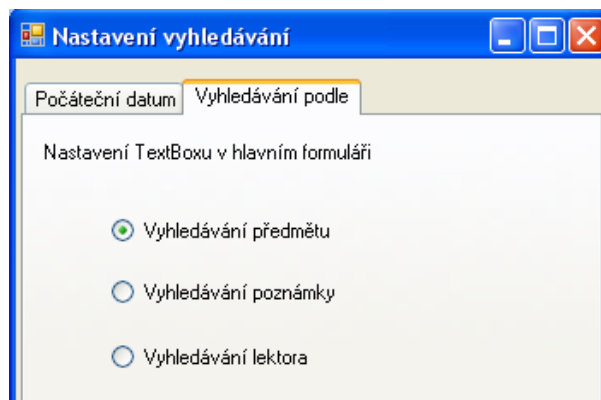


Obrázek 3. První spuštění „Hlavní formulář“

Pole č. 11 je určeno k vyhledávání. Úzce souvisí s ikonou lupy a se záložkou Vyhledávání/Nastavení vyhledávání, kde je možné si nastavit počáteční datum vyhledávání, viz obr. 4. „Nastavení počátečního datumu“, nebo položku, kterou chceme vyhledat, viz obr. 5. Nastavené datum je parametr, kterým zamezíme vyhledávání ve starších ročnících, které jsou uloženy v databázi a při vyhledávání by zbytečně prodlužovaly vyhledávání. To probíhá tak, že při opakovaném kliknutí na ikonu lupy cyklicky prochází záznamy od navoleného data až po poslední položku.



Obrázek 4. Ovládání programu „Nastavení počátečního datumu“



Obrázek 5. Ovládání programu „Nastavení podle“

- Spodní část je kalendář/diář, ve kterém je zachycen týdenní rozvrh hodin. Listovat v něm můžeme po jednom dni pomocí rolleru na myši, nebo po týdnech šipkami č. 17, nebo kliknutím na komponentu čtyřměsíčního kalendáře umístěnou na levé straně formuláře č. 20.

Popis denního záznamu kalendáře, viz obr. 6.



Obrázek 6. Ovládání programu „Denní záznam“

1. Datum, které se zobrazuje jen tehdy, je-li v tento den nějaký záznam v databázi rozvrhu.
2. Datum a den v kalendáři.
3. Tlačítka volby. Tímto tlačítkem provedeme záznam navolenému žákovi v ComboBoxu č. 1. Popis na tlačítku znázorňuje vyučovací čas. Jestliže je na tlačítku pomlčka či prázdný znak, znamená to, že v tento časový úsek neprobíhá v učebně výuka.
4. První vyučovaný předmět v daný vyučovací čas. V jednom vyučovacím čase mohou být vyučovány dva předměty. Většinou se jedná o praktické předměty, kdy se v učebně vyskytují žáci různých kurzů. Lektor dohlíží nad oběma předměty. Každý předmět může mít jiný počet naplnění, např. praxe pedikúry obsadí tři místa učebny, praxe manikúry dvě místa učebny.
5. Druhý vyučovaný předmět v daný vyučovací čas.

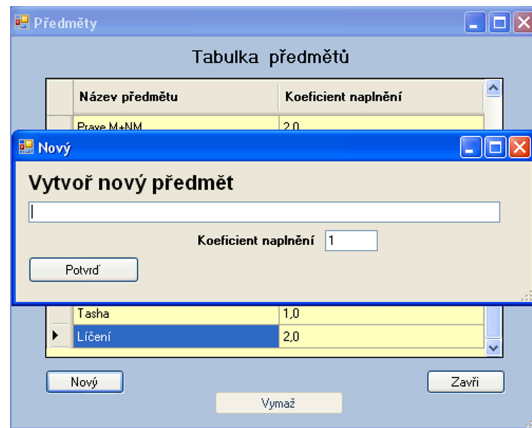
6. Lektor, který v tento den a čas vyučuje.
 7. Naplnění třídy. Číslo, které nám říká, kolik míst je v tento vyučovací čas obsazených. Kapacita místnosti je kontrolována a při překročení uživatele upozorní změnou barvy. Informaci, kolik míst je třeba pro výuku předmětu, si nese s sebou každý předmět v databázi, v tabulce předmětu. Hodnotu je možno kdykoli změnit.
 8. CheckBox nás informuje, že aktivní žák, navolený v ComboBoxu č. 1, je v tento den zapsán na vyučování. Klinutím na označený CheckBox můžeme tento záznam žákovi odstranit.
 9. Pole pro poznámku. Poznámku je možné psát kdykoli. Uložení se provede překliknutím na jiné políčko.
- Záložky v horní liště formuláře
Popis záložek, viz obr. 7.



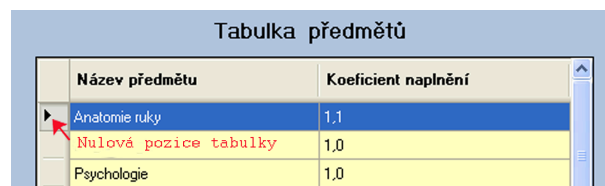
Obrázek 7. Ovládání programu „Záložky“

- Předměty: V záložce Předměty je možné pomocí tlačítka „Nový“ vytvářet nové předměty. Později budou přiřazovány kurzům a lektorům, kteří tento předmět vyučují. Koeficient naplnění v pravé části formuláře „Nový“ je číslo, udávající počet míst v učebně, které připadne na jednu osobu. Například teorie zabere jedno místo/osoba, praxe manikúra dvě místa/osoba. Pole koeficient je zabezpečeno proti špatně zadaným hodnotám. Zadáme-li nepřipustnou hodnotu, pak po její validaci je systémem nastavena na hodnotu 1. V případě, že chceme hodnotu koeficientu nebo název předmětu změnit, je možné to provést i později dvojklikem na buňku, kterou chceme editovat. Chceme-li řádek vymazat, musíme kliknout na nulovou pozici¹ tabulky tohoto předmětu, čímž aktivujeme tlačítko „Vymaž“, kterým označený záznam můžeme vymazat, viz obr. 9. Je-li předmět již používán, tj. jsou-li na něj vazby kurzů, nemůže být odstraněn.
- Časy: V záložce Časy je možné pomocí tlačítka „Nový“ vytvářet nové časy. Důvod vzniku této tabulky je rychlejší zadávání při tvorbě rozvrhu. V ComboBoxech můžeme zadat předvolené hodiny začátku a konce vyučování a taky jeden ze tří typů časů, viz obr. 10. Tři typy časů proto, že vyučovací

¹Nulová pozice se bude v textu vyskytovat častěji. Příklad je znázorněn na tabulce předmětů, viz obr. 9.



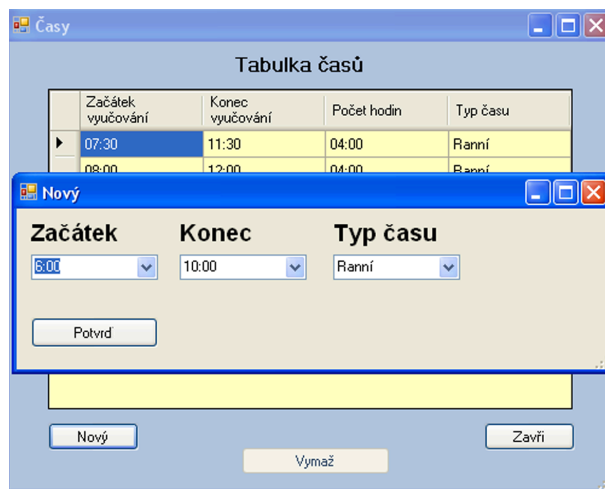
Obrázek 8. Ovládání programu „Vytvoření nového předmětu“



Obrázek 9. Ovládání programu „Nulová pozice“

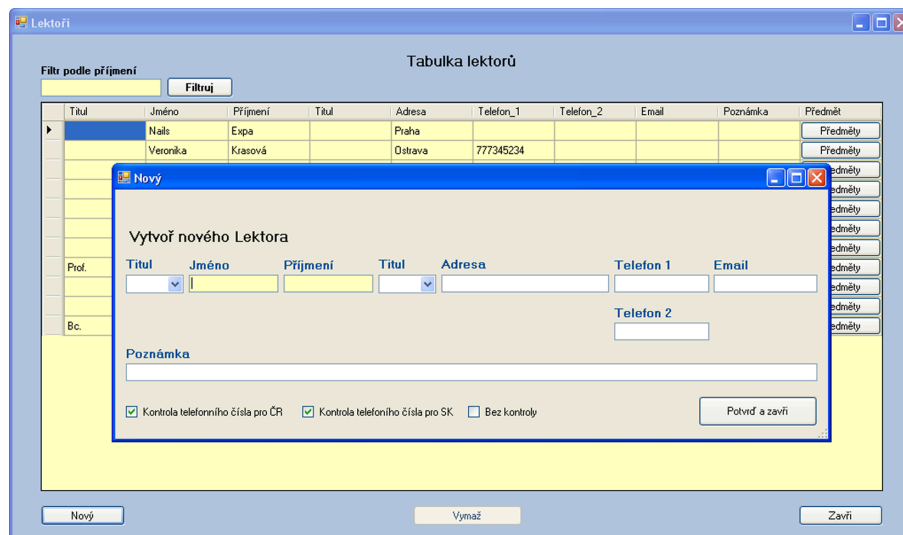
den je možné rozdělit do tří úseků. Předem zapsané časy se pak zobrazují v těchto úsecích ve formuláři „Rozvrh“. Potvrzením se záznam uloží do databáze tabulky časů. Sloupec „Počet hodin“ se vypočítává ze zadaných časů a nelze ho editovat. Ostatní buňky lze měnit stejným způsobem jako u ostatních tabulek, a to pomocí dvojkliku. Vymazání je možné kliknutím na nulovou pozici tabulky, čímž aktivujeme tlačítko „Vymaž“, kterým označený záznam můžeme vymazat. Je-li čas používán, tj. jsou-li na něho vazby v rozvrhu, nemůže být odstraněn.

- Lektoři: V záložce lektoři je možné pomocí tlačítka „Nový“ vytvářet nové lektory. Žlutá políčka jsou povinná. Databázi lektorů můžeme editovat a mazat stejným způsobem jako ostatní tabulky. V posledním řádku každého lektora se nachází tlačítko „Předměty“. Tímto tlačítkem otevřeme další okno, ve kterém můžeme zapsat lektorovi jeho vyučovací předměty, viz obr. 12. Přidání předem navoleného předmětu v ComboBoxu provedeme pomocí tlačítka „Přidej předmět“. Odebrání provedeme kliknutím na tlačítko „Předmět“ a dále kliknutím na nulovou pozici tlačítko „Odeber předmět“ aktivujeme. Aktuální předměty se okamžitě znázorňují v tabulce tohoto formuláře. Formulář lektorů obsahuje textové pole pro výběr lektora podle

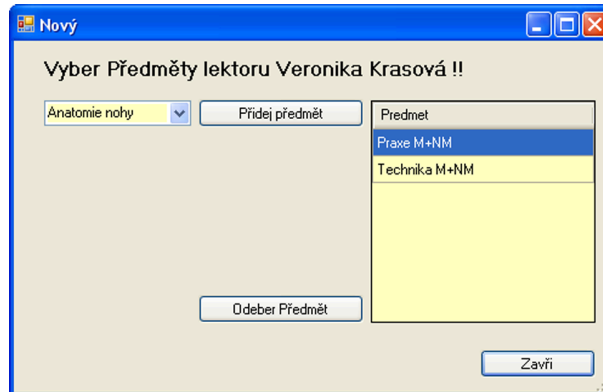


Obrázek 10. Ovládání programu „Vytvoření nového času“

příjmení. Filtr hledá shodu podle posloupnosti písmen do něj zadaných a po filtraci se pole nastaví na prázdný znak. Kliknutím na tlačítko „Filtr“ s prázdným znakem se provede načtení všech lektorů z databáze.

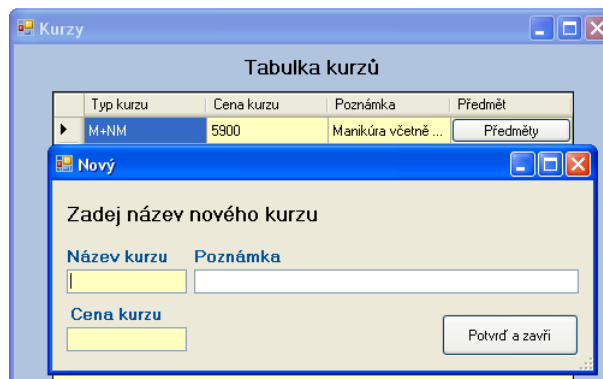


Obrázek 11. Ovládání programu „Vytvoření nového lektora“

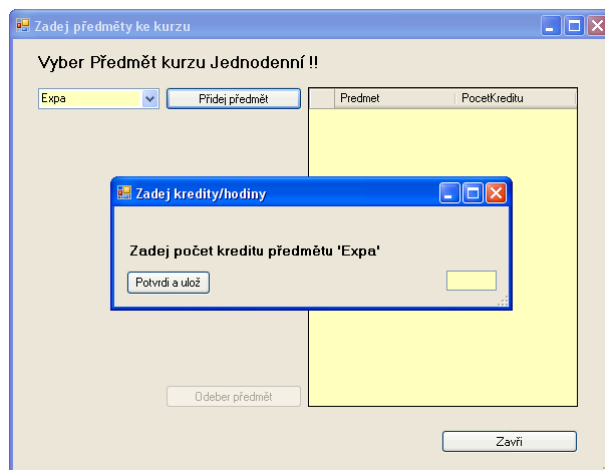


Obrázek 12. Ovládání programu „Přidání předmětu lektorovi“

- Kurzy: Akreditované kurzy mají ministerstvem školství pevně stanovený počet vyučovacích hodin jednotlivých předmětů. Pro vytvoření slouží záložka kurzy, ve které je možné pomocí tlačítka „Nový“ kurzy nebo školicí akce vytvářet, viz obr. 13. Později se tyto přednastavené kurzy zadávají žákům, kteří mají zájem v tomto oboru rozšířit své znalosti. Žlutá políčka jsou povinná. Potvrzením se kurz uloží do databáze tabulky kurzů. Databázi kurzů můžeme editovat a mazat stejným způsobem jako ostatní tabulky. V posledním řádku každého kurzu se nachází tlačítko „Předměty“. Tímto tlačítkem otevřeme další okno, ve kterém můžeme kurzu přiřadit předměty, které obsahuje, viz obr. 14. Přidání předmětu v ComboBoxu provedeme pomocí tlačítka „Přidej předmět“. Po přidání vyskočí okno, do kterého je nutno zadat počet kreditů předmětu. Hodnota musí být v rozsahu 1 - 200. Odebrání předmětu provedeme tlačítkem „Odeber předmět“.

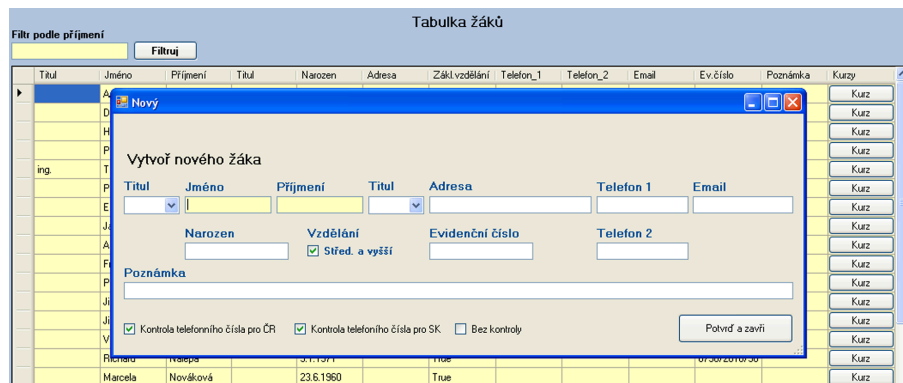


Obrázek 13. Ovládání programu „Vytvoření nového kurzu“

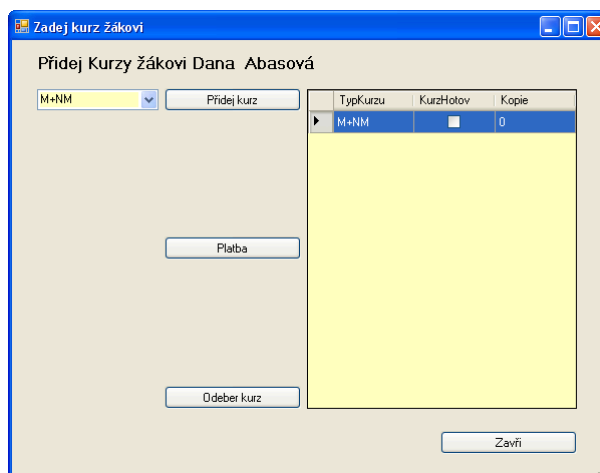


Obrázek 14. Ovládání programu „Přidání předmětu kurzu“

- **Žáci:** V záložce Žáci je možné pomocí tlačítka „Nový“ vytvářet nové uchazeče o kurzy a doškolovací akce. Zadávání je obdobné jako u lektorů, viz obr. 15. Žlutá políčka jsou povinná. Databázi žáků můžeme editovat a mazat stejným způsobem jako ostatní tabulky. V posledním řádku každého lektora se nachází tlačítko „Kurz“. Tímto tlačítkem otevřeme další okno, ve kterém můžeme zapsat žákovi požadovaný kurz či školení, viz obr. 16. Přidání předem navoleného kurzu v ComboBoxu provedeme pomocí tlačítka „Přidej kurz“. Navíc se tím otevře formulář „Platby“, poněvadž musí proběhnout zálohová platba. Editovat platby můžeme pomocí tlačítka „Platby“. Odebrání provedeme označením kurzu kliknutím na nulovou pozici. Tím aktivujeme tlačítko „Odeber kurz“, kterým předmět odebereme. Aktuální kurzy se okamžitě znázorňují v tabulce tohoto formuláře. Sloupec „Kurz hotov“ nás informuje o tom, zda je tento kurz celý naplánován a přesunut do archívu. Formulář žáků obsahuje textové pole pro výběr žáka podle příjmení. Filtř hledá shodu podle posloupnosti písmen do něj zadaných a po filtraci se pole nastaví na prázdný znak. Kliknutím na tlačítko „Filtř“ s prázdným znakem se provede načtení všech žáků z databáze. **Žáky, kteří již navštěvovali nějaký kurz, nelze odstranit z databáze.**
- **Rozvrh:** V záložce Rozvrh je možné pomocí tlačítka „Nový“ vytvářet rozvrh hodin lektorům. S lektory jsou předem dohodnuty jejich časové požadavky, které je možno zapsat do poznámek v tabulce lektorů. Rozvrh se tvoří podle celkového počtu přihlášených uchazečů a současně s ohledem na jejich časové možnosti. V průběhu kurzu je nutné rozvrh operativně upravovat. Předpokládá se, že se rozvrh tvoří po časových úsecích jdoucích po sobě, proto pro urychlení vytváření rozvrhu se po otevření formuláře „Nový“ datum automaticky nastaví na poslední datum v databázi tabulky



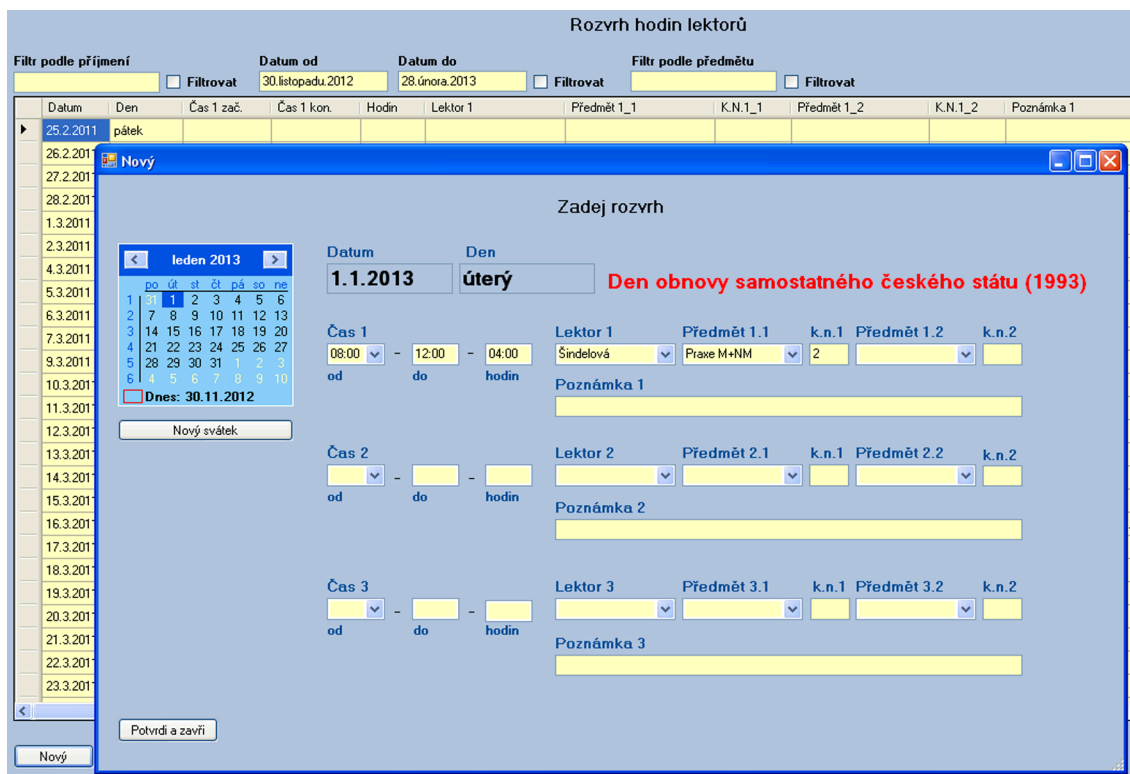
Obrázek 15. Ovládání programu „Přidání nového žáka“



Obrázek 16. Ovládání programu „Přidání kurzu žákovi“

„Rozvrh“. Pro změnu tohoto data slouží komponenta kalendář. Jestliže navolíme nějaké datum, které obsahuje nějaký státní svátek, zobrazí se tato informace červeně v pravé části formuláře vedle TextBoxu dne. Tím upozorňuje uživatele při plánování na dny, které mají nějakým způsobem vliv na plánování výuky. Uživatel má možnost vytvořit si vlastní svátek či poznámku do databáze kliknutím na tlačítko „Nový svátek“. Potvrzením se záznam uloží do tabulky rozvrhu a po zavření formuláře „Rozvrh“, se záznam přenesou do Hlavního formuláře a je možno jej používat při plánování rozvrhu jednotlivým žákům.

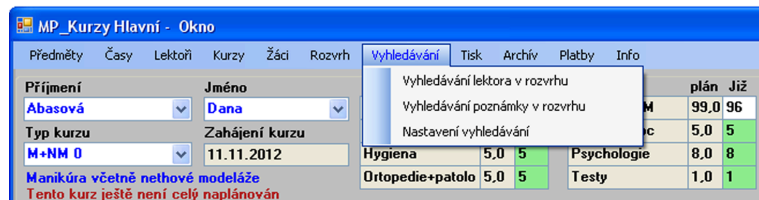
- Vyhledávání: V záložce Vyhledávání je možno vybrat jednu ze tří voleb, viz obr. 18.
 - **Vyhledávání lektora v rozvrhu** - Touto volbou otevřeme malé ne-



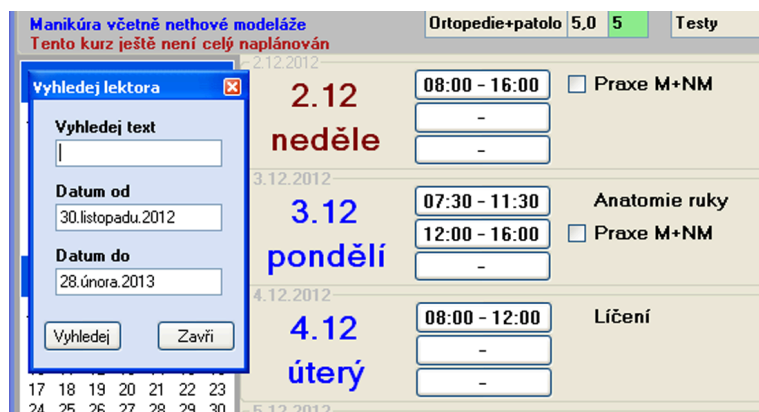
Obrázek 17. Ovládání programu „Tvorba rozvrhu lektorům“

modální okno, v němž můžeme zadat období, ve kterém se zadaná posloupnost znaků vyhledá, viz obr. 19. Opakovaným kliknutím na tlačítko „Vyhledej“ procházíme všechny záznamy. **Nalezený záznam se vždy zobrazí na první pozici denního kalendáře.**

- **Vyhledávání poznámky v rozvrhu** - Touto volbou otevřeme stejné nemoďální okno, jako při vyhledávání lektora v rozvrhu, které ovšem vyhledává jen poznámky v rozvrhu. Opakovaným kliknutím na tlačítko „Vyhledej“ procházíme všechny záznamy. **Nalezený záznam se vždy zobrazí na první pozici denního kalendáře.**
- **Nastavení vyhledávání** - V tomto okně je možné nastavit počáteční datum vyhledávání. Tím eliminujeme vyhledávání v předešlých letech. Nastavené datum se ukládá do pomocného souboru stanice.txt. V záložce „Vyhledání podle“ můžeme nastavit vyhledávání podle předmětu, poznámky nebo lektora. Nastavení se ukládá do pomocného souboru vyhledat.txt. **Všechna tato nastavení se týkají textového pole v hlavním formuláři určeného pro vyhledávání.** Samotné vyhledávání se pak provede kliknutím na ikonu s lupou.

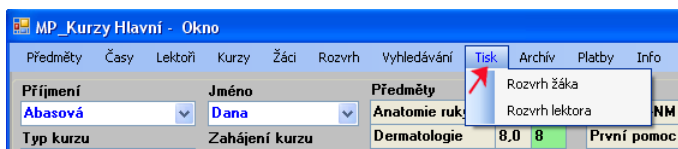


Obrázek 18. Ovládání programu „Záložka vyhledávání“



Obrázek 19. Ovládání programu „Vyhledání lektora v rozvrhu“

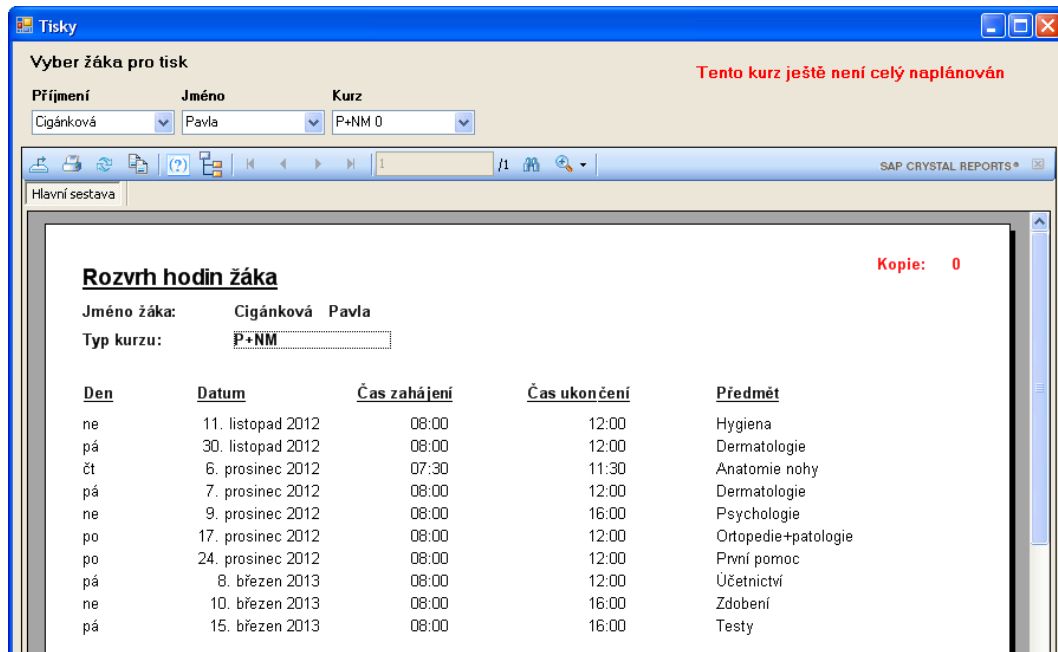
- Tisk: V záložce Tisk je možno vybrat jednu ze dvou voleb, viz obr. 20.



Obrázek 20. Ovládání programu „Volby tisku“

- **Rozvrh hodin žáka** je rozpis naplánovaných předmětů, které musí žák absolvovat. Kliknutím na tuto položku se otevře formulář „Tisky žáků“, který obsahuje ComboBox „Příjmení“, ve kterém si navolíme žáka, čímž načteme jeho kurzy. Kurz žáka, který chceme vytisknout, vybereme v ComboBoxu „Kurz“ a ten se zobrazí v sestavě pro tisk. V ComboBoxu se zobrazují i žáci, kteří **nemají** naplánovaný celý kurz, poněvadž zpravidla dostávají čtrnáctidenní dílčí učební plán. Informace, zdali je kurz zcela naplánován, se zobrazuje červeně v pravém horním rohu. Tisková sestava obsahuje informaci „Kopie“, toto číslo udává, pokolikáté žák navštívuje stejný kurz. V praxi se to stává vět-

šinou u jednodenních akcí či školení. Akreditované kurzy není třeba navštěvovat vícekrát, jen pokud by některý žák tento kurz v minulosti nedokončil a chtěl by jej navštěvovat znovu. Sestava je zachycena na obrázku 21.



Obrázek 21. Ovládání programu „Rozvrh hodin žáka“

- **Rozvrh hodin lektora** je rozpis naplánovaných předmětů v konkrétních časech, které bude vyučovat. Kliknutím na tuto položku se otevře formulář „Tisky lektorů“, který obsahuje ComboBox, ve kterém si navolíme lektora, čímž načteme jeho naplánované vyučovací dny. Vzhledem k tomu, že záznamy lektorů neustále přibývají, můžeme je vybírat filtrem podle data, v jehož rozmezí se sestava zobrazí a je připravena pro tisk. Kliknutím na textové pole datumu se otevře formulář „Kalendář“, ve kterém je možné si zvolit datum a díky tomu nemůžeme zadat špatný formát datumu do tohoto pole. Při odznačeném políčku „Filtrovat“ se zobrazí všechny záznamy lektora. Sestava lektora je zachycena na obrázku 22.
- **Archív:** Kliknutím na záložku se otevře formulář, který zachycuje seznam žáků, kteří již absolvovali nějaký kurz či školení. Je-li absolvent přesunut do archívu, nezobrazuje se už v hlavním formuláři v ComboBoxu žáci. Chceme-li absolventa přesunout zpět do hlavního formuláře za účelem úpravy dat, můžeme to provést kliknutím na CheckBox ve sloupci Kurz

Tisky lektoru

Vyber lektora pro tisk

Filtr podle datumu

Příjmení: Šindelová, Jméno: Kamila, Datum od: 04.prosince.2012, Datum do: 22.prosince.2012, Filtrovat

Hlavní sestava

Rozvrh hodin lektora Šindelová

Datum	Den	Začátek	Konec	Předmět 1	Naplnění	Předmět 2	Naplnění
02 březen 2011	středa	12:00	16:00	Praxe M+NM	2,00	Praxe P+NM	3,00
07 březen 2011	pondělí	12:00	16:00	Praxe M+NM	2,00	Praxe P+NM	3,00
09 březen 2011	středa	12:00	16:00	Praxe M+NM	2,00	Praxe P+NM	3,00
14 březen 2011	pondělí	12:00	16:00	Praxe M+NM	2,00	Praxe P+NM	3,00
16 březen 2011	středa	12:00	16:00	Praxe M+NM	2,00	Praxe P+NM	3,00
24 březen 2011	čtvrtek	16:00	20:00	Testy	1,00		
26 březen 2011	sobota	08:00	16:00	Zkoušky	1,00		
30 březen 2011	středa	08:00	16:00	Účetnictví	1,00		
16 listopad 2012	pátek	16:00	20:00	Technika M+NM	2,00		
17 listopad 2012	sobota	08:00	12:00	Technika M+NM	2,00		

Obrázek 22. Ovládání programu „Rozvrh hodin lektora“

hotov v tabulce „Archív žáků“, viz. obr. 23. Formulář rovněž obsahuje dva filtry. První podle příjmení, druhý podle kurzu. Filtr hledá shodu podle posloupnosti písmen do něj zadaných a po filtraci se pole nastaví na prázdný znak. Kliknutím na tlačítko „Filtr“ s prázdným znakem se provede načtení všech absolventů z databáze.

Archív

Archív žáků

Filtr podle příjmení: Filtr podle kurzu:

Příjmení	Jméno	Typ kurzu	Poznámka	Cena kurzu	Datum zahájení	Kurz hotov	Kopie
Nálepa	Richard	M+NM	Manikúra včetně n...	5900	25.2.2011	<input checked="" type="checkbox"/>	1
Dolníček	Tomáš	Jednodenní NM		1500	10.3.2011	<input checked="" type="checkbox"/>	0
Dulík	Petr	Jednodenní NM		1500	10.3.2011	<input checked="" type="checkbox"/>	0
Vašica	Jiří	Jednodenní NM		1500	10.3.2011	<input checked="" type="checkbox"/>	0
Peršala	Marek	Jednodenní NM		1500	10.3.2011	<input checked="" type="checkbox"/>	0

Obrázek 23. Ovládání programu „Archív žáků“

- **Platby:** Záložka Platby obsahuje čtyři položky, viz. obr. 24.
- **Nová platba** je formulář, ve kterém je možno zaznamenávat přijaté finanční částky. V ComboBoxu si navolíme žáka, zvolíme datum, typ platby



Obrázek 24. Ovládání programu „Platby“

a poté se tlačítko „Platba“ stane aktivním a je možné záznam uložit. Platba může být provedena hotově, převodem či kartou, a to najednou, nebo po částech. V tabulce je vidět přehled plateb navoleného žáka, dole pod tabulkou pak informace o ceně kurzu, zaplacené částce celkem a o doplatku, který je nutno ještě uhradit. Dále formulář umožňuje vytisknout navolenou platbu kliknutím na nulovou pozici tabulky, popřípadě všechny platby navoleného žáka a kurzu. Formulář je znázorněn na obrázku 25.

Nová platba žáka

Příjmení a jméno žáka: Elingová Eva Název kurzu: M+NM 0 Manikúra včetně nethové modeláže Číslo dokladu: 19
Cena : 5900 Kč Kopie : 0

Datum: 19.prosince.2012 Částka: Typ platby: Hotově Poznámka: Platba

Tabulka plateb vybraného žáka

Datum	Jméno	Příjmení	Adresa	Evidenční číslo	Kurz	Cena	Částka	Typ platby	Kopie	Číslo dokladu
6.11.2012	Eva	Elingová	Havířov		M+NM	5900	3000	Hotově	0	7
19.12.2012	Eva	Elingová	Havířov		M+NM	5900	1000	Hotově	0	18

Cena kurzu : 5900 Kč **Zaplaceno :** 4000 Kč **Chybí zaplatit :** 1900 Kč

Tisk Tisk všech dokladů Vymaž Zavři

Obrázek 25. Ovládání programu „Nová platba“

- **Přehled plateb** je formulář, ve kterém jsou zobrazeny všechny platby žáků. Formulář obsahuje filtry podle příjmení a podle kurzu. Filtr hledá shodu podle posloupnosti písmen do něj zadaných a po filtraci se pole nastaví na prázdný znak. Kliknutím na tlačítko „Filtr“ s prázdným znakem se provede načtení všech žáků z databáze. Kliknutím na textové pole „Filtr podle datumu“ se otevře formulář „Kalendář“, ve kterém je možné si zvolit datum a díky tomu nelze zadat špatný formát datumu.
- **Pohledávky** je formulář pro zobrazení žáků, kteří navštěvují kurz a nemají zaplacenou celou částku za kurz. Formulář obsahuje filtry podle příjmení a podle kurzu. Filtr hledá shodu podle posloupnosti písmen do něj zadaných a po filtraci se pole nastaví na prázdný znak. Kliknutím na tlačítko „Filtr“ s prázdným znakem se provede načtení všech žáků z databáze.
- **Typy plateb** je formulář, kde přednastavujeme typy plateb.
- Info: V záložce Info se nachází informace o programu a podrobný manuál.

5. Programátorská část

5.1. Použité technologie

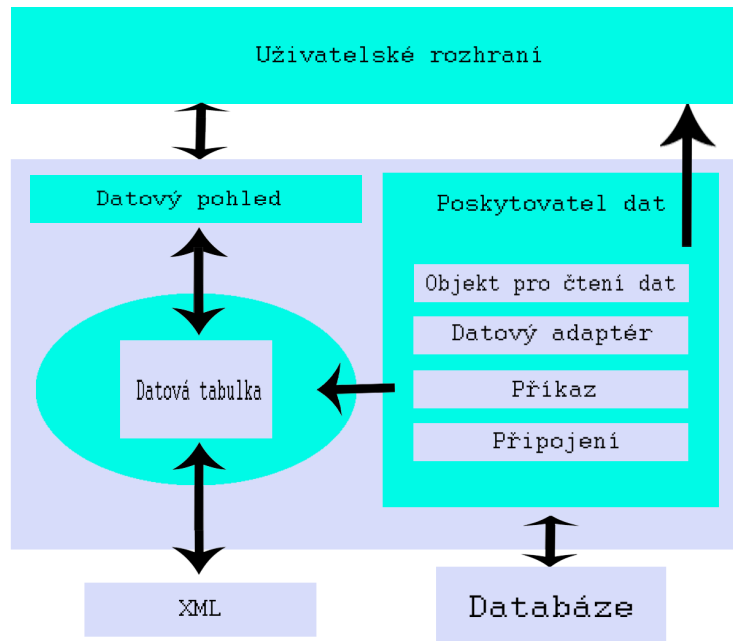
Aplikace byla postupně vytvářena ve vývojovém prostředí Microsoft Visual Studio 2008 Professional SP1, později překlopena do Microsoft Visual Studio 2010 Professional SP3. K programování byl použit programovací jazyk C# s použitím technologie .NET Framework 4. Pro uložení dat byla užitá lokální souborová databáze MS SQL Server ve spojení s technologií ADO.NET. Většina SQL dotazů byla vytvořena v Microsoft SQL Server Management studiu, později kombinována s SQL dotazy ve Wizard Manageru C#. Aplikace je založena na třívrstvě modelu, který tvoří vrstvy: datová, logická a prezentační. Tento model se zpravidla využívá ke tvorbě aplikací pracujících s databázemi.

5.1.1. Platforma .NET

Je to spojující název pro technologie, které jsou použity v softwarových produktech tvořících celou platformu pro Web, Windows i Pocket PC. Hlavní komponenta je .NET framework, což je prostředí potřebné pro běh aplikací, které obsahuje spouštěcí rozhraní i potřebné knihovny. Bez ohledu na to, v čem byla aplikace původně napsána, se vždy přeloží do mezijazyku Common Intermediate Language.

5.1.2. Technologie ADO.NET

Je to soubor softwarových komponent, které se dají využít k přístupu k datům a datovým službám, viz obr. 26. Jsou součástí základní knihovny, která je v Microsoft .NET Framework. Běžně je užívána programátory k přístupu a změnám dat v relačních databázových systémech, v tomto projektu konkrétně k datům v databázi Microsoft SQL Serveru. Výhody použití spočívají v jednoduchosti a rychlosti při zpracování dat, kde postačuje vytvořit spojení s potřebným serverem, a pak za pomoci zvoleného adaptéru a vhodně zadaným SQL dotazem je možné získat data z databáze a ta pak načíst do některé z připravených konstrukcí pro práci s daty z tabulek. Třídy knihovny ADO.NET se nacházejí v souboru System.Data.dll.



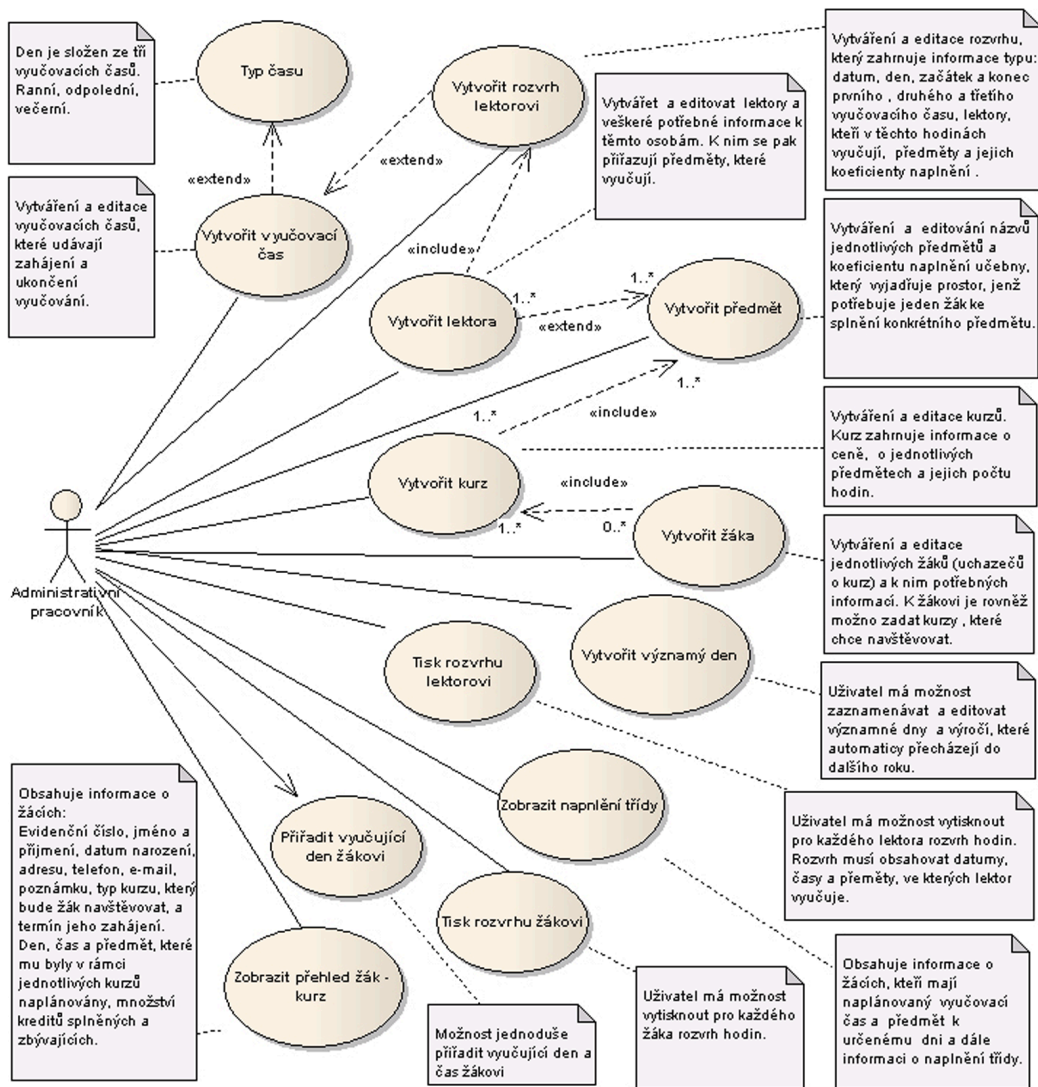
Obrázek 26. Architektura ADO.NET

5.1.3. MSSQL

Microsoft SQL Server je nejmodernější relační databázový systém společnosti Microsoft. Programátorům umožňuje navrhovat a spravovat relační databáze či vyhledávat větší objemy dat. Dále zajišťuje optimální výkon a odolnost databáze. Dotazy jsou prováděny v jazyce SQL.

5.2. USE CASE diagram aplikace

Na základě několika jednání se zadavatelem byl vytvořen Use Case diagram, do kterého se promítly jeho požadavky o fungování budoucí aplikace. Administrativní pracovník vyjádřil svou představu provozovat aplikaci na vlastním PC a zformuloval konkrétní požadavky, viz obr. 27.



Obrázek 27. USE CASE diagram aplikačního programu „Use Case diagram“

5.3. Databáze

Aplikace potřebuje pro svůj chod souborovou databázi ve formátu .mdf (MSSQL) s názvem KURZ, která je součástí programu. Databáze byla vytvořena v Microsoft SQL management studiu a později byla přesunuta do aplikace.

5.3.1. Popis základních vazeb databáze

- Vazba Žák - Kurz (M:N)
 - Analýza (zdůvodnění) - Kolik žáků(uchazečů) může navštěvovat jeden kurz? Předpokládejme, že obecně platí více (M). Kolik kurzů může navštěvovat jeden žák? Předpokládejme, že platí více (N). Proto je kardinalita vazby M:N.
 - Realizace vazby – Byla vytvořena vazební tabulka (Zak_Kurz). Primární klíč (Id_Zak) z první relace (Zaci) byl zařazen do nové relace (Zak_Kurz). Podobně primární klíč (Id_Kurzu) z druhé relace (Kurz) byl přidán do nové relace (Zak_Kurz). Dále do vazební tabulky byly přidány položky (Datum_Zahajeni), (Kurz_Hotov) a (Kopie).
- Vazba Kurz - Předmět (M:N)
 - Analýza (zdůvodnění) - Může kurz obsahovat více předmětů? Předpokládejme, že může (M). Může předmět být součástí více kurzů? Předpokládejme, že může (N). Proto je kardinalita vazby M:N.
 - Realizace vazby - Byla vytvořena vazební tabulka (Kurz_Predmet). Primární klíč (Id_Kurz) z první relace (Kurzy) byl zařazen do nové relace (Kurz_Predmet). Podobně primární klíč (Id_Predmet) z druhé relace (Predmet) byl přidán do nové relace (Kurz_Predmet). Dále do vazební tabulky byla přidána položka (Pocet_Kreditu).
- Vazba Lektor - Předmět (M:N)
 - Analýza (zdůvodnění) - Může lektor vyučovat více předmětů? Předpokládejme, že může (M). Může předmět být vyučován více lektory? Předpokládejme, že může (N). Proto je kardinalita vazby M:N.
 - Realizace vazby - Byla vytvořena vazební tabulka (Lektor_Predmet). Primární klíč (Id_Lektor) z první relace (Lektor) byl zařazen do nové relace (Lektor_Predmet). Podobně primární klíč (Id_Predmet) z druhé relace (Predmet) byl přidán do nové relace (Lektor_Predmet). Další atributy tato vazební tabulka nemá.

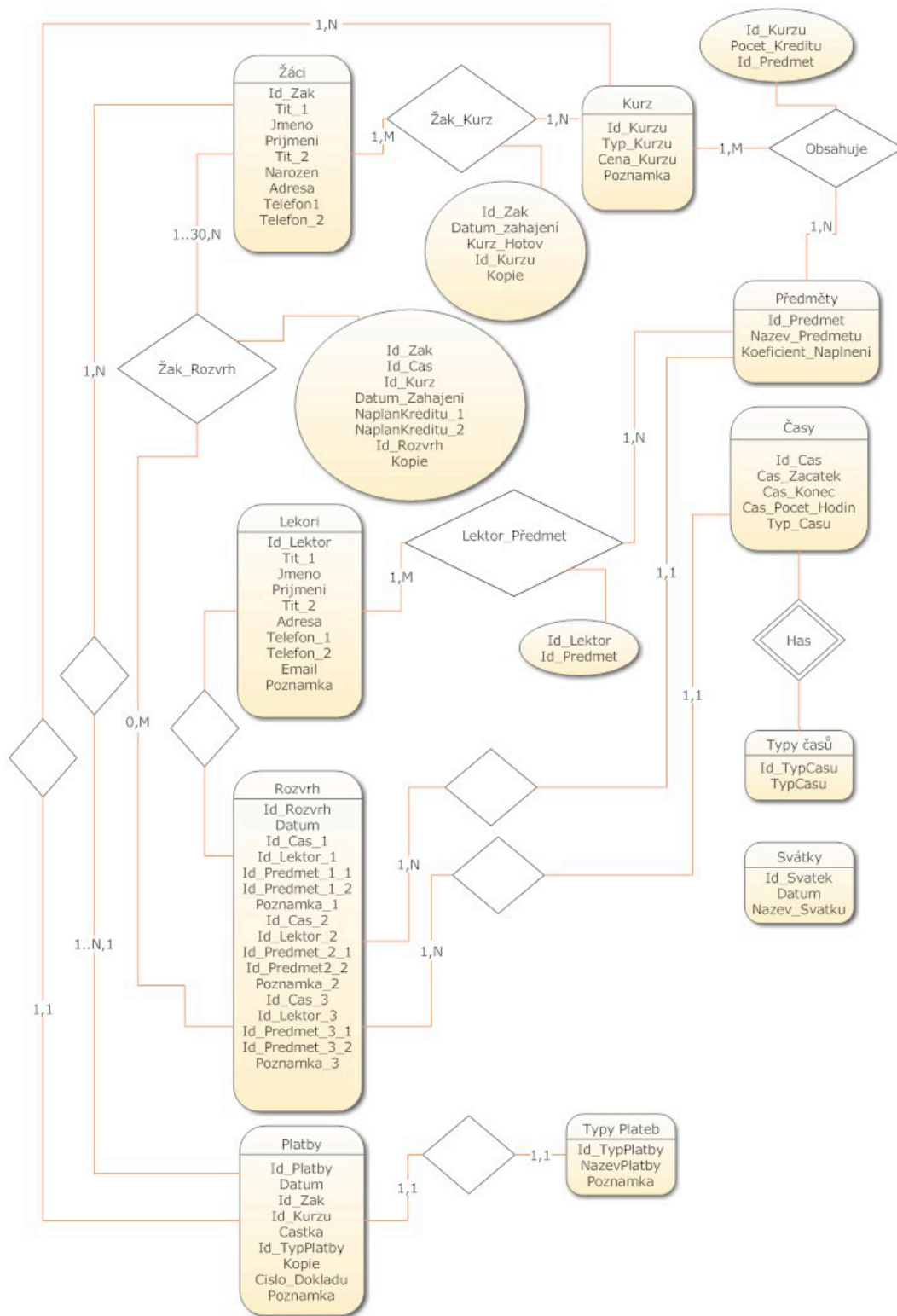
- Vazba Žák - Rozvrh (M:N)
 - Analýza (zdůvodnění) - Kolik žáků může přijít na vyučovací hodinu? Předpokládejme, že 0 - 30, tj. omezení kapacitou třídy. Kolika žákům může být přiřazen jeden učební den? Předpokládejme, že více (N). Proto je kardinalita vazby M:N. V tomto případě je použit unikátní klíč(Id_Zak/Datum), poněvadž žák může přijít v jeden den jen jednou.
 - Realizace vazby - Byla vytvořena vazební tabulka (Zak_Rozvrh). Primární klíč (Id_Zak) z první relace (Zak) byl zařazen do nové relace (Zak_Rozvrh). Podobně primární klíč (Id_Rozvrh) z druhé relace (Rozvrh) byl přidán do nové relace (Zak_Rozvrh). Dále do vazební tabulky byly přidány položky (Datum_ZahajeniKurzu, Id_Kurzu, _Cas, Kopie, Kredity_1, Kredity_2).
- Vazba Platba - Žák (N:1)
 - Analýza (zdůvodnění) - Kolik plateb může mít jeden žák? Obecně platí, že více (N). Kolik žáků může mít jednu platbu? Vždy jen jeden (1).
 - Realizace vazby - Primární klíč (Id_Zak) z relace s kardinalitou 1 (Zaka) byl přidán do relace s kardinalitou N (Platby).

5.3.2. Seznam základních pohledů

- View_Archiv
- View_Platby
- View_PrehledPlatby
- View_Rozvrh_Lektori
- View_Rozvrh_Zaci
- View_Suma_Plateb
- View_Prehled_Platby
- View_Rozvrh

5.3.3. ER diagram

ER diagram je zachycen na obrázku 28.



Obrázek 28. ER model databáze „ER model“

5.4. Popis hlavních tříd a jejich metod

5.4.1. Třídy formulářů

- MP_Kurzy.cs - Hlavní dynamický formulář vytvořen jako dlouhodobý kalendář zachycující veškeré informace o žácích, vyučovacích časech a předmětech lektorů, naplnění učebny a přehledy žáků. Pro naplnění je použito několik přesně definovaných dotazů do databáze KURZ.mdf, které se mění v závislosti na výběru žáka, pohybem rollerem myši či výběrem v komponentě kalendáře. Každá událost vyvolaná uživatelem se okamžitě promítne do SQL dotazu, data se aktualizují a znázorní na přesně definované pozici.
- Predmety.cs - Formulář, ve kterém je zobrazen přehled předmětů a jejich koeficient naplnění. Položky v tabulce je možno editovat.
- ZadejPredmet.cs – Formulář pro vytváření nových předmětů.
- Casy.cs – Formulář, ve kterém je zobrazen přehled vyučovacích časů, jejich počet hodin a typ. Položky v tabulce je možno editovat.
- ZadejTX.cs – Víceúčelový formulář pro zadávání časů, koeficientu naplnění a pro editaci jednotlivých položek. Pomocí metod, které obsahuje, je z něj možno přenastavit formulář pro daný účel a vstupní hodnoty validuje pomocí regulárních výrazů.
- Kurzy.cs – Formulář, ve kterém je zobrazen přehled všech kurzů, jejich názvy, cena, poznámka a rovněž editace předmětu každého kurzu. Položky v tabulce je možno editovat.
- ZadejKurz.cs - Formulář pro vytváření nových kurzů.
- ZadejPredmetKurz.cs - Formulář pro přiřazení předmětů kurzu při jeho vytváření.
- ZadejKredity.cs - Formulář pro zadávání kreditů předmětu při vytváření kurzu.
- ZadejKurzZakovi.cs - Formulář pro zadávání kurzu žákovi.
- Lektori.cs – Formulář, ve kterém jsou zachyceny veškeré informace o lektorech. Položky v tabulce je možno editovat včetně předmětů, které vyučují.
- Zaci.cs – Formulář, ve kterém jsou zachyceny veškeré informace o žácích. Položky v tabulce je možno editovat včetně kurzů, které žák navštěvuje.
- ZadejLektorZak.cs - Formulář pro zadávání nových žáků nebo lektorů.

- [PrehledPlateb.cs](#) - Formulář, který zachycuje veškeré informace o žácích a jejich platbách.
- [Neplatici.cs](#) - Formulář, který zachycuje informace o žácích a jejich pohledávkách.
- [TypyPlateb.cs](#) - Formulář, ve kterém lze předvolit druh platby.
- [NovaPlatba.cs](#) - Formulář pro zadávání plateb jednotlivým žákům.
- [TiskPlatby.cs](#) - Formulář tisku příjmových dokladů, popřípadě přehledu plateb.
- [Rozvrh.cs](#) - Formulář, který zachycuje přehled vyučovacích hodin, předmětů a lektorů, kteří je vyučují.
- [ZadejRozvrh.cs](#) - Formulář pro vytváření nového rozvrhu.
- [Kalendar.cs](#) - Formulář pro zadávání datumu ve všech formulářích.
- [NastaveniVyhл.cs](#) - Formulář, ve kterém se nastavují parametry pro vyhledávání.
- [TiskyLektori.cs](#) - Formulář, ve kterém je možno navolit parametry pro tisk lektorů a následně je vytisknout.
- [TiskyZaci.cs](#) - Formulář, ve kterém je možno navolit parametry pro tisk žáků a následně je vytisknout.
- [ListWiu.cs](#) - Přehledy obsazení třídy ve zvolený den.
- [Vyhledat_X.cs](#) - Formulář pro podrobnější vyhledávání poznámky či lektora v hlavním formuláři.
- [ZadejSvatek.cs](#) - Formulář pro zadávání svých svátků.
- [Archiv](#) - Formulář, ve kterém je zobrazen přehled absolventů za předchozí období.
- [Help.cs](#) - Formulář podrobného návodu.
- [Oprogramu.cs](#) - Formulář zachycující informace o programu.

5.4.2. Třídy

- Program
- Pripojeni.cs - Třída, která obsahuje metody pro práci s databází.
- Prevody.cs - Třída, která obsahuje metody pro převody různých typů s validací vstupních hodnot.
- Poruchy.cs - Třída, která obsahuje přednastavená dialogová okna.
- Svatky.cs - Třída, která obsahuje metody pro kontrolu svátků zadaného dne, popřípadě Velikonoc v daném roce.
- MultiCombo.cs - Spojení dvou sloupců v tabulce. Příprava pro zobrazení např. Jména a Příjmení v jednom ComboBoxu.

5.4.3. Tiskové třídy Crystal Report

- CrystalReportDataSet
- crNaplneni.rpt
- crPlatby.rpt
- crTiskRozvrhLektora.rpt
- crTiskZaka.rpt

5.4.4. Vybrané třídy a jejich metody

Aplikace se skládá z mnoha formulářů, z nichž každý disponuje řadou metod, proto uvádím jen ty nezákladnější.

Třída „Pripojeni“

Třída připojení obsahuje statické metody pro připojení k databázi. Tyto metody se používají ve většině formulářů, kdy je třeba vytvořit, načíst či nějak upravit data v databázi. Připojení k SQL serveru je zprostředkováno třídou System.Data.SqlClient.SqlConnection. Pro připojení je důležité znát potřebné údaje, které jsou obsaženy v tzv. připojovacím řetězci. Ten můžeme třídě přiřadit v konstruktoru této třídy nebo pomocí vlastnosti ConnectionString. Ukončení připojení k databázi zprostředkováváme metodou Close ze třídy Connection.

Metody třídy „Pripojeni“

- Public static bool pripojSQL(ref bool blPripojeno) - metoda zajistí připojení k databázi a vrací binární hodnotu true, jestliže je databáze připojena. V případě, že se z jakéhokoliv důvodu nepodaří připojit k databázi, metoda zachytí výjimku, vypíše ji a vrátí hodnotu false.
- Public static bool pripojeniClose(ref bool blPripojeno) - metoda odpojí databázi a vrátí binární hodnotu false.
- Public static DataTable nactiData(string selectQuery) - metoda vrací naplněnou tabulku daty z databáze po provedení dotazu, který je jako parametr.
- Public static void metUpdateQuery(string strUpdateQuery) - metoda provádí změny jednotlivých hodnot v databázi.
- Public static string metVycti_String(string strQuery) - metoda vyčte jednu položku z databáze na základě dotazu, který je jako parametr. Vrací hodnotu string.
- Public static TimeSpan metVycti_Time(string strQuery) - metoda vyčte jednu položku z databáze na základě dotazu, který je jako parametr. Vrací hodnotu TimeSpan.
- Public static int metVycti_INT(string strQuery) - metoda vyčte jednu položku z databáze na základě dotazu, který je jako parametr. Vrací hodnotu INT.
- Třída rovněž obsahuje několik překladů chybových hlášení, která jsou postupně doplňována. Na základě čísla poruchy vrátí text chybového hlášení.

```
public static string metSQLporuchy(int intCisloPoruchy)
{
    switch (intCisloPoruchy)
    {
        case 2627: textPoruchy = "Tento záznam již existuje !! ";
            break;
        case 109: textPoruchy = "Uložení do databáze neproběhlo korektně !! ";
            break;
        case 241: textPoruchy = "Špatný format datumu !! ";
            break;
        case 547: textPoruchy = "Nejde odstranit, záznam se používá !! ";
            break;
        ..... atd.
    }
}
```

Veškerá práce s daty je vždy prováděna ve třech krocích:

1. Připojení k databázi pomocí metody „PripojSQL()“.
2. Provedení SQL dotazu pomocí metod „nactiData()“, „metUpdateQuery()“, „metVycti_String()“, „metVycti_Time()“, „metVycti_INT()“.
3. Zavření databáze pomocí metody „pripojeniClose()“.

Příklad části kódu metody pripojSQL()

```
public static bool pripojSQL(ref bool blPripojeno)
{
    SqlConnection datovePripojeni = new SqlConnection();
    datovePripojeni.ConnectionString = "Data Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|
    \\KURZ.mdf;Integrated Security=True;
    Connect Timeout=30;User Instance=True";

    try
    {
        datovePripojeni.Open();
        blPripojeno = true;
    }
    catch (Exception ex)
    {
        string porucha = ex.Message;
        Poruchy.metErrorr(porucha);
        blPripojeno = false;
    }

    blPomocnaPripojenoStatic = blPripojeno;

    return blPripojeno;
}
```

Metody formuláře „MP_Kurzy“

Při spuštění aplikace se otevře hlavní formulář, který slouží jako základní obrazovka. Je rozdělen do několika GroupBoxů. O načtení horního GroupBoxu se starají lokální metody „private void metNactiFormular()“ a „private bool metProjdiGroup1_naplňTextBox()“, která vrací hodnotu true, jestliže je kurz celý naplánován. V opačném případě hodnotu false. Základem metody je projít všechny komponenty GroupBoxu1, zjistit, zda jde o TextBox, a naplnit jej hodnotou z předem načtené tabulky. Metoda rovněž obarvuje TextBox „Již naplánováno“ správnou výše zmíněnou barvou. Při změně indexu v jednom ze tří ComboBoxů se aktualizují data v tabulce, ze které se plní GroupBox1 (žáků), a po načtení hodnot z databáze se opět zavolá metoda „private bool metProjdiGroup1_naplňTextBox()“ a tím se údaje v TextBoxech aktualizují.

- Při načtení formuláře se do ComboBoxu, který se nachází v GroupBoxu1 (žáků), načtou všechna příjmení a jména žáků a jejich aktuální kurzy.
- Do TextBoxu Zahájení kurzu se načte datum první vyučovací hodiny (pokud existuje). Ten se automaticky zaznamená při prvním naplánování. V případě, že druhé a další naplánované datum je nižší, dojde vždy k zapsání nejmenší hodnoty do databáze.
- Do TextBoxu **Předmět, Plán a Již** se zapisují dynamicky měnící se hodnoty z databáze na základě výběru hodnoty jednoho ze tří ComboBoxů. TextBox „Předměty“ znázorňuje všechny předměty, které obsahuje vybraný kurz. „Plán“ zachycuje počet hodin, které je třeba naplánovat, a „Již“ je hodnota, která udává, kolik hodin má aktuální žák již naplánováno. Pole

„Již“ disponuje pro přehlednost třemi barvami. Bílá, jestliže plán je větší než doposud naplánovaná hodnota. Zelená, jestliže hodnoty se rovnají, a červená, jestliže plán je menší než naplánovaná hodnota.

- O přepis oznamovacího textu, zda je kurz celý naplánován či nikoli, se stará metoda „private void metPrepisLabelKurzHotov(bool blKurzHotov)“.
- Metoda „private void metPresunDoArchivu()“ zajistí přesun do archívu. Před samotným přesunem je možno si vybrat, zda chceme žáka přesunout či nikoli. Po přesunutí do archívu se absolvent nezobrazuje v ComboBoxu hlavního formuláře a tím pádem nelze provádět změny v jeho rozvrhu. Chceme-li u absolventa provádět nějaké změny, je nutné jej vyjmout z archívu kliknutím na jeho CheckBox „Kurz hotov“ ve formuláři „Archív“.

Kód metody metPresunDoArchivu()

```
private void metPresunDoArchivu()
{
    string sqlUpdatekurzHotov =
        "update Zak_Kurz set KurzHotov = 1 where Id_Zak = " + Id_Zaka_akt_CBox + "
        and Id_Kurzu = " + Id_Kurzu_akt_CBox + " and Kopie = " + ID_Kopie_akt_CBox + ";
    Pripojeni.pripojSQL(ref blPripojeno);
    Pripojeni.metUpdateQuery(sqlUpdatekurzHotov);
    Pripojeni.pripojeniClose(ref blPripojeno);
}
```

- V pravém horním rohu se zobrazují státní svátky, které jsou pevně zadány v databázi anebo mohou být také přidány ručně. Zavoláním metody „Svatky.metVypisSvatek(dtDnesniDatum, true)“, ze třídy svátky, se přepíše label, který nese tuto informaci. Metoda taky vypočítá a zkontroluje, zda datum, zadané v jejím parametru, nejsou Velikonoce.
- Kliknutím na button „Přehled žáka“, který se nachází po pravé straně GroupBoxu1 (žáků), se zobrazí náhled tiskové sestavy vytvořené prostřednictvím komponenty „CrystalReportViewer“, která znázorňuje naplánované dny aktuálního žáka a kurzu.
- Po načtení horního GroupBoxu1 je nutné naplnit spodních sedm GroupBoxů, které představují jednotlivé dny v kalendáři, vyučovací časy, vyučujícího lektora, naplnění třídy a „CheckBox“, který znázorňuje, zda je v tento den aktivně navolený žák přihlášen. Údaje se dynamicky mění na základě vyvolání události, kterou může být:
 - **Kliknutím na datum v kalendáři** - Tímto se provede výběr rozvrhu hodin lektorů následujících sedmi dnů z databáze od tohoto data. Tyto hodnoty se zapíšou pomocí „private void metNaplnBoxDatumem(Control conThis,DateTime dtDatum)“, „private void

metNapnGroupBox(int intOd)“ a „private void metProjdiKomponenty(Control rootControl,int intRadek)“. Zvolené datum se zobrazí vždy na první pozici shora a od něho směrem dolů následujících šest dní. Kromě těchto údajů se ještě vypočítávají hodnoty celkového obsazení třídy v aktuální den a vyučovací čas. Rovněž se kontroluje, zda je navolený žák v tento den zapsán na vyučování. O toto vše se stará metoda „private void metNaplnControls33()“, která také hlídá překročení obsazení třídy v daný vyučovací den a vyučovací čas a uživatele na to upozorní změnou barvy této hodnoty.

- **Posun kliknutím na button s ikonou šipky „Nahoru/Dolu“** se zavolá pomocná metoda s parametrem `cislo_1`, která obsahuje v sobě výše zmíněné metody. Parametr udává počet dní, o kolik se mají dny v kalendáři posunout.

Kód metody metPosunKalendare_o_1(cislo_1)

```

}
private void metPosunKalendare_o_1(cislo_1)
{
    vybraneDatum = monthCalendar1.SelectionStart;
    DatumPosun = vybraneDatum.AddDays(cislo_1);

    metNaplnBoxDatumem(this, DatumPosun);
    metNapnGroupBox(0);
}

```

- Pohybem kolečka na myši vyvoláme událost `OnMouseWheel()`, která je obsloužena níže uvedeným kódem.

Kód metody posun pomocí kolečka na myši

```

protected override void OnMouseWheel(MouseEventArgs e)
{
    int delta = e.Delta;
    if (delta == +120)
    {
        cislo_1 = cislo_1 - 1;
        metPosunKalendare_o_1(cislo_1);
        return;
    }

    if (delta == -120)
    {
        cislo_1 = cislo_1 + 1;
        metPosunKalendare_o_1(cislo_1);
        return;
    }
}

```

Takto načtený formulář máme připravený pro zadávání vyučovacích časů navolenému žákovi a k jeho kurzu. Pomocí události „`button_Click`“ jednotlivých tlačítek, jejichž vlastnosti `Button.Text` nesou informace o vyučovacích hodinách

jednotlivého časového úseku v daný den, přiřazujeme vyučovací časy jednotlivým žákům. Kliknutím na jedno z jednadvaceti tlačítek voláme stejnou metodu „metClikNaTlacitko(groupBox2, 0)“ pouze s jinými parametry. První parametr je číslo GroupBoxu, druhý parametr udává typ času. Hodnota nula představuje ranní čas, jednička odpolední a dvojka večerní. Metoda „metClikNaTlacitko()“ má v sobě implementováno několik funkcí.

- Button, na který jsme kliknuli, z důvodu přehlednosti změni barvu, na světle modrou.
- Zkontroluje, zda je v tento den naplánován nějaký předmět.
- Zkontroluje, zda vyučovací předmět v tento den spadá do navoleného kurzu.
- Jestliže vyučovací den a čas obsahuje dva předměty, zkontroluje, zda je zvolený jeden ze dvou předmětů.
- Zkontroluje, zda v tento den a čas již nemá žák zapsaný předmět.
- Otevře se dialogové okno „Zadej kredity/hodiny“ s přednastavenou hodnotou hodin vypočtenou z časů, které tlačítko nese ve vlastnosti „Button.Text“. Hodnota se může měnit podle toho, jestli žák může navštívit celé vyučování, nebo jen zvolenou část.
- Provede validaci zadané hodnoty.
- Provede záznam do databáze.
- Přepočítá a přepíše naplnění a obsazení třídy.
- Překontroluje datum zahájení kurzu, případně zapíše do databáze nové, nižší datum zahájení kurzu.
- Zavolá výše zmíněné metody pro občerstvení dat horního GroupBoxu1.
- Zjistí, zda kurz je celý naplánován, a v případě, že je, zavolá metody potřebné pro přesun do archívu.
- Button, na který bylo kliknuto, se přebarví na původní barvu.

Takovým způsobem lze naplánovat každému žákovi jeho vyučovací hodiny, přitom kontrolovat, zda počet hodin nepřekročil plán a zda naplnění třídy nepřesáhlo její kapacitu. Kromě toho je možné sledovat obsazení třídy a v přehledu žáka vyučovací termíny. To vše lze dynamicky měnit na základě zvoleného datumu. Kromě toho horní GroupBox0 disponuje rychlým vyhledáváním předmětu při vytváření rozvrhu žákovi, tj. kliknutím na předmět navoleného kurzu se zavolá metoda „private void metVyhledej(string strNazevTabulky, string strNazevRetezce,string

coVyhledat)“, která obstará vyhledání předmětu v tabulce rozvrhu, zavolá metody pro nastavení hlavního formuláře `metNaplnBoxDatumem(this, dtHledaneDatum)` a `metNapnGroupBox()`. Vyhledávání se provádí od datumu zahájení kurzu, tj. od minimálního data. To, že kliknutím na textové pole vyvoláme nějakou událost, konkrétně vyhledání předmětu, poznáme změnou kurzoru na šipku při najetí na textové pole.

Chceme-li odstranit žákovi záznam, stačí kliknout na zatrhnutý `CheckBox`, který je umístěn vedle naplnění třídy, tímto se otevře dialogové okno s dotazem, zda si přejeme záznam odstranit, a potvrzením záznam odebereme. Po odebrání se aktualizují data ve všech `GroupBoxech` pomocí výše uvedených metod.

Závěr

Aplikace je aktuálně používána v praxi a ze strany uživatele stále ještě dochází k požadavkům na nové funkce programu. Některé jsou minimální, jiné předpokládají větší zásah do systému. Proto změny průběžně konzultujeme, sepisujeme a plánujeme vhodný způsob a pořadí těchto úprav.

Plánované úpravy:

- Generování evidenčních čísel pro osvědčení.
- Grafické zpracování a tisk osvědčení.
- Tisk třídní knihy u rekvalifikačních kurzů akreditovaných MŠMT, která musí obsahovat přehled denní docházky všech absolventů u jednotlivých oborů.
- Statistické přehledy návštěvnosti absolventů za určitá období, úspěšnosti skládání závěrečných zkoušek absolventů aj.
- Odesílání e-mailů přímo z aplikace (zrušení hodin, nový termín, nový rozvrh, nabídky nových kurzů atd.).

Vytvořená aplikace odpovídá zadání práce a disponuje veškerými požadavky zadavatele. Při vytváření programu MP_Kurzy jsem čerpal z teoretických znalostí, kterých jsem nabyl až při studiu na Univerzitě Palackého v Olomouci, a z níže uvedené literatury.

Conclusions

Application is currently working and from side of users are still incoming requirements to add new functions in program. Some requirements are simple, other requirements suppose large changes in program. We are continuously discussing about changes and we planning best method and sequence of this changes.

Planned modifications:

- Generating registration numbers for certification.
- Graphic design and print certificates
- Print class book in retraining courses which are accredited by Ministry of Education, which must contain an overview of the daily attendance of all graduates of the fields of study.
- Statistical reports about visits of graduates in certain time periods, success composing the final examinations of graduates, etc.
- Sending e-mails directly from the application (cancel of lessons, new date, new schedule, offering new courses, etc.).

The application corresponds with bachelor thesis and it has all requirements from task. During creating program MP_Kurzy I used theoretical knowledge from my study at Palacky University in Olomouc and from below list of literature.

Reference

- [1] Kent, Jeff *Visual C# 2005 bez předchozích znalostí*. Computer Press, a.s, Brno, 2007.
- [2] Sharp, John *Microsoft Visual C# 2008 - Krok za krokem*. Computer Press, a.s, Brno, 2008.
- [3] Mareš, Amadeo *1001 tipů a triků pro C#*. Computer Press, a.s, Brno, 2008.
- [4] Agarwal, Vrat, Vidya a Huddleston, James *Databáze v C# 2008 - Průvodce programátora*. Computer Press, a.s, Brno, 2009.
- [5] Kanisová, Hana a Müller, Miroslav *UML - srozumitelně* Computer Press, a.s, Brno, 2004.
- [6] Hotek, Mike *Microsoft SQL Server 2008- Krok za krokem*. Computer Press, a.s, Brno, 2008.

A. Obsah přiloženého CD

`bin/`

- instalátor programu

`doc/`

- dokumentace k bakalářské práci ve formátu PDF, zhotovená podle závazného stylu KI PřF UP Olomouc pro diplomové práce (získaná z webových stránek katedry), včetně všech příloh a souborů pro následné vygenerování PDF souboru dokumentace (uložená v ZIP archívu)

`src/`

- kompletní zdrojové soubory

`readme.txt`

- instrukce pro instalaci aplikace