

Integrace databázového systému do podnikového IS

Bc. Viktor Mechl

Diplomová práce
2006



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2005/2006

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Viktor MECHL**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Integrace databázového systému do podnikového IS**

Zásady pro vypracování:

- 1. Automatizace standardních úloh.**
- 2. Implementace bezpečnostních prvků.**
- 3. Rozšíření komunikace přes Intranet.**
- 4. Propojení báze dat s podnikovým IS.**

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

Pokorný Jaroslav: Spolupráce aplikací MS Office, KOPP, 1997.

Scott, Barker, F.: MS Access 2002 – Programování databázových aplikací, Computer Press, Praha, 2002.

Lacko, L.: Web a databáze, Praha, Computer Press 2001.


Schurman, E.M.: Dynamické HTML v akci, Computer Press, Brno, 2004.

Vedoucí diplomové práce: **Ing. Zdenka Prokopová, CSc.**
Ústav aplikované informatiky


Datum zadání diplomové práce: **14. února 2006**

Termín odevzdání diplomové práce: **26. května 2006**

Ve Zlíně dne 14. února 2006


prof. Ing. Vladimír Vašek, CSc.
pověřený děkan




doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Úkolem této diplomové práce je integrace databáze pro organizaci povinných školení do existujícího informačního systému firmy Barum Continental spol. s r.o. Jádrem aplikace je již používaná databáze Řidičů manipulačních vozíků zpracovaná v rámci bakalářské práce.

Nejdůležitějším požadavkem je zahrnutí všech oblastí povinných školení realizovaných u zadavatele, přehlednější poskytování většího množství informací při nižší časové náročnosti na obsluhu a umožnění jednoduché kontroly nad průběhem vzdělávací akce.

K vlastnímu vývoji je využito prostředků MS Access se zaměřením na využití programování ve VBA a API. Práce je součástí projektu přechodu vzdělávacího informačního systému na platformu Oracle.

Klíčová slova: data, databáze, informace, informační systém, MS Access, VBA, API, autentizace, autorizace, audit

ABSTRACT

The aim of this thesis is the integration of the annual compulsory tests database into the existing information system of the Barum Continental Company. The core of the application is the forklift drivers database that is already in use. This database was composed as the dissertation.

The most important criteria laid on the project are: full coverage of all the education areas that are relevant, clear organization, the accessibility of more pieces of information more easily and the availability of user friendly control over the whole educational course.

The development itself took place in MS Access with the concentration on the utilization of the VBA and API programming. The dissertation is part of the transition of the educational information system to the Oracle platform.

Keywords: data, database, information, information system, MS Access, VBA, API, authentication, authorisation, audit

Chtěl bych tímto poděkovat vedoucímu diplomové práce paní Ing. Zdence Prokopové, CSc. a garantovi panu Ing. Petru Kunzovi za odborné vedení, rady a připomínky, které mi poskytovali při řešení diplomové práce a také za poskytnutí vhodných podmínek pro úspěšné dokončení této práce. Dále bych chtěl poděkovat svým pracovním kolegům za vstřícné jednání při vzniku tohoto projektu.

Velký dík patří také celé mé rodině, především manželce Janě za trpělivost, kterou projevovala po celou dobu mého studia při výchově našich čtyř dětí.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 AUTOMATIZACE ÚLOH	10
1.1 EFEKTIVNÍ VYUŽÍVÁNÍ PC.....	10
1.2 POČÍTAČOVÁ GRAMOTNOST	11
1.3 PROPOJENÍ KANCELÁŘSKÝCH APLIKACÍ	12
1.4 VBA (VISUAL BASIC FOR APPLICATION)	13
1.5 API (APPLICATION PROGRAMMING INTERFACE).....	16
1.6 OPAKOVANĚ POUŽITELNÉ PROCEDURY.....	18
1.6.1 Pohyb mezi formuláři	18
1.6.2 Nastavení titulku formuláře.....	19
1.6.3 Export tabulek, dotazů a sestav	20
1.6.4 Vlastní Zprávy zobrazované na základě tabulky.....	21
1.7 VYUŽITÍ AUTOMATIZACE.....	23
2 BEZPEČNOST DAT	24
2.1 PRÁVNÍ ASPEKTY.....	24
2.2 RIZIKA A NEDOSTATKY	25
2.2.1 HW – Hardware	26
2.2.2 SW – Software.....	26
2.2.3 DW – Dataware	27
2.2.4 PW – Peopleware	27
2.2.5 OW – Orgware	27
2.3 PŘÍSTUP K DATABÁZI.....	28
2.4 AUTENTIZACE	29
2.5 AUTORIZACE	30
2.6 AUDIT	31
2.7 APLIKACE BEZPEČNOSTNÍCH OPATŘENÍ.....	32

II

PRAKTICKÁ ČÁST	33
3 UPŘESNĚNÍ CÍLŮ A ZADÁNÍ PROJEKTU.....	34
3.1 PROJEKTOVÝ RÁMEC	34
3.1.1 Současný stav	35
3.1.2 Cíle projektu	35
3.1.3 Etapy projektu	35
3.1.4 Harmonogram.....	36
3.2 DATABÁZOVÁ PLATFORMA	37
4 AUTOMATIZACE ÚLOH.....	38
4.1 KONTROLA PŘÍSTUPNOSTI ZDROJE DAT	38
4.2 DOPLNĚNÍ ÚDAJŮ O ZAMĚSTNANCI.....	41
4.3 PŘÍPRAVA PODKLADŮ NOVÉHO KURZU.....	43
4.3.1 Nový kurz	44
4.3.2 Opakovaný kurz volně přístupný	45
4.3.3 Opakovaný kurz uzavřený	47
4.4 VÝSTUPY PRO ROZHODOVÁNÍ	47
5 BEZPEČNOSTNÍ PRVKY.....	50
5.1 OVĚŘENÍ UŽIVATELE PODLE PŘIHLÁŠENÍ VE WIN XP – API.....	50
5.2 ÚPRAVA UŽIVATELSKÝCH PRÁV A VZHLEDU APLIKACE.....	52
5.3 SLEDOVÁNÍ ZMĚN KLÍČOVÝCH POLOŽEK	53
6 PROPOJENÍ SE STÁVAJÍCÍM IS	55
6.1 EXPORT DAT Z IS	55
6.1.1 Seznam zaměstnanců.....	55
6.1.2 Seznam oddělení.....	57
6.2 ARCHIV HISTORICKÝCH STAVŮ.....	58
6.3 KOMUNIKACE PŘES INTRANET	59
6.3.1 Informace o volných místech	60
6.3.2 Přehled o termínech vypršení platnosti oprávnění	60
6.3.3 Možnost objednávat se automaticky	61
6.4 EXPORT SOUHRNNÝCH DAT PRO PREZENTACE	61
6.5 ZAHRNUTÍ VŠECH OPAKOVANÝCH POVINNÝCH ŠKOLENÍ	62
ZÁVĚR.....	64
SEZNAM POUŽITÉ LITERATURY	65
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	67
SEZNAM OBRÁZKŮ	69
SEZNAM TABULEK.....	70
SEZNAM PŘÍLOH.....	71

ÚVOD

Tato diplomová práce se zabývá vývojem aplikace pro firmu Barum Continental, spol. s r.o. se zaměřením na navržení integrovaného řešení pro organizaci všech povinných školení a uvedením tohoto systému do provozu.

Základem aplikace je již používaná databáze Řidičů manipulačních vozíků zpracovaná v rámci bakalářské práce v prostředí MS Access. Záměrem této práce je rozšířit oblast úkolů o další typy školení a přidat další funkce. Vývoj je směřován k automatizaci, vyšší bezpečnosti a užšímu propojení se stávajícím informačním systémem. Hlavní myšlenkou je snaha o co nejefektivnější využívání možností IT a redukci nutných administrativních činností, které nepřinášejí přidanou hodnotu..

Teoretická část práce se zaměřuje na možnosti efektivního využívání PC, programování ve VBA a formy zabezpečení dat, nepokrývá tedy celou problematiku tvorby databáze. Elementární pojmy a znalosti potřebné pro vývoj databází jsou již popsány ve zdroji [5].

Praktická část objasňuje postup implementace požadovaných funkcí. Názorným způsobem jsou popsány jednotlivé moduly včetně ukázek programového kódu a vývojových diagramů. Nechybí ani popis projektového rámce pro zasazení do širších souvislostí.

Aplikace se stala součástí projektu přechodu na nový vzdělávací systém na platformě Oracle. Tímto rozhodnutím byla jasně specifikována role této aplikace jako funkčního modelu pro nalezení optimálního způsobu zpracování agendy povinných školení se současnou výukou uživatelů novým metodám práce. Tato skutečnost pochopitelně ovlivnila směr a priority dalšího vývoje aplikace.

I. TEORETICKÁ ČÁST

1 AUTOMATIZACE ÚLOH

Záměrem první části této rešerše je přiblížit několik základních metod vedoucích k efektivnějšímu využívání kancelářských aplikací. Druhá část se zabývá zvýšením bezpečnosti dat v databázových systémech.

1.1 Efektivní využívání PC

Každým rokem dochází k nárůstu rychlosti HW a s každou novou verzí SW hlásí výrobce vyšší efektivitu a více funkcí. Dnes už je zavádění výpočetní techniky do firem ve stadiu, kdy v každé kanceláři má odpovědný zaměstnanec svůj počítač. Nakolik zavedení informačních technologií skutečně zvýšilo produktivitu těchto firem, zůstává stále otázkou, na kterou není snadné odpovědět. Každá investice musí prokázat svou návratnost, u výpočetní techniky však není takové prokazování obvyklé. Místo snadné odpovědi typu "Všichni počítače používají, my musíme také", je nezbytné hledat širší význam PC pro konkrétní práci. Zaměstnanec se nestává výkonnějším jen tím, že používá počítač nebo určitý balík softwaru. Ke zvýšení výkonu potřebuje znalosti jak daných možností SW co nejlépe využít. Logickým důsledkem zavádění výpočetní techniky do praxe musí být také změna organizačních postupů, jinak jde pouze o "hraní si na počítači".

Není pochyb o tom, že současný HW i SW pro stolní počítače poskytují kapacitu pro zvýšení produktivity, ale aby bylo dosaženo skutečného nárůstu produktivity, musí dojít k restrukturalizaci pracovního procesu jako celku. Lidé mají totiž sklon používat počítač naprosto stejným způsobem, jako používali to, co PC nahradily. Uvažování na úrovni psacího stroje a kalkulačky však obvykle znamená spíše zhoršení efektivity, neboť funkcí a voleb je najednou příliš mnoho a znechucený uživatel konstatuje, že vyplnit formulář ručně bylo dříve daleko rychlejší (více o efektivitě ve zdroji [6]).

Tři principy zvýšení efektivity pracovních aplikací: (čerpáno ze zdroje [12])

- Změna pracovní plochy PC na tzv. "hlavní nádraží" pracovní komunikace. Standardní cesty napsat -> vytisknout -> okopírovat -> odfaxovat -> zaslat poštou nahradit distribucí dat přímo na, a z pracovní plochy.
- Ukládání dat pouze jednou s jejich opakovaným používáním. Shromažďování dat jako vedlejší produkt práce, namísto přímo zaměřené činnosti. Sdílení společných dat.

- Odstranění zbytečných činností a kroků navíc. Publikování dat v běžných aplikacích tak, aby byly ostatními dále využitelné k analýzám a dalšímu zpracování bez nutnosti zbytečných úprav.

1.2 Počítačová gramotnost

Kancelářské balíky obsahují několik samostatných aplikací. Každá z těchto aplikací je primárně určena k jinému účelu. Pro jiné úlohy je tedy určen textový editor, jiný význam má tabulkový kalkulátor a zcela jiné využití očekáváme od prezentačního softwaru. Moderní kancelářské balíky umožňují propojení dat mezi jednotlivými aplikacemi, ale praktické využití těchto funkcí není úplně jednoduché a uživatelé mnohdy nejsou na takové úrovni, aby dokázali těchto výhod efektivně využít. Nejčastější spojnici mezi jednotlivými aplikacemi je kopírování a vkládání objektů přes schránku.

Řešením tohoto neutěšeného stavu je důsledné měření úrovně znalostí ovládání aplikací, které uživatel při své práci potřebuje. Školení a testy tohoto druhu jsou běžně dostupné u celé řady vzdělávacích agentur, v rámci finančních i časových úspor je možno využít e-learningu na nosičích CD-ROM nebo na platformě Internetu (portál).

Jednou z cest je mezinárodně uznávaná, objektivní, standardizovaná metoda pro ověření počítačové gramotnosti European Computer Driving Licence (ECDL). Pomocí praktických testů zjišťuje, zda je uchazeč schopen efektivně využívat základní informační technologie (v České republice například zdroj [19]).



Obr. 1. Certifikát ECDL ve formátu kreditní karty

Projekt ECDL vznikl jako reakce na problémy spojené s prudkým rozvojem informačních technologií, které si vynutily požadavek definovat pojem počítačová gramotnost. Rozumí se tím objektivní minimum znalostí, které člověk potřebuje, aby mohl efektivně využívat informační technologie. Jednotný způsob testování a mezinárodně platných certifikátů

pro uživatele výpočetní techniky, který ECDL koncept představuje, se rychle stává standardem nejen v mnoha zemích Evropy, ale i v zámoří. Řada významných společností integrovala ECDL do vzdělávacích programů svých zaměstnanců.

Zaměstnavatelům ECDL slouží:

- K objektivnímu posouzení kvalifikace uchazečů o zaměstnání nebo svých stávajících zaměstnanců.
- K zhodnocení investic do výpočetní techniky a zajištění konzistentní úrovně IT znalostí v celé firmě.
- K zefektivnění práce ve firmě a snížení podpůrných nákladů na práci s výpočetní technikou.
- Ke zvýšení konkurenceschopnosti firmy na trhu.

Konkrétně ECDL znamená, že jeho držitel úspěšně složil jeden teoretický test ze základních pojmů informačních technologií a šest praktických testů, které osvědčují jeho znalost práce s PC a všeobecně rozšířenými aplikacemi.

1.3 Propojení kancelářských aplikací

Použití nejrozšířenějších aplikací jako jsou MS Excel nebo MS Word je snadné. Jejich účinné využívání už tak snadné není. Možnosti a funkce těchto aplikací zůstávají z velké části nevyužity, neboť uživatelé se spokojí se základními funkcemi, po jejichž zvládnutí získají dojem, že danou aplikaci umějí používat a další získávání znalostí v této oblasti považují za ztrátu času.

Často se pak stane, že zběhlí uživatelé MS Wordu neznají styly, záložky nebo automatické seznamy. Zasvěceným uživatelům MS Excelu zase zůstávají skryty funkce, matice nebo dokonce kontingenční tabulky.

Celý programový balík MS Office je považován za elementární znalost PC, ale v praxi je tato znalost degradována na psaní textu a výpočty v tabulkách. Logickým důsledkem je neefektivní využití možností výpočetní techniky a rozčarování při podrobnějším průzkumu úspor získaných nákladnými investicemi do informačních technologií.

Nejobtížnější úkol je rozpoznat optimální aplikaci pro daný typ úlohy. Nejméně zkušeností přitom mají uživatelé se "seznamy". Je telefonní seznam spíše text, nebo tabulka? Pokud je

jednoduchý, může to být úkol pro MS Excel. Bohužel v žádném návodu se uživatel nedočte, jak má takový seznam strukturovat, aby bylo možné jednoduše vyhledávat, třídit a řadit nebo dokonce k datům přistupovat automaticky.

Pokud se skutečně podaří data uložit podle zásad tvorby databází, otevře se možnost uživatelsky připravit zpracování dat hromadnou korespondencí. Na základě šablon pak může docházet ke generování automatických dokumentů, které budou určené pro tisk, nebo ještě lépe pro rozesílání elektronickou poštou.

Další výrazné usnadnění práce s delšími texty přináší využití záložek a odkazů mezi dokumenty, nebo jejich částmi. Užitečnou funkcí je OLE (Object Linking and Embedding). Jde o techniku dynamického propojování objektů. Například v tabulkovém kalkulátoru je možno vytvořit tabulku, která je následně zahrnuta do textového dokumentu. Místo obvyklého vložení je možno využít propojení. Při prohlížení a tisku dokumentu není znát rozdíl, ten se projeví až v případě změny původní tabulky. V takovém případě dojde ke změně i v textovém dokumentu. Tabulku je dokonce možno editovat přímo z textového editoru, pak je volána pro editaci mateřská aplikace.

Celá řada dat, která jsou shromažďována, jsou společná pro skupiny uživatelů. v takovém případě je víc než logické tato data společně zadat, udržovat na jediném místě a umožnit každému členu skupiny pohodlné využití. K takovému způsobu práce už je vhodné využití databází.

Zavedením výše popsaných možností do běžných pracovních postupů přinese jednoznačně produktivnější využití výpočetní techniky v kombinaci s kvalitativně vyšší úrovní práce obsluhy (Další možnosti uvedeny ve zdrojích [7] a [8]).

1.4 VBA (Visual Basic for Application)

MS Office přitom nabízí více než pouhé aplikace. K dispozici jsou také výkonné vývojové nástroje. Každý produkt MS Office obsahuje vestavěný programovací jazyk, který umožňuje přizpůsobit aplikaci tak, aby splnila specifické pracovní požadavky. V praxi to znamená namísto vytváření specifického software, přizpůsobení existujících hotových aplikací MS Office. Jednoznačnou výhodou pro uživatele je známé prostředí a snadné průběžné úpravy systému. Dále jsou popsány některé výhody vývojových nástrojů založených na MS Office (Podrobněji popsáno ve zdroji [12]):

Zrychlený vývoj

- Rozšíření stávajících funkcí – nemusí se začínat od počátku.

Komponenty programování

- Využití aplikací MS Office jako komponenty jiných aplikací.

Snadné použití

- Uživatelé tyto aplikace dobře znají, nepotřebují nákladný výcvik a podporu.

Nízké náklady

- Oproti vývoji zcela nového SW, který bývá velmi nákladný.

Pracovní pružnost

- Snadná úprava a upgrade z důvodu změn pracovních úkolů a nároků na SW.

Vyzkoušené nástroje

- MS Excel od r. 1987, MS Word 1990, MS Access 1992, MS PowerPoint 1995.

VBA je společným programovacím jazykem Microsoft Office – používá Automatizaci pro programové ovládání objektů. Je to jádro, které každý produkt MS Office rozšiřuje tak, aby umožnil přístup ke svým vlastním speciálním vlastnostem. Vývojář ve VBA musí zvládnout nejen jádro, ale také knihovny jednotlivých aplikací, aby jich dokázal co nejlépe využít.

Vlastní programování pak probíhá v samostatné aplikaci nazvané Editor Visual Basic, kterou je možné spustit z libovolné aplikace MS Office. Jedním z nejsnazších způsobů pro pochopení použití Editoru Visual Basic je záznam makra, jeho spuštění a úprava. Takto zaznamenané makro může pomoci rychle najít hledaný příkaz. Prakticky použitelná aplikace musí dále obsahovat proměnné, struktury řízení, cykly a ošetření chybových stavů.

Vlastnosti VBA v MS Office:

- Plný přístup k příkazům hotových aplikací.
- Vestavěný textový editor pro psaní kódu.
- Možnost odladění.
- Přizpůsobení standardního rozhraní hotové aplikace přidáním nebo odstraněním příkazů, položek nabídek, rolovacích lišt apod.
- Používání programového rozhraní aplikace Windows (API) a vlastních dynamicky propojených knihoven (DLL).

Vývojové nástroje MS Office nejsou určeny k nahrazení jazyka C, pro mnoho pracovních aplikací určených pro pracovní skupiny však tato platforma přináší efektivnější a pružnější řešení.

Pro mnohé programátory není snadné smířit se s přechodem aplikace pod MS Office. Z historického hlediska byl BASIC používán začátečníky a pro vytváření malých aplikací, zatímco jazyk C byl vyhrazen vývoji složitějších aplikací. Nyní je situace odlišná. BASIC se ve VBA změnil na programovací jazyk s dostatečnou strukturou a rychlostí, aby byl schopen podporovat několik tisíc řádků programu. Jinými slovy, VBA je nyní dostatečně silný, aby mohl podporovat rozhodující pracovní aplikace v určitých odděleních, týmech nebo pracovních skupinách. Nespornou výhodou programování ve VBA je dostupnost a nenáročnost potřebných nástrojů. Kdokoli může na své úrovni vylepšovat nebo přizpůsobovat aplikace svým potřebám. Vývojáři s tradičním programovacím zázemím mají stále tendenci podceňovat vývoj pracovních aplikací ve VBA.

Při vývoji aplikací pomocí nástrojů založených na MS Office vzrůstá počet koncových uživatelů se znalostí problematiky daného oboru, kteří se mohou podílet na programování. Mohou tak ovlivňovat vytvářenou aplikaci ještě ve fázi vývoje (inspirováno zdrojem [12]).

Tab. 1. Výhody a nevýhody obou přístupů k programování

	Tradiční programátoři	Zkušení uživatelé
Pro:	Mají zkušenosti a metodologický přístup pro vytváření robustních pracovních aplikací.	Rozumí pracovním problémům.
	Mohou si rychle osvojit jazyk ve stylu BASIC.	Mohou připravit jasnější programové ovládání desktop aplikace.
	Pokud mají zkušenosti s vývojem ve Windows a v C++, pochopí způsob, jakým Windows API rozšiřuje schopnosti vývojových nástrojů MS Office.	Znají pracovní postup a mohou vytipovat příležitosti pro kombinaci krokování a automatizování úloh nejlépe, což může ovlivnit rozhodování o nevhodnějším nástroji pro určitou úlohu.
Proti:	Protože nejsou nezbytně zkušenými uživateli MS Office, nemusí pochopit možnosti nejlepšího přizpůsobení těchto aplikací.	Mají sklony k tomu, aby byli fanoušky té aplikace, kterou znají nejlépe.
	Nerozumí pracovním činnostem, které tyto aplikace podporují.	Musí se naučit programovat a musí se naučit metodologickému uvažování při vývoji robustních aplikací.

Pro seriózní vývoj na základě aplikací MS Office je nezbytné sladění práce programátorů a uživatelů. Programátoři se musí seznámit s používáním aplikací MS Office na sofistikované úrovni, protože znalost programování nenahrazuje znalost hotového produktu, ve kterém programují. Zkušení uživatelé se musí naučit používat standardní metody a správné programovací návyky.

1.5 API (Application Programming Interface)

Aplikační programové rozhraní, sada rutin, které aplikace používá pro vyžádání služeb na nižší úrovni. API umožňuje každému použití "kouzelné skříňky" Windows bez toho, že by musel znát všechno, co je uvnitř ukryto. Volání API funkcí je možné ve všech obvyklých programovacích jazycích. v případě použití uvnitř VBA je velmi výhodné uzavřít funkci do rutiny VBA sloužící jako obálka pro volání API. Obálka obsahuje všechny potřebné deklarace rutin API i proměnných, takže potom stačí volat tuto obálku jako libovolnou funkci či proceduru VBA. Velmi praktické je v takovýchto případech funkci přejmenovat (ALIAS) tak, aby nedocházelo k "duplikaci procedur" při volání této funkce.

```

1 Declare Function wu_NetAddConnection Lib "mpr" Alias _
2   "WNetAddConnectionA" (ByVal NetPath$, ByVal Password$, _
3   ByVal LocalDrive$) As Integer

```


Použití takových funkcí musí být deklarováno příkazem *Declare*. Tento příkaz vyžaduje specifikovat název funkce nekolidující s jinými klíčovými slovy (*wu_NetAddConnection*), umístění knihovny funkcí (zde soubor *mpr.dll*), argumenty (*NetPath*, *Password*, *LocalDrive*) a typ, který funkce vrací (*Integer*).

Volání API funkce je zapouzdřeno ve standardní funkci VBA. v tomto případě přijímá funkce všechny parametry, které vyžaduje API a výsledkem je návratová hodnota, na základě které může být rozhodnuto o dalším běhu programu. Parametry funkce mohou být zadány pevně, vyhledány v tabulce, nebo je může volit uživatel formou dialogového okna.

```
4 Private Function PripojDisk(Cesta As String, Heslo As String, Pismo-
  no As String) As Boolean
5     Dim lngResult As Long
6     lngResult = wu_NetAddConnection(Cesta, Heslo, Pismo)
7     If lngResult = 0 Then
8         PripojDisk = True
9     Else
10        PripojDisk = False
11    End If
12 End Function
```

Nejčastější typy úkolů využívající API funkce jsou práce s disky, hledání aplikace, s níž je soubor asociován, nalezení systémových složek, zobrazení jména přihlášeného uživatele a název PC, atd. Použití takto zapouzdřené funkce v programu je pak velmi snadné.

```
13 If PripojDisk("\\NasDisk\Pokus", "Mojeheslo1", "K") Then
14     '-- pokračuj
15 Else
16     '-- neuspesne pripojeni - storno
17 End If
```

Funkcí API je ve Win32 API více než 1800. Bohužel nemůže VBA využívat všech výhod některých funkcí API, které využívají zpětné volání, neboť VBA toto nepodporuje. Přes velký počet funkcí API samozřejmě nemůže nabídnout všechno, univerzálním řešením je zde napsání vlastních knihoven DLL, nebo jejich modifikací XLL pro MS Excel a WLL pro MS Word. Pomocí těchto knihoven je možno přidat libovolné funkce, jejichž použití je velmi efektivní, neboť jsou připraveny v programovacím jazyku nízké úrovně, například C++.

DLL (Dynamic Linked Library) jsou dynamicky propojené knihovny procedur. Aplikace tyto procedury při běhu podle potřeby využívají, jako protiklad jejich uložení v době překladu nebo propojení. To znamená, že dynamicky propojené knihovny mohou být nezávislé na aplikaci, která jejich procedury používá, aktualizovány. Další výhodou knihoven DLL je to, že do doby, než jsou použity, nevyžadují žádný prostor v paměti (více příkladů ve zdroji [1]).

1.6 Opakovaně použitelné procedury

Struktura programu ve VBA je daná programátorem. Pro vyšší přehlednost je možné využít komentování kódu a velmi praktické je také vytvoření společných modulů funkcí a procedur, které jsou spouštěny z několika míst opakovaně.

V dalším textu budou naznačeny možnosti opakovaného využívání kódu v aplikacích MS Access, kde se nejčastěji používané příkazy týkají pohybu v aplikaci, některé formuláře obsahují dokonce pouze takové rutiny. Pokud jsou převedeny na společný modul, pak může být formulář bez modulu kódu, což zvýší rychlost jeho načítání.

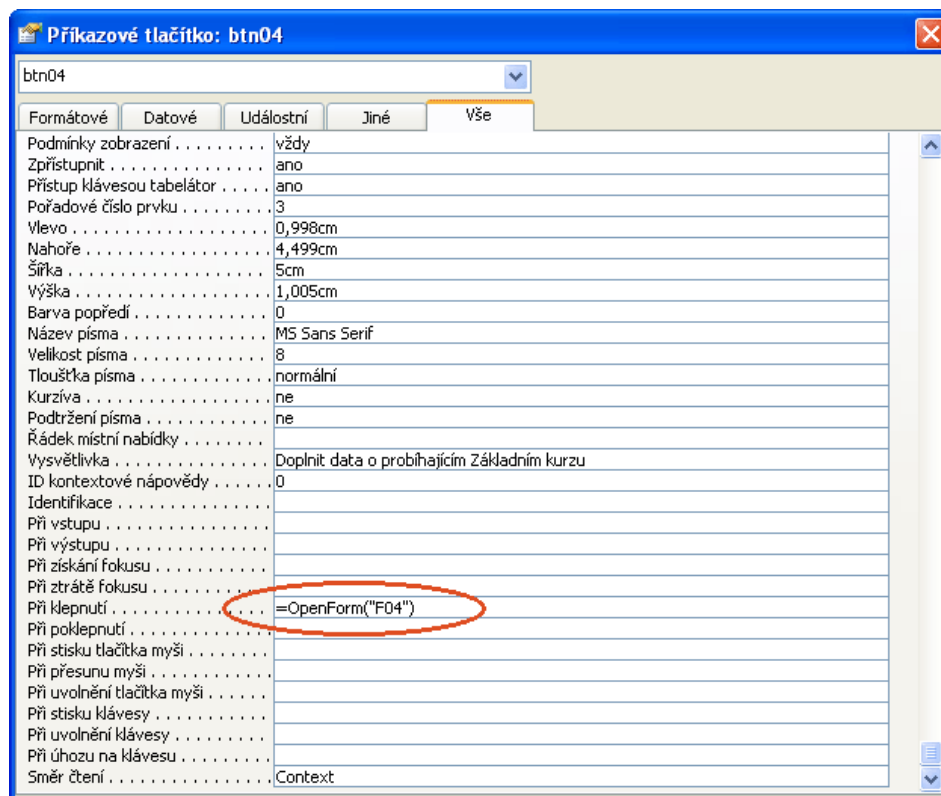
Ukázky jsou maximálně zjednodušeny (např. bez ošetření chyb), aby vynikla podstata, kterou je vytvoření tzv. obálky pro volání z kteréhokoliv místa aplikace (inspirováno zdroji [1], [9], [13], [14], [15] a [20]).

1.6.1 Pohyb mezi formuláři

Nejjednodušším příkladem univerzální procedury použitelné v libovolné aplikaci může být tato funkce volaná po stisku tlačítka, která umožní otevření formuláře.

```
1 Public Function OpenForm(strForm As String)
2     DoCmd.OpenForm strForm
3 End Function
```

Tato funkce je umístěna ve standardním modulu, její volání pak nemusí být v kódu, ale pouze z karty vlastnosti – událost "*Při klepnutí*" s argumentem název formuláře



Obr. 1. Ukázka karty vlastnosti tlačítka s voláním funkce k otevření jiného formuláře

Identickým způsobem může být řešen náhled nebo tisk sestav (reportů).

1.6.2 Nastavení titulku formuláře

Při otevření formuláře může být nastaven text jeho titulku. Toto řešení je výhodné v případě využití jednoho formuláře pro více účelů, kdy je vhodné mít pokaždé jiný titulek. V následující ukázce je využito vložení informací o aktuální verzi aplikace, které se nalézá v tabulce "T99Aplikace".

```

1 Public Function SetCaption(Formular As Form)
2   Dim Nazev As String
3   Dim Verze As Double
4   Nazev = DLookup("[AplikaceNazev]", "T99Aplikace")
5   Verze = DLookup("[AplikaceVerze]", "T99Aplikace")
6   Formular.Caption = Formular.Caption & " " & Nazev & " " & Verze
7 End Function

```

Volání této funkce je při načítání formuláře do paměti. Toto je možno zajistit umístěním následujícího příkazu do karty vlastností formuláře – událost "Při zavedení".

```

8 =SetCaption (Formuláře.NazevFormulare)

```

1.6.3 Export tabulek, dotazů a sestav

Pro přiblížení dat uživatelům a umožnění dalšího zpracování nad rámec systému je vhodné využít možností exportu do dokumentů MS Office. Podle typu výstupu se jeví jako nevhodnější export dotazů a tabulek do MS Excelu a reportů do MS Wordu. Kombinací zdrojů a exportních formátů je zde celá řada, proto je vhodné přednastavit nejpoužívanější kombinace do společného modulu kódu a dále je využívat. Jedna z častých variant je popsána v následující funkci.

```

1 Public Function ExpDotazuDoXLS(Dotaz As String, Cesta As String)
2     DoCmd.OutputTo acOutputQuery, Dotaz, acFormatXLS, CestaSoubor
3 End Function

```

Tato funkce uloží zadaný dotaz do souboru *.xls. Volání této funkce z karty vlastností je možné následujícím příkazem.

```
4 =ExportDotazuDoXLS(Dotaz,Cesta)
```

Znak rovnítko před názvem funkce se používá pouze na kartě vlastností. Jeho význam je rozlišení vlastní funkce od makra. Při volání funkce z modulu VBA se znak rovnítko nepoužívá.

Pro export je možno vybrat některý z následujících formátů: ASP, DAP, HTML, IIS, RTF, SNP, TXT, XLS. Zvolený typ je nutno zadat do konstanty *acFormat???* funkce *OutputTo*.

Tab. 2. Konstanty parametrů příkazu *DoCMD.OutputTo*

Konstanta	Hodnota	Databázový Objekt
acOutputDataAccessPage	6	Datová stránka
acOutputForm	2	Formulář
acOutputFunction	10	Funkce
acOutputModule	5	Modul
acOutputQuery	1	Dotaz
acOutputReport	3	Sestava
acOutputServerView	7	Pohled
acOutputStoredProcedure	9	Procedura
acOutputTable	0	Tabulka

1.6.4 Vlastní Zprávy zobrazované na základě tabulky

V průběhu práce s aplikací je velmi často vyžadováno odeslání informace uživateli formou dialogu nebo zobrazení informace. K tomuto účelu slouží funkce *MsgBox*. Její argumenty jsou:

MsgBox (zpráva[, tlačítka] [, titulek] [, nápověda, kontext])

Povinný argument je pouze "zpráva", což je text zobrazený zprávou. Celá řada situací vyžaduje zobrazení zprávy, ovšem text se často opakuje. Je tedy výhodné tyto texty uložit do tabulky spolu s indexem určujícím, která tlačítka mají být zobrazena. Informace o stisku konkrétního tlačítka je pak přenesena formou návratové hodnoty funkce.

```
1 Public Function ShowMsg(MsgNo As Long, Pip As Boolean) As Integer
2 Dim rsSeznamZprav As New ADODB.Recordset
3 With rsSeznamZprav
4     '-- Nacti pozadovane hlaseni
5     .Open "Select * from T93VlastniZpravy where MsgNo = " & _
6         MsgNo, CurrentProject.Connection, adOpenStatic
7     '-- Pokud je zapnuto Pip, pak pipni
8     If Pip Then
9         Beep
10    End If
11    '-- Zobraz vlastni hlaseni
12    ShowMsg = MsgBox(!MsgText, !MsgType + !MsgButtons, !MsgTitle)
13 End With
14 End Function
```

Volání takové funkce z modulu VBA je pak velmi jednoduché. Stačí uvést číslo zprávy a příznak, jestli má dojít při zobrazení i ke zvukovému signálu.

```
15 ShowMsg 2, False
```

Na základě tohoto příkazu bude zobrazena zpráva č. 2, a nebude použito akustického signálu. V tabulce zpráv je mimo vlastní zprávy a její titulek uložena informace o podobě zprávy. Tuto informaci nese jediná hodnota, která je součtem konstant vyjadřujících všechny parametry.

Tab. 3. Konstanty pro parametry zprávy

Konstanta	Hodnota	Popis
vbOKOnly	0	OK
vbOKCancel	1	OK – Storno
vbAbortRetryIgnore	2	Přerušit – Opakovat – Přeskočit
vbYesNoCancel	3	Ano – Ne – Storno
vbYesNo	4	Ano – Ne
vbRetryCancel	5	Opakovat – Storno
vbCritical	16	Ikona Křížek
vbQuestion	32	Ikona Otazník
vbExclamation	48	Ikona Vykřičník
vbInformation	64	Ikona Informace
vbApplicationModal	0	Aplikace je do odpovědi uživatele nepřístupná
vbSystemModal	4096	Systém je do odpovědi uživatele nepřístupný

Pro odpověď slouží návratová hodnota funkce, která je závislá na uživatelem zvoleném tlačítku. Takto je možno pokračovat v běhu programu podle přání uživatele.

Tab. 4. Návratové hodnoty po stisku tlačítka

Konstanta	Hodnota	Zvolené tlačítko
vbOK	1	OK
vbCancel	2	Storno
vbAbort	3	Přerušit
vbRetry	4	Opakovat
vbIgnore	5	Přeskočit
vbYes	6	Ano
vbNo	7	Ne

Jednoduchá rozhodovací konstrukce pro využití návratové hodnoty pak může vypadat například takto.

```

16 If ShowMsg(8, False) = 2 Then      '2=tlacitko storno
17     '-- pokračuje pokud storno
18 Else
19     '-- pokračuje pokud neco jineho
20 End If

```

1.7 Využití automatizace

Výše popsané metody jsou prvním krokem pro pochopení efektivního využívání informačních technologií. Seriózní přístup k celé problematice je završen měřením efektivity práce na PC a specifikací požadavků na obsluhu. Vývoj naznačeným směrem je nevyhnutelným vyústěním konkurenčního boje, který se projevuje zaměřením na snižování nákladů, rychlost a maximální využití stávajících zdrojů.

Pro vývoj aplikací to znamená zapojení vlastních zaměstnanců, zkrácení doby cyklu upgradu a programy přizpůsobené konkrétním potřebám uživatelů.

2 BEZPEČNOST DAT

Platí, že nejcennějším majetkem podniku jsou data a v nich uložené informace. Data se v drtivé většině případů ukládají do databázových systémů. Z toho vyplývá, že zabezpečení databází by mělo být integrální součástí informační bezpečnosti každé firmy. Bezpečnost bývá při hodnocení a srovnávání databázových platform považována za jednu z nejdůležitějších oblastí. Na co ovšem z pohledu bezpečnosti poměrně často zapomínáme, je lidský faktor. Záměrem je poukázat na nejběžnější a přesto stále opomíjené bezpečnostní mezery, jejichž odstraněním položíme základ pro zodpovědné zacházení s informacemi.

2.1 Právní aspekty

Právně je v České republice problematika ochrany informací řešena zákonem 101/2000 Sb.¹, který vychází ze Směrnic EP 95/45/ES² a 2002/58/ES³. Výkon záměru tohoto zákona včetně dozoru nad jeho dodržováním má od poloviny roku 2000 na starosti Úřad pro ochranu osobních údajů.

Uvedený zákon obsahuje celou řadu pravidel, které je nutno v rámci ochrany osobních údajů dodržovat – od registrace subjektů zpracovávajících osobní údaje až například po likvidaci údajů. Zpracování údajů je přitom definováno jako jakákoli operace či soustava operací systematicky prováděných správcem či zpracovatelem s osobními údaji, bez ohledu na to, zda se jedná o automatizované či ruční zpracování.

Poměrně široce je vymezen samotný pojem zpracování, pod kterým se rozumí nejen samotné zpracování, ale také shromažďování, ukládání a uchovávání, šíření a předávání či výměna údajů. Zjednodušeně řečeno tak pojem zpracování zahrnuje prakticky všechny možné operace s údaji, které je možné si představit. Z hlediska databázových technologií

¹ Zákon 101/2000 Sb. o ochraně osobních údajů a o změně některých zákonů.

² Směrnice Evropského parlamentu a rady 95/45/ES o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a s volným pohybem těchto informací.

³ Směrnice Evropského parlamentu a rady 2002/58/ES o zpracování osobních údajů a ochraně soukromí v odvětví elektronických komunikací.

je přitom zajímavá především ochrana v rámci ukládání, uchovávání a zpřístupňování údajů.

V souvislosti s ochranou osobních údajů je zákonem citlivý údaj definován jako osobní údaj vypovídající o národnostním, rasovém nebo etnickém původu, o politických postojích, o členství v politických stranách, o náboženství, o sexuálním životě apod. V širší souvislosti lze za citlivá data v souvislosti se zabezpečením považovat jakákoli data, která jsou natolik důležitá, že stojí za ochranu.

Pravidla pro ochranu osobních údajů jsou tedy v rámci našeho právního řádu poměrně přesně vymezena, je třeba si ovšem uvědomit, že se opravdu jedná jen o pravidla. Technické řešení v případě využití informačních technologií již zůstává na samotném systému, respektive jeho tvůrcích a uživatelích.

Z předchozího odstavce vyplývá jedna mnohdy velmi opomíjená skutečnost. Problematiku zabezpečení dat⁴ je nutno rozdělit minimálně na dvě základní skupiny – organizační a technickou.

2.2 Rizika a nedostatky

Pro zabezpečení dat je nutné znát jejich cenu, dokázat ohodnotit rizika a mít ochotu investovat do protiopatření. Bezpečnost lze definovat jako zajištěnost proti hrozbám, minimalizaci rizik a komplex administrativních, technických, logických a fyzických opatření pro prevenci a detekci neautorizovaného využití dat. I z tohoto důvodu je nutné si vymezit rámec, který má na bezpečnost dat zásadní vliv, kde bezpečnost v informačním prostředí lze zjednodušeně rozdělit na následující oblasti: (Informačním zdrojem byl článek [24])

Fyzická integrita

- Ochrana před technickými závadami, odolnost proti výpadkům napájení, zotavení z poškození.

Logická integrita

- Musí být zachována struktura dat a vzájemné vazby.

Elementární integrita

⁴ dále nás již nebudou zajímat jen osobní údaje definované zákonem, ale všechna zpracovávaná data

- Data obsažená v jednotlivých položkách jsou korektní.

Autentizace

- Každý, komu je povolen přístup musí být pozitivně identifikován.

Autorizace

- Kdo co může s čím dělat.

Auditabilita

- Kdo a jak přistupoval k položkám v databázi.

Komunikační bezpečnost

- Ochranu přenášených dat a zamezování nežádoucího datového provozu.

2.2.1 HW – Hardware ⁵

Do této skupiny patří veškeré technické vybavení sítě, objekty typu počítače, periferie, kabely, telekomunikační linky, obvody či jakákoliv jiná technická zařízení připojená k síti. Každý z těchto objektů může být ohrožen poškozením či zničením. Rozsah možnosti ohrožení hardwaru je téměř neomezený (krádež, přerušení kabelů, fyzické zničení zařízení, neoprávněné zapojení komponent atd.). Tady je nutná pravidelná technická kontrola všech objektů v síti, fyzické zabezpečení hardwaru proti krádeži, používání ochranných jističů a dalších zabezpečovacích zařízení.

2.2.2 SW – Software

Mezi SW může být zařazeno veškeré programové vybavení pracovních stanic a serverů. Jsou to operační systémy, aplikace a jiné programy nutné pro provoz sítě. Data budou zařazena do jiné oblasti. Softwaru hrozí následující: Smazání, krádež, napadnutí virem, trojským koněm nebo parazitem, mohou mu hrozit skryté chyby, které se projevují po určitém běhu programu a za určitých okolností. Síťové správní programy mohou umožnit sledování pokusů o smazání nebo poškození programů, antivirové programy mohou zabezpečit software proti virům.

⁵ Jakákoliv kategorizace je nepřesná, zde využito členění na HW, SW, DW, PW a OW dle zdroje [6].

2.2.3 DW – Dataware

V této oblasti jde o uložená data a vazby, která jsou potřebná pro zpracování informací. K ohrožení může patřit: Úmyslné smazání databáze, krádež, ztráta dat během havárie sítě, poškození dat. Některé druhy poškození informací mohou být odhaleny pomocí cyklických nadbytečných kódů nebo dalších metod detekce chyb. Pokusy o smazání datových souborů mohou být detekovány některými řídicími nebo antivirovými programy.

2.2.4 PW – Peopleware

Zabezpečení proti výše uvedeným rizikům je mnohdy náročné, ale dá se systémově řešit. Kde systémová opatření obvykle nefungují tak snadno, jsou uživatelé. Za největší bezpečnostní slabinu v rámci informačních systémů bývá oprávněně považován lidský faktor. Hesla k uživatelským účtům napsaná na lístečcích nalepených na monitoru nejsou ani dnes ničím výjimečným. Ale tím nebezpečí lidského faktoru zdaleka nekončí.

Nepřehledný systém záloh často znamená odstranění nejaktuálnější verze, nebo její nechtěné přepsání verzí starší. Obyčejný překlep dokáže snadno znehodnotit data i program a obvykle se velmi obtížně hledá. Nezabezpečené kopie choulostivých dat se mohou snadno dostat k nepovolaným osobám. Nedůslednost v praktickém naplňování bezpečnostních pravidel může otevřít cestu k nežádoucímu útoku. Nabourání se do informačního systému, či přesněji databázi, může být nepříjemné, zejména v případě, kdy průlom vůbec není objeven. Mnohé z útoků bývají nedestruktivní a mající za účel pouze získat data.

2.2.5 OW – Orgware

Organizační opatření jsou velmi důležitá a souvisí úzce s chováním uživatelů. Nezodpovědné chování uživatelů není možné úplně vyloučit. Jasná a pokud možno jednoduchá pravidla, se kterými je uživatel seznámen, však výrazně omezí možnosti nežádoucích průniků. Jsou však společnosti, které si plně problematiku informační bezpečnosti neuvědomují, včetně případných dopadů a důsledků na fungování samotné společnosti. Celá řada společ-

ností se zaměřuje na firewall⁶ s přesvědčením, že chránit interní systémy před vlastními pracovníky je zbytečné, případně to není na pořadu dne.

Mnohdy ani neexistuje povědomí o případných ztrátách nebo dopadech, které může vyvolat nekontrolovatelné jednání pracovníka, počínaje smazáním důležitých souborů, odcizením dat a jejich prozrazením konče. Zejména nemožnost sledovat konání pracovníka a zasáhnout v případech, kdy překračuje rámec svých pravomocí, je limitujícím faktorem informační bezpečnosti nejen v menších, ale i ve velkých společnostech.

V neposlední řadě jsou zde také nedostatky procedurálního charakteru, tzn. v metodách, pomocí kterých je nakládáno s informacemi a pravomocemi ze strany samotných pracovníků. Nutnost odpovídajícího vzdělávání vlastních uživatelů, případně politiky přístupu k interním informacím s vyjádřením odpovídajících penalizací a postihů při jejich porušení je tedy nezbytnost, bez níž bude řada bezpečnostních mechanismů i nadále neúčinná.

2.3 Přístup k databázi

Moderní databázové technologie z hlediska zabezpečení dat nabízejí celou řadu možností – od šifrovaného uložení dat přes řízení přístupu k datům až po zabezpečený přenos dat ze serveru na stranu klienta. Z pohledu uživatele je přitom důležité, aby jej bezpečnostní omezení při běžném provozu příliš neomezovala – pokud je to možné, měla by veškerá oprávnění daného uživatele vycházet ze způsobu jeho přihlášení k danému systému.

Přestože se jednotlivé databázové platformy v možnostech zabezpečení dat značně liší, mnohé základní přístupy zůstávají shodné či podobné. Podobnost z hlediska uživatele se týká zejména řízení přístupu k datům – od identifikace uživatele přes jeho autentizaci až po autorizaci zajišťující kontrolu oprávnění daného uživatele k provedení určité operace s daty (k tomuto tématu více ve zdroji [3]).

⁶ Firewall (bezpečnostní filtr pro komunikaci s okolím) je pochopitelně důležitá součást zabezpečení. Na tomto místě jsou zdůrazněna méně nápadná interní rizika.

2.4 Autentizace

Nejrozšířenější formou autentizace je zadání uživatelského jména a hesla, která si ve většině případů mohou uživatelé sami zvolit. Zadáním této dvojice sdílených "tajemství" uživatel získá přístup k výpočetnímu systému, aplikaci, databázi, službě apod.

V současné době se velmi často používá sdružování jednotlivých uživatelů do skupin, kterým jsou přidělována jednotlivá oprávnění jako celku. Přístup přitom může být umožněn ke všem datům či k určité podmnožině dat – v případě některých platform je realizovatelné zabezpečení až na úrovni konkrétního záznamu či dokonce konkrétního atributu z daného záznamu.

Přiřazování uživatelů do skupin je přitom možné považovat za organizační záležitost – operátor pořizující skladová data o nakoupených zásobách přece nemusí mít přístup k ekonomickým informacím o zaměstnancích. Stejně tak je možné za samostatnou problematiku spadající do oblasti organizačních postupů považovat správu hesel.

Problémem autentizace pomocí hesla je to, že si uživatelé volí velmi často hesla triviální nebo taková, která je možno snadno uhodnout. Zarážející jsou hesla napsaná na monitorech a nezměněná přednastavená hesla v databázových platformách. To dává možnost útočníkům využít odhalených hesel k proniknutí do výpočetního systému nebo neoprávněně získat přístup k datům, aplikacím, službám apod. Při stanovení bezpečnostní politiky organizace by měla být pro uživatele podnikových serverů stanovena povinnost⁷ volit uživatelská hesla určitého stupně složitosti (složitost, délka platnosti či možnost opakování). Vhodné je prostřednictvím systémových prostředků nutit uživatele, aby po určité době svá přístupová hesla do systému měnil.

K autentizaci mohou být použity i speciální aplikace (například bezpečnostní či adresářové servery), hardwarová zařízení či služby operačního systému (je-li uživatel platně přihlášen do operačního systému, má umožněn přístup i k databázi). Vysoký stupeň bezpečnosti autentizace na bázi hesla je zajištěn speciálními přenosnými zařízeními v angličtině nazývanými token. Příkladem tohoto typu zařízení jsou smart cards (čipové karty) nebo iKeys (identifikační klíče). Pouze majitel tohoto zařízení zná "tajný klíč"⁸, který zařízení zpro-

⁷ Technické řešení je v tomto případě nesrovnatelně účinnější, než organizační předpis.

⁸ PIN (Personal Identification number) – jedinečné číslo.

vozní. Ztráta nebo odcizení zařízení bez současného vyrazení příslušného PINu neumožní tedy neoprávněnému novému vlastníku zneužít přístupová práva vlastníka původního – oprávněného.

Smart cards připomínají svým tvarem platební karty. Obsahují mikroprocesor a paměť pro uchování potřebných informačních a identifikačních dat. Pro jejich aplikaci je třeba instalovat v počítači uživatele mechanickou jednotku, která dokáže data uložená v paměti karty, přečíst. Na rozdíl od smart card nevyžadují iKeys zvláštní čtecí mechaniku. Je možno jej zasunout do standardního sériového konektoru typu USB. Z hlediska provozních nákladů je tudíž řešení verifikace uživatelů pomocí iKeys méně nákladné (více informací ve zdroji [21]).



Obr. 2. Smart Card a iKey

Administrátor by měl mít možnost oprávnění pro připojení k databázi nejen přidělit, ale také odebrat či časově omezit jeho platnost. Výhodou dále je, umožní-li SŘBD⁹ definovat akce, které se mají vykonat při nesplnění autentizačních podmínek (kontaktovat bezpečnostního správce, zablokovat uživatelský účet nebo odebrat oprávnění přístupu ke chráněným datům). Samozřejmostí by mělo být vygenerování záznamu do sledovacích protokolů (stejně jako v případě porušení ostatních bezpečnostních mechanismů). Nelze jednoznačně konstatovat, který způsob autentizace a reakce na selhání je nejlepší – vždy záleží na konkrétní situaci.

2.5 Autorizace

Další oblastí je zajištění zabezpečení provádění systémových akcí v databázi – vytváření, modifikace a rušení uživatelů, úložných prostorů, databázových objektů (tabulek, indexů atd.), spouštění a zastavování databáze či nastavování vlastností bezpečnostních me-

⁹ SŘBD (Systém řízení báze dat) – programové vybavení databázového systému, které vytváří uživatelské prostředí pro uživatele a aplikační programátory.

chanismů. V návaznosti na zvolenou bezpečnostní strategii bývají oprávnění na provedení těchto akcí rozdělena mezi více subjektů (nejčastěji mezi databázového administrátora, bezpečnostního správce a uživatele – v případě databázových objektů, jako jsou zmiňované tabulky, indexy).

Pokud konkrétní SŘBD neposkytuje dostatečné možnosti v řízení přístupu k datům, bývá nutné tento handicap řešit programově.

2.6 Audit

Při vyjmenovávání vlastností zabezpečení se poměrně často zapomíná na sledování jednotlivých událostí a jejich následnou analýzu (hovoříme o tzv. auditu). Do speciálních souborů (případně tabulek) – deníku auditu jsou ukládány tzv. auditní záznamy.

Běžně jsou tímto způsobem monitorovány přihlášení / odhlášení uživatele, informace o tom s jakými tabulkami pracoval, jaké operace prováděl a také informace o tom, jaký SQL dotaz byl spuštěn (informačním zdrojem byl článek [22]).

Auditní záznamy by měly obsahovat co možná nejpřesnější informace charakterizující danou událost – například v případě neoprávněného přístupu k datům uživatele, datum a čas, terminál (klientský počítač), identifikaci připojení, údaje o dalších připojeních stejného uživatele atd. Mnohé SŘBD umožňují určit, které události a kdy (při úspěchu či selhání) se mají sledovat. Podstatné je, aby byl deník auditu průběžně kontrolován – jinak mnohé průniky do systému ani nemusí být odhaleny.

Obdobně jako vlastní data je nutno i auditní záznamy považovat za velmi citlivé informace. SŘBD by měl také umožnit nadefinovat událost, která se vykoná při zapsání auditního záznamu daného typu (obdobně jako v případě nesplnění podmínek autorizace). Ideální ovšem je, pokud je sledování prováděno na více, než jednom článku systému – kvalitní útočníci po sobě dokážou zamést stopy a pokud jsou tyto informace k dispozici na více místech, mají situaci mnohem složitější.

Uživatel SŘBD může ovlivnit množství informací ukládaných do systémových tabulek nastavením vlastností pro audit systému. v případě úplného auditu systému musíme mít na zřeteli, že při velkém počtu uživatelů mohou být systémové tabulky velmi rozsáhlé, a proto je současně nutné nastavit pravidla pro archivaci, jejich uložení do jiné databáze

sloužící k účelům monitorování databáze. Může se samozřejmě také jednat o odmazávání dat z těchto tabulek, opět, v definovaném cyklu.

2.7 Aplikace bezpečnostních opatření

Zabezpečení dat je dnes ve vyspělých databázích řešeno komplexně a v kombinaci s dalšími prvky nemusí být vytvoření bezpečného informačního systému nepřekonatelným problémem, jak by se na první pohled mohlo zdát. Nezbytnou podmínkou je však proškolení uživatelů a správců systémů a následně důsledná a pravidelná kontrola dodržování stanovených pravidel. K průniku může dojít i do sebelépe zabezpečeného systému – zejména při selhání lidského faktoru či nedodržení bezpečnostních opatření na organizační úrovni. Ochrana dat pouze snižuje pravděpodobnost úspěchu útoku, tj. útočník například musí vynakládat nesrovnatelně více prostředků k dosažení cíle.

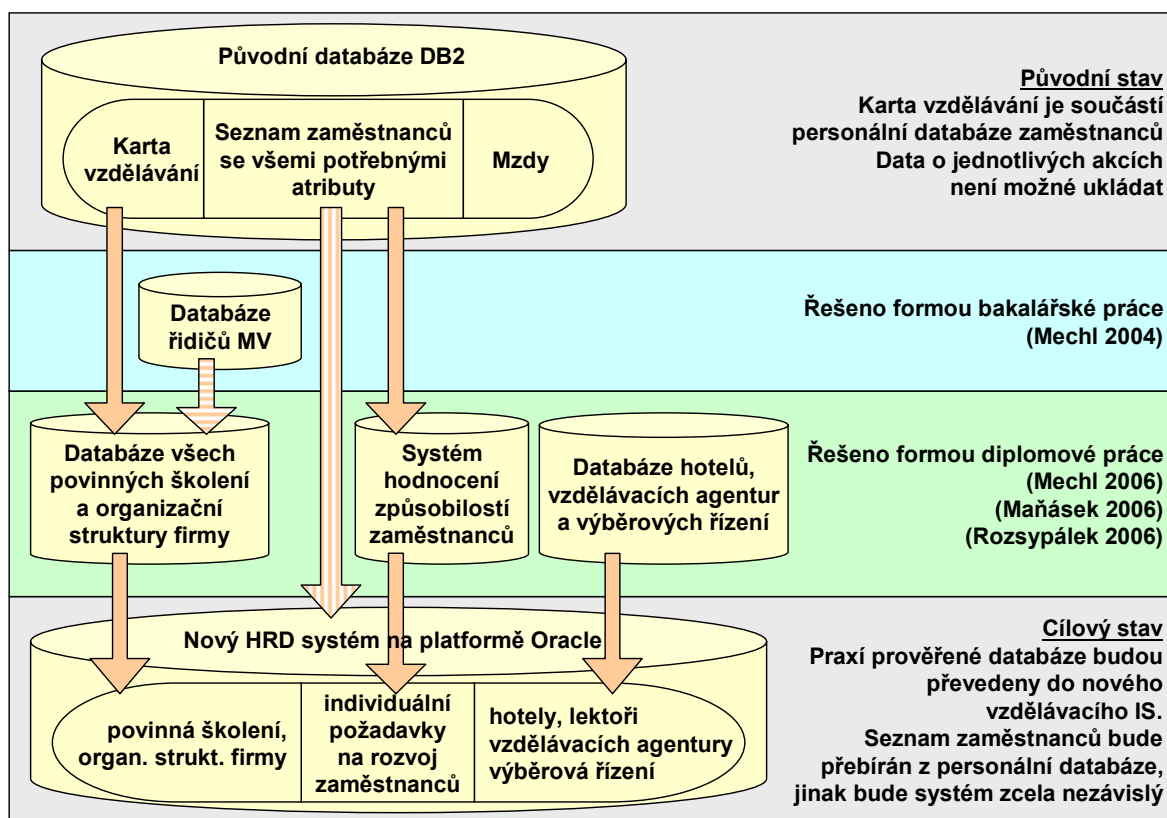
II. PRAKTICKÁ ČÁST

3 UPŘESNĚNÍ CÍLŮ A ZADÁNÍ PROJEKTU

V rámci bakalářské práce byla vybrána jedna z oblastí organizace povinných školení podle zákona. (podrobnosti ve zdroji [5]) Tato databáze byla prověřena praktickým provozem v multiuživatelském prostředí. Na základě těchto zkušeností bylo rozhodnuto o zahrnutí organizace všech ostatních povinných školení a rozšíření o další funkce. Uživatelé poznali výhody databázového řešení a přišli s návrhy zdokonalení funkčnosti, které považovali za prospěšné.

3.1 Projektový rámec

Podrobnější analýzou požadavků byl spuštěn projekt nového vzdělávacího informačního systému. Tento projekt byl rozdělen na tři části. První část staví na databázi zpracované formou bakalářské práce a zabývá se povinným školením a propojením dat se stávající databází zaměstnanců. Druhá část, která není součástí této diplomové práce, má za úkol zpracovat podklady pro výběrová řízení dodavatelských agentur, výběry hotelů a lektorů pro rozvojové vzdělávací kurzy. Třetí část pak zpracovává hodnocení způsobilostí zaměstnanců a stane se zdrojem požadavků budoucího vzdělávacího systému.



Obr. 3. Struktura a návaznost jednotlivých částí projektu

Všechny tři části projektu musí být zpracovány tak, aby je bylo možné po ověření funkčnosti snadno použít pro definitivní řešení informačního systému na platformě Oracle s uživatelskou nadstavbou ve Visual Basicu.

3.1.1 Současný stav

Personální divize používá databázi zaměstnanců na platformě DB2 pro informace o jednotlivých zaměstnancích. Součástí této databáze je "*Karta vzdělávání*", která byla přidána do systému pro evidenci absolvovaných školení. Záznamy jsou připisovány na karty zaměstnanců, což znamená, že každá účast na vzdělávací akci musí být zapsána každému zaměstnanci zvlášť (viz Obr. 3). Toto řešení znamená opakované zadávání dat a umožňuje chyby při zápisu. Významným omezením je chybějící možnost zapsat jakákoliv podrobnější data o absolvované akci. Plánování nákladů na vzdělávání a podrobnější analýzy tento systém neumožňuje. Chybí tedy významný prvek pro snadné řízení a vyhodnocování vzdělávání.

3.1.2 Cíle projektu

Cílem je vytvořit vzdělávací IS, který bude přehledněji poskytovat více informací při nižší časové náročnosti na obsluhu a umožní jednoduchou kontrolu nad průběhem celé akce. Základním prvkem bude vzdělávací akce, součástí bude integrovaný seznam nákladových středisek a zadávání dat bude zjednodušeno¹⁰. Systém umožní práci se skupinami zaměstnanců (obdoba hromadné korespondence).

3.1.3 Etapy projektu

1. Analýza stávajícího stavu a budoucích potřeb.
2. Standardizace současných postupů.
3. Vypracování vhodné struktury budoucí aplikace.
4. Nalezení vhodné podoby uživatelského prostředí.
5. Zaškolení uživatelů a zavedení systému do každodenní praxe.

¹⁰ Data jednou zadat a na mnoha místech opakovaně využívat.

3.1.4 Harmonogram

Následující diagram ukazuje rozdělení projektu na tři oblasti úkolů a tři milníky, které byly naplánovány jako kontrolní zastávky spojené s prezentací postupu a výsledků managementu zadavatele.

Zeleně vyznačené plochy již byly dokončeny, nebo jsou rozpracovány. Žluté pole jsou úkoly ve fázi příprav. Po dokončení této části projektu bude aplikace využívána k zaškolení nových uživatelů a rozšiřování dalších funkcí na základě stále přísnějších požadavků.

Tab. 5. Časová osa projektu

2006		17.3. 30.6. 31.8.											
		1	2	3	4	5	6	7	8	9	10	11	12
20	Příprava podkladů												
21	Úprava již existujících číselníků podle nové struktury	■											
22	Vytvoření nových zatím nepoužívaných číselníků		■										
23	Standardizace postupu organizace vzdělávací akce		■	■									
25	Shromáždění požadavků uživatelů			■	■								
30	Úpravy systému												
31	Rozšíření stávající databáze o nové funkce			■	■	■	■						
32	Testování funkčnosti vybraným uživatelem					■	■						
40	Ladění a testování												
41	Zaškolení uživatelů						■	■	■				
42	Vyhodnocení splnění požadavků zadání						■	■					
43	Přechod na platformu Oracle							■	■				

Harmonogram zobrazuje pouze zde popisovanou část projektu. Konkrétní termíny jsou vždy koordinovány v rámci všech tří projektů popisovaných v kapitole 3.1.

3.2 Databázová platforma

Základ testované databáze byl zpracován v systému MS Access (podrobněji zdůvodněno ve zdroji [5]). Tento systém je dostatečně uživatelsky přívětivý, je možno ho do jisté míry modifikovat podle aktuálních požadavků a také pro zamýšlený počet uživatelů je dostatečný.

Výhodou je jednoduchá spolupráce s ostatními produkty MS Office. Především export a import do MS Excelu a hromadná korespondence v MS Wordu. Díky tomu je možné využít stávajících formulářů. Výstupy pak mají jednotnou podobu a strukturu bez ohledu na to, který uživatel přehledy a sestavy právě tiskne. V případě nutnosti je ale možné individuálně upravit sestavy i formuláře pro každého uživatele jednotlivě. MS Access je součástí kancelářského balíku MS Office Professional a není tedy nutné přikupovat žádné nové licence.

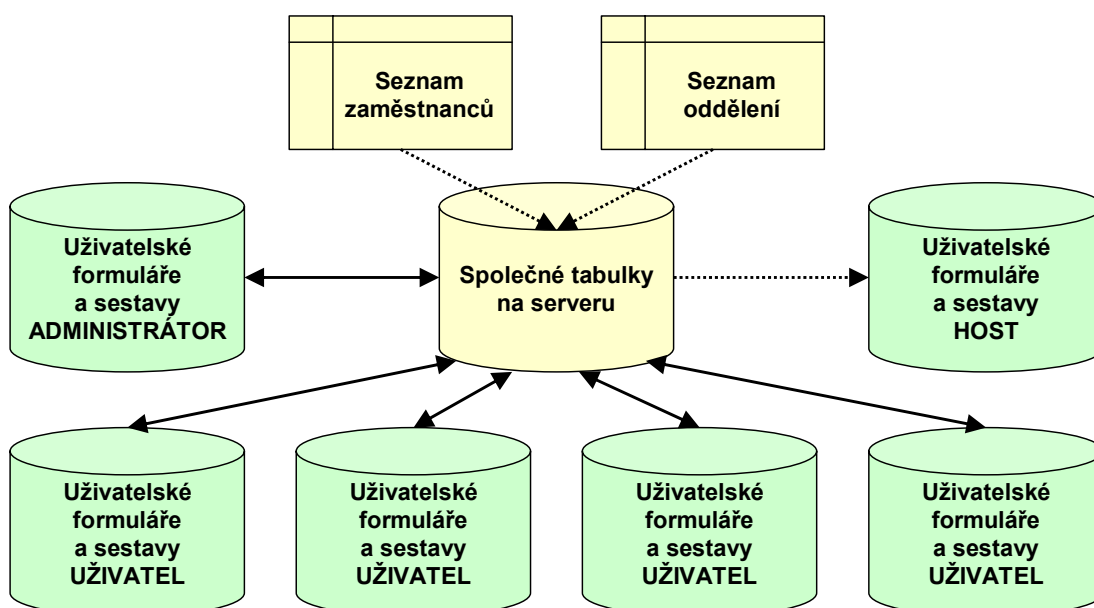
Je vhodné zdůraznit, že zpracované řešení je funkčním modelem pro nalezení optimální struktury budoucí databáze. Nespornou výhodou je pak skutečnost, že již zpracované moduly funkcí VBA bude možno v budoucnu snadno upravit pro verzi ve Visual Basicu. Tato skutečnost bude plně doceněna právě při přechodu na platformu Oracle, neboť uživatelské rozhraní bude zpracováno právě ve Visual Basicu.

4 AUTOMATIZACE ÚLOH

Z uživatelského hlediska největší změny nastaly vlivem automatizování úloh, které doposud uživatel prováděl sám mimo systém. Jedná se o nejrůznější výpočty a souhrny zpracovávané v jednoúčelových tabulkách MS Excel, přihlašování zájemců o kurzy probíhající telefonicky nebo papírovou podobou apod. V této kapitole jsou dále popsány univerzální procedury, které jsou využívány z více míst aplikace. Záměrem bylo minimalizovat vstupy uživatelů do chodu aplikace, tzn. dotazovat se pouze v případě, že program není schopen jednoznačně rozhodnout, kterou variantu uživatel požaduje. Automatizace se týká většiny úprav systému, tedy i zabezpečení, přístupu na intranet nebo propojení dat s informačním systémem. Podrobnosti o těchto úlohách budou v dalších kapitolách (čerpáno především ze zdrojů [1], [13] a [15]).

4.1 Kontrola přístupu k zdroji dat

Aplikace je od počátku konstruovaná jako víceuživatelská. Datová část je na serveru a každý uživatel má na svém lokálním PC umístěn soubor uživatelského rozhraní s formuláři, sestavami a tabulkami, které jsou v čase neměnné, nebo jsou pro chod uživatelského rozhraní nezbytné. Toto řešení znamená velkou komplikaci pro chod systému v případě nedostupnosti serveru, nebo poruchy sítě. v takovém případě by uživatel nebyl schopen aplikaci spustit.



Obr. 4. Struktura a umístění jednotlivých komponent aplikace

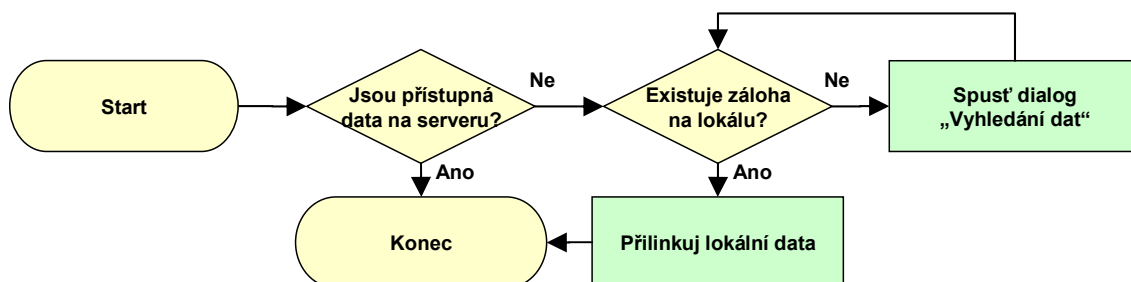
Eliminací tohoto nedostatku je kontrola dostupnosti zdroje dat, která probíhá při spuštění aplikace. Při události *load* úvodního formuláře, který je nevázaný na zdroj dat je volána funkce *RefreshLinks* s argumentem název datového souboru.

```

1 Private Sub Form_Load()
2     If RefreshLinks("Data.mdb") Then
3         '-- Pripojeni se podarilo, pokracuj
4     Else
5         MsgBox "Nepodařilo se připojit zdroj dat", vbOKOnly, ""
6         DoCmd.Quit acQuitSaveAll
7     End If
8 End Sub

```

Pokud není zdroj dostupný, dojde k automatickému připojení lokálního zdroje dat. Tento zdroj je kopií serverových dat, které byly uloženy při posledním zavírání aplikace tímto uživatelem. Data pak nejsou aktuální a do lokálních dat nemá význam cokoli zapisovat. Uživatel je na tuto skutečnost upozorněn při startu systému a v záhlaví formulářů se objevuje informace *"Data pouze pro čtení"*.



Obr. 5. Kontrola přístupnosti dat na serveru

V případě, že data na lokální stanici nebyla nalezena, je spuštěno standardní dialogové okno systému Windows pro vyhledání datové části aplikace. Teprve v případě nenalezení souboru dojde k zavření aplikace z důvodu nenalezení dat.

Vlastní funkce *RefreshLinks* obsahuje na počátku deklaraci proměnných a připojení k systémové tabulce *MSysObjects*, která obsahuje seznam všech objektů databáze. Důležitým prvkem je zde podmínka *WHERE*, která filtruje pouze objekty typu "6", což znamená propojené tabulky (názorně je vidět na řádce 8 kódu).

```

1 Function RefreshLinks(ZdrojTAB As String) As Boolean
2     Dim dbs As Databáze, tdf As TableDef, strDatabase As String
3     Dim rst As Recordset, rstTry As Recordset
4     Dim strOldConnect As String, strNewConnect As String
5     Dim strFullLocation As String, strMsg As String

```

```
6 '-- Nastav pripojeni do dbf
7     Set dbs = CurrentDb()
8     Set rst = dbs.OpenRecordset("SELECT MSysObjects.Connect, MsysOb-
    jects.Database, MSysObjects.Name from MSysObjects " & "WHERE MSysOb-
    jects.Type = 6")
```

Tímto krokem došlo k výběru propojených tabulek. Následující podmínka zjišťuje, zda existuje alespoň jedna taková tabulka, pokud ano dojde ke kontrole každé tabulky tím, že bude otevřena s pokusem přejít na její první záznam, pokud se to podaří, připojení funguje. Důležitý je zde pokus o načtení záznamu, samotné připojení tabulky totiž nemusí vyvolat chybu, i když tabulka nebyla nalezena. Dojde-li k chybě při načítání záznamu z tabulky, přejde procedura k další části, kde změní cestu ke zdroji dat na aktuální adresář. Následuje vynulování chybové hodnoty a opět kontrola připojení. V praxi je výhodné mít cesty k datovým souborům uložené v tabulce a vkládat je do parametrů funkce přes proměnné.

```
9     If rst.RecordCount <> 0 Then
10         rst.MoveFirst
11     Do
12         On Error Resume Next
13 '-- Pokud funguje, over vyberem prvnio zaznamu, pak zavri
14         Set rstTry = dbs.OpenRecordset(rst![Name].Value)
15         If Err = 0 Then
16             rstTry.Close
17             Set rstTry = Nothing
18         Else
19 '-- Pokud nefunguje pripoj soubor z tohoto adresare data
20             Set tdf = dbs.TableDefs(rst![Name].Value)
21             strOldConnect = tdf.Connect
22             strNewConnect = ";DATABASE=" & CurrentDBDIR & ZdroJTAB
23             For Each tdf In dbs.TableDefs
24                 If tdf.Connect = strOldConnect Then
25                     tdf.Connect = strNewConnect
26                     tdf.RefreshLink
27                 End If
28             Next tdf
29             dbs.TableDefs.Refresh
30         End If
31         Err = 0
32         rst.MoveNext
33     If rst.EOF Then
```



```

34             Exit Do
35         End If
36     Loop
37 End If

```

Na závěr funkce je vhodné "uklidit", tzn. vyprázdnit proměnné a vložit příznak pro návratovou hodnotu funkce signalizující, že celá operace proběhla bez chyb.

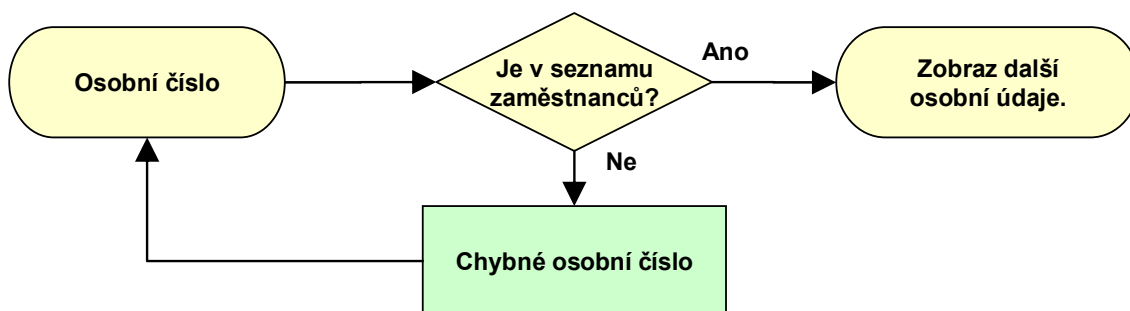
```

38 '-- Odstran veskere promemne
39 Set tdf = Nothing
40 Set rst = Nothing
41 Set rstTry = Nothing
42 RefreshLinks = True
43 End Function

```

4.2 Doplnění údajů o zaměstnanci

Každý zaměstnanec má přiděleno jedinečné osobní číslo. Toto je využíváno v řadě systémech pro evidenci namísto rodného čísla, jež lze považovat za citlivý údaj. Zaměstnanci toto číslo znají a velmi často používají. Příkladem mohou být prezenční listiny ze školení, kde je osobní číslo vždy uvedeno. V této aplikaci bylo využito osobních čísel jako klíčové položky pro identifikaci zaměstnance. Díky seznamu zaměstnanců pak stačí zapsat toto číslo, a následně funkce sama vrátí jméno a další osobní údaje o zaměstnanci.



Obr. 6. Doplnění osobních údajů s ověřením existence osobního čísla

Funkce je vázána na aktualizaci pole *Osobní číslo*. Tato akce spouští funkci, v jejímž těle je několikrát spuštěna funkce *DLookup*, která vyhledá v předem dané tabulce potřebný atribut záznamu. v případě neúspěchu vyprázdní pole a vypíše hlášení uživateli. Následující ukázka doplňuje pouze příjmení se jménem a adresou.

```

1 Private Sub txtOsobniCislo_AfterUpdate()
2 If Len(DLookup("[PrijmeniJmeno]", "T11Zamest", "[OsobniC] = " &
  Me.txtOsobniCislo)) > 0 Then
3     Me.txtPrijmeniJmeno = DLookup("[PrijmeniJmeno]", "T11Zamest",
  "[OsobniC] = " & Me.txtOsobniCislo)

```

```
4     Me.txtAdresa = DLookup("[Adresa]", "T11Zamest", "[OsobniC] = " &
    Me.txtOsobniCislo)
5 Else
6     Me.txtPrijmeniJmeno = ""
7     Me.txtAdresa = ""
8     MsgBox "Zaměstnanec s tímto osobním číslem neexistuje"
9     Me.txtOsobniCislo.SetFocus
10 End If
11 End Sub
```

Funkce *DLookup* je vhodná pouze pro malý počet vyhledávaných atributů. V opačném případě je vhodnější připojení k recordsetu podobným způsobem, jako je naznačeno dále při ukládání dat do tabulky.

Pokud uživatel rozhodne o dalším využití záznamu, může spustit proceduru, která tento záznam uloží do tabulky.

```
1 Private Sub btnUloz_Click()
2 Dim db As Database
3 Dim rs As Recordset
4 Set db = CurrentDb
5 '-- Otevri tabulku
6 Set rs = db.OpenRecordset("T41Objednavka")
7 '-- Zapis zaznam do tabulky
8     With rs
9         .AddNew           '-- Pridej zaznam
10            !OsobCislo = Me!txtOsobniCislo
11            !PrijmeniJmeno = Me!txtPrijmeniJmeno
12            !Adresa = Me!txtAdresa
13            .Update       '--Uloz zmeny
14            .Bookmark = rs.LastModified  '-- Bez na nový záznam
15        End With
16 rs.Close
17 End Sub
```

Tímto způsobem jsou do tabulky objednávek zapsány všechny důležité údaje, na základě kterých může být vygenerován požadavek směrem k lektorům. Další postup je již standardní tiskový výstup ve formě reportu nebo export do MS Excel.

4.3 Příprava podkladů nového kurzu

Při zakládání nových kurzů je možno využít faktu, že jde o předem známý okruh možností. Z tohoto důvodu je uživatelům nabídnut seznam typů kurzů, ze kterého si mohou vybrat. Po výběru je založen kurz se základními, předem známými atributy, a stačí doplnit podrobnosti nebo případně pozměnit údaje automaticky předdefinované. Uživatelsky ještě příjemnější je možnost duplikovat již existující kurz se všemi atributy včetně seznamu účastníků, což je užitečné především u opakovaných kurzů. V takovém případě stačí změnit datum a kurz je připraven.

Seznam účastníků školení je možno sestavit třemi různými způsoby podle okolností a typu kurzu.

Nový kurz

- Jednotlivé účastníky přihlašují jejich nadřízení, předem se nedá určit jejich počet.

Opakovaný kurz volně přístupný

- Základem je seznam zaměstnanců, jimž končí platnost oprávnění. Takový seznam může systém vygenerovat, ovšem účastníci se hlásí individuálně a dochází k velmi častým změnám.

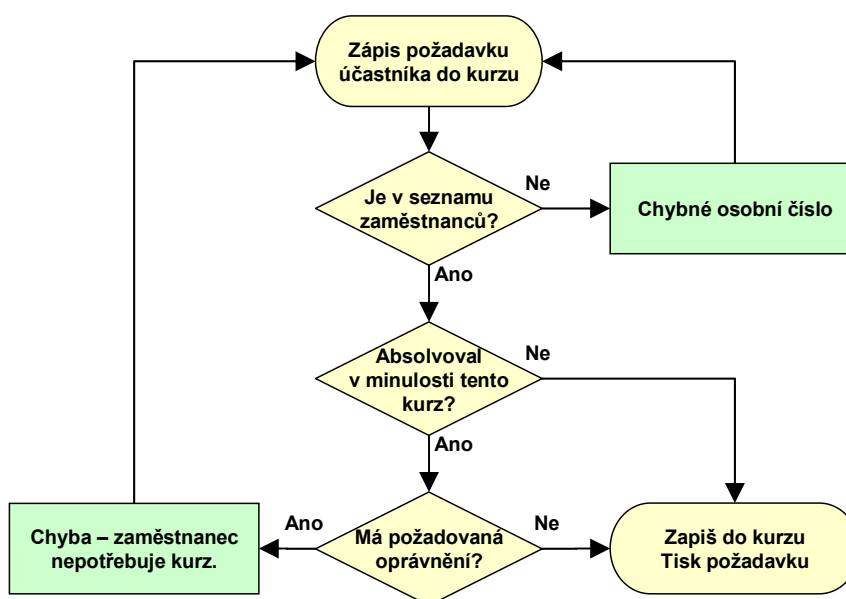
Opakovaný kurz uzavřený

- Základem je opět seznam zaměstnanců s končící platností, ovšem tento druh kurzu je určen předem definované skupině zaměstnanců, účast je tedy pravidelná a změny jsou minimální.

Z důvodu zásadních rozdílů mezi jednotlivými způsoby nominace účastníků bylo nutno tuto úlohu rozdělit na tři části, aby byly pokryty všechny varianty. Vznikly tak tři samostatné procedury s vlastními formuláři a datovými toky.

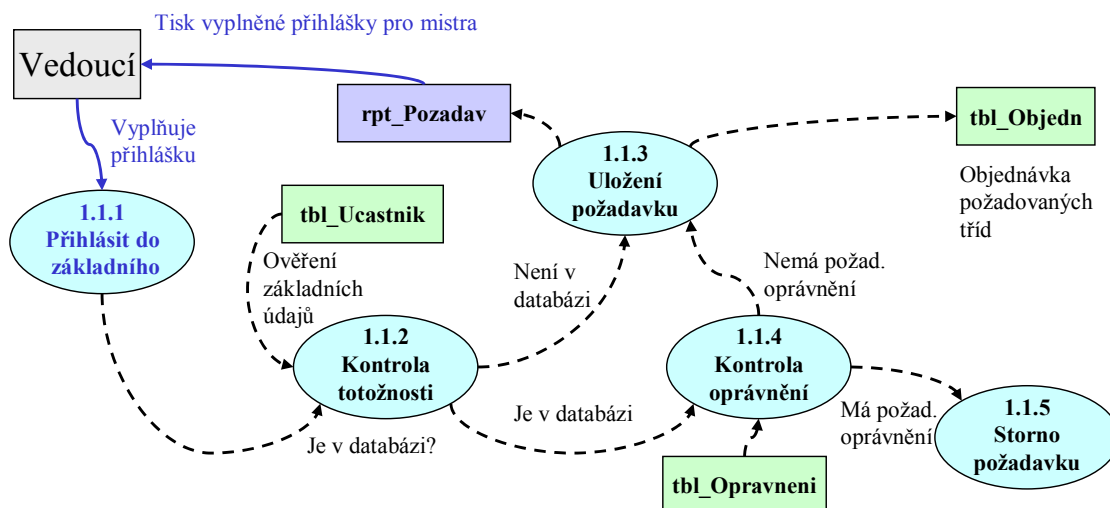
4.3.1 Nový kurz

Nový kurz je pro organizaci nejobtížnější z pohledu plánování. Nefunguje zde možnost využití historických dat pro predikci budoucího zájmu. Z tohoto pohledu se jeví jako nejúčinnější naprosto nesystémový odhad na základě zkušeností. Systémovým řešením je sběr požadavků ještě dříve, než je kurz založen, což umožní určit termín kurzu až v případě zájmu dostatečného počtu účastníků. Další zájemci se pak přihlašují na konkrétní termín až do naplnění kapacity kurzu.



Obr. 7. Zápis požadavku nového kurzu

Mistr vyplňuje přihlášku do základního kurzu. Data jsou uložena do tabulky Požadavků. Prvním krokem je potvrzení totožnosti. Systém porovnává osobní číslo se seznamem zaměstnanců a v případě neúspěchu okamžitě informuje mistra o chybě. Další krok je prohledání databáze účastníků kurzu. Nový účastník je zařazen mezi účastníky okamžitě, pokud však jde o zaměstnance, který již daný kurz absolvoval, musí dojít k porovnání požadovaných, a úspěšně dokončených tříd. Pokud zde není rozdíl, jde o chybu mistra a požadavek je vyřazen. V opačném případě je zaměstnanec zařazen mezi účastníky. Na závěr je celý proces potvrzen tiskem vyplněné přihlášky na základě uložených údajů.



Obr. 8. Datové toky přihlášky do nového kurzu

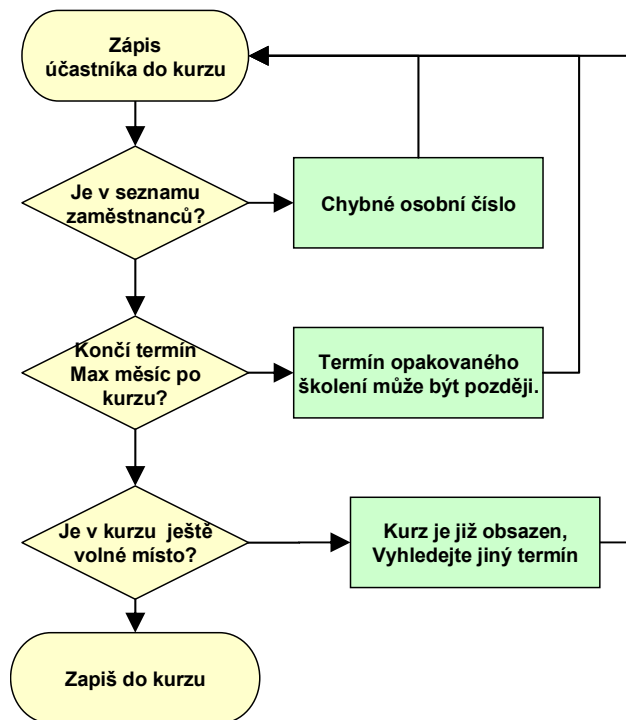
Každá úspěšně dokončená přihláška znamená zaplnění jednoho místa na seznamu účastníků a tedy snížení počtu volných míst v kurzu.

Samostatnou úlohou je zde příprava podkladů pro objednávku lektora. Podle rozsahu a náročnosti kurzu rozpočítá systém počty hodin výuky teorie a praxe a následně náklady rozdělí jednotlivým lektorům.

4.3.2 Opakovaný kurz volně přístupný

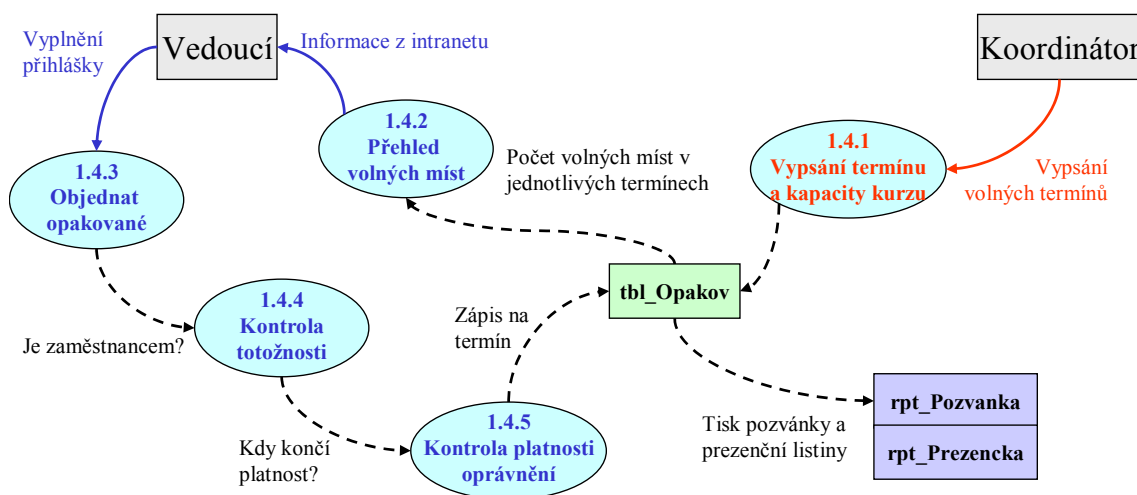
Tento typ kurzu je určen pro opakované školení velkého množství zaměstnanců, typicky např. školení řidičů nebo první pomoci. Na tyto školení se individuálně přihlašují zájemci, kterým končí platnost oprávnění. Vzhledem k počtu účastníků jsou tyto kurzy organizované velmi často a stávající systém telefonického přihlašování je pro koordinátory vzdělávání velmi zatěžující.

Odpovědnost za včasné absolvování kurzu je na nadřízeném daného zaměstnance, přesto byla jako podpůrný prostředek spuštěna na intranetu stránka, na níž se zobrazují jména a osobní čísla zaměstnanců, jimž končí oprávnění. Tento seznam slouží především vedoucím k snadnější orientaci při objednávání opakovaných školení.



Obr. 9. Zápis do opakovaného kurzu přes intranet

Po zápisu osobního čísla ověřuje systém podle seznamu zaměstnanců reálnou existenci zaměstnance, dále kontroluje termín vypršení platnosti a případně informuje o naléhavosti nebo zbytečnosti požadavku. Na závěr v případě obsazeného kurzu nabídne místo v nejbližším kurzu, který je volný. Po zaplnění kurzu dojde k rozeslání pozvánek a přípravy prezenční listiny.

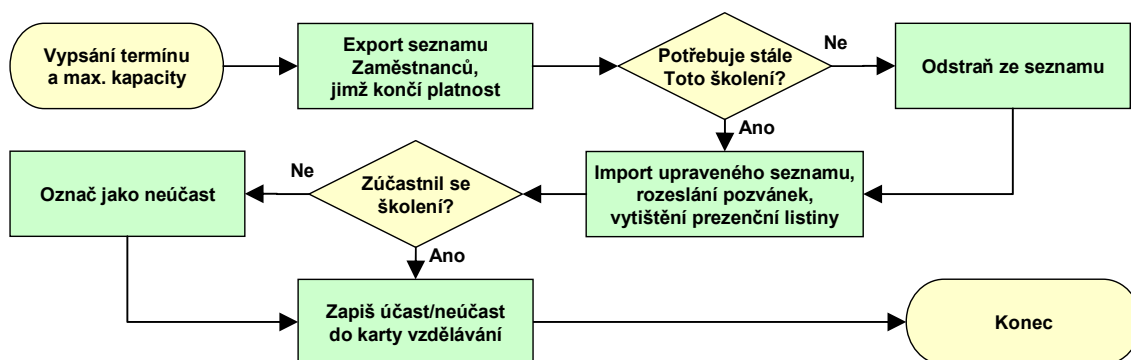


Obr. 10. Datové toky přihlášky do volně přístupného kurzu

4.3.3 Opakovaný kurz uzavřený

Nejjednodušší variantou jsou kurzy, které jsou cíleně organizovány pro předem danou skupinu účastníků. Příkladem mohou být školení svářečů nebo elektrikářů. Takových kurzů je malé množství a účastní se jich obvykle celá skupina současně.

Po vypsání termínu následuje automatické generování seznamu zaměstnanců, jimž končí platnost oprávnění. Tento seznam je odeslán nadřazeným daných zaměstnanců, kteří odstraní záznamy zaměstnanců, kteří daný kurz již nepotřebují například z důvodu změny pracovního zařazení. Odstraňování je daleko snazší a příjemnější, než vypisování seznamu a vyhledávání všech potřebných údajů. Další výhodou je možnost zpětného importu do systému na základě osobních čísel účastníků. Rozeslání pozvánek a tisk prezenční listiny je pak snadným úkolem bez nutnosti zásahu uživatele.



Obr. 11. Postup upřesňování seznamu účastníků

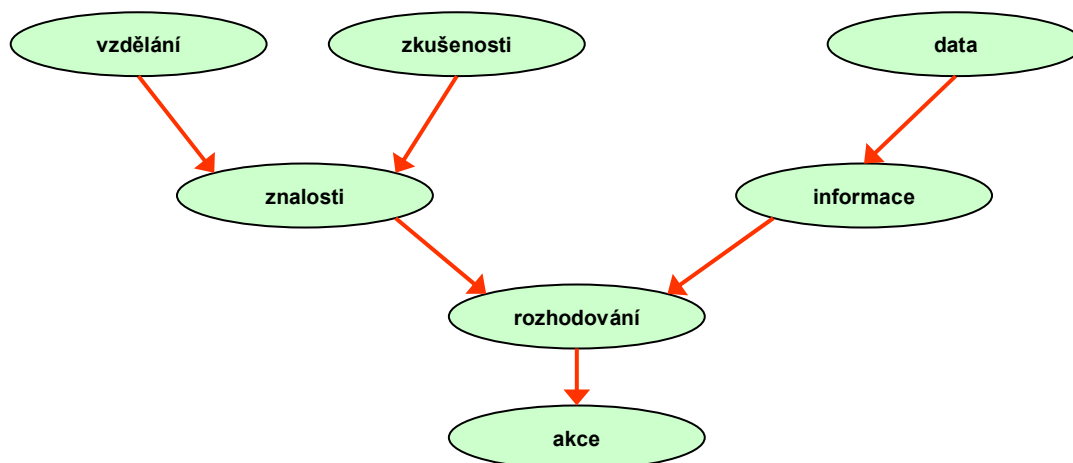
Po uskutečnění akce uživatel označí zaměstnance, kteří se nezúčastnili, a systém všem ostatním zapíše účast a na základě znalosti periody opakování nastaví datum vypršení oprávnění.

4.4 Výstupy pro rozhodování

Z výše popsaného je zřejmé, že vhodným využitím základních programovacích praktik bylo dosaženo kombinovaného sběru dat od mistrů formou objednávky s ověřením a doplněním aktuálních dat ze seznamu zaměstnanců personální divize. (náhorně vysvětluje Obr. 4) Tímto krokem byla omezena nutnost opisovat znovu data, která už byla někdy zadána.

Jedna z ústředních myšlenek této práce je maximální využití shromážděných dat. Samotná data zobrazují stavy objektů či probíhající procesy v realitě kolem nás, příjemce však nechávají netečným. Teprve informace vyvolává změnu stavu nebo chování příjemce.

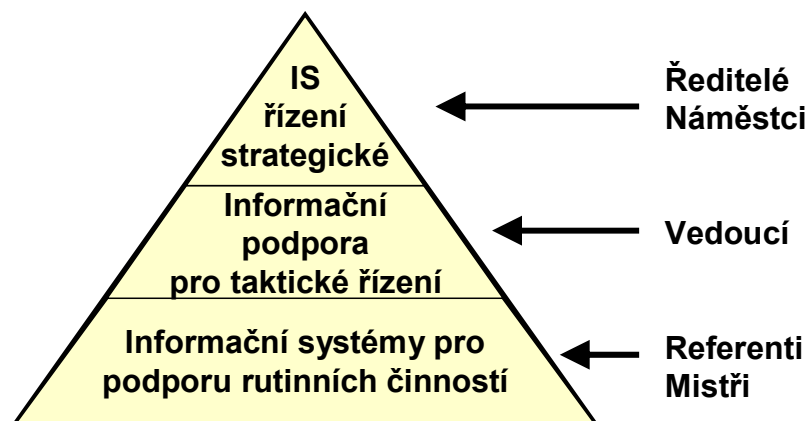
Pro kvalifikované rozhodnutí jsou nezbytné nejen informace, ale také znalosti, které představují zobecněné poznání reality (inspirováno zdroji [2] a [18]).



Obr. 12. Vztah mezi daty, informacemi a znalostmi

Význam nového informačního systému je v jeho přidané hodnotě, kterou přináší kvalitnější zpracování a prezentace dat. Původní pohled na data, která mají primárně sloužit nezbytným rutinním činnostem zabezpečení jednotlivých vzdělávacích aktivit, se mění. Pro kvalitativně vyšší úroveň řízení celého procesu mají stále důležitější roli informace získané vhodným zpracováním dat. Nový systém tyto informace umí nejen poskytnout, ale také prezentovat ve vhodné podobě. Není nutno čekat na účetní uzávěrku na konci roku, nebo zjišťovat data z různých zdrojů. Vše je přehledně seskupeno ve standardních výstupech. Dokonce i nestandardní úlohy jsou díky pružnosti zvolené platformy MS Access snadno řešitelné dodatečným vytvořením SQL dotazů a speciálních sestav. Jedním z úkolů této aplikace je zjistit všechny potřebné výstupy a nalézt optimální formát jejich prezentace.

Systém se přesouvá z roviny zabezpečení rutinních operací do pozice nástroje pro střednědobé taktické řízení a výhledem do budoucna jsou výstupy pro strategické rozhodování vrcholového managementu.



Obr. 13. IS jako podpora jednotlivých úrovní řízení

Typickým příkladem jsou statistiky a různé pohledy na data umožňující mít celý proces pod kontrolou. Pro vlastní chod aplikace i samotný proces organizace povinných školení nejsou tyto informace nezbytné, ale vhodná prezentace získaných informací inicializuje zlepšování procesu.

Příklady statistických dat:

- Přehledný rozpis odučených hodin každého instruktora včetně rozdělení nákladů na jednotlivá oddělení usnadní účetním rozpočítání částky faktur podle počtu účastníků.
- Přehled platnosti oprávnění podle jejich typů.
- Procento zaměstnanců vlastnících určité oprávnění vzhledem k celkovému počtu zaměstnanců oddělení umožní cílenější výběr zaměstnanců.
- Počty účastníků opakovaných školení v jednotlivých měsících umožní přesnější plánování termínů pro příští rok.
- Počty termínů je nutné volit s ohledem na počty proškolených v předchozím roce.
- Počty absencí objednaných účastníků školení umožní přesnější plánování.
- Poměr školení realizovaných interními lektory.

Velmi cenné je porovnání aktuálních dat s historií. Z tohoto důvodu jsou vybraná data ukládána do souhrnných tabulek s atributem roku pořízení. Reporty tohoto druhu nejsou doposud specifikovány a budou upřesňovány postupně podle požadavků.

5 BEZPEČNOSTNÍ PRVKY

Tato část obsahuje změny nejdůležitější pro vlastní systém. Uživatelé by tyto změny měli pocítit co nejméně. Realizace takto zvolené bezpečnostní politiky není snadnou záležitostí, neboť každý bezpečnostní prvek obvykle znamená určité omezení uživatele.

Nezbytným základem je řízený přístup do systému. Po ověření uživatele následuje úprava práv a také vzhledu aplikace. Nejméně nápadným prvkem je pak evidence klíčových kroků každého uživatele a historizace dat. Podrobněji jsou tyto kroky popsány v následujících kapitolách.

5.1 Ověření uživatele podle přihlášení ve WIN XP – API

Na základě popsaných informací o způsobech autentizace v kapitole 2.4 byla řešena základní otázka, jestli bude vhodnější vlastní heslo nebo bude lépe využít přihlašovacího profilu do Windows. Pro tvorbu samostatných aplikací je většinou doporučován vlastní přihlašovací dialog. V tomto případě ale bylo rozhodnuto pro variantu pro uživatele příjemnější. Stačí být přihlášen ve Windows jako jeden z oprávněných uživatelů a aplikace toto přihlášení převezme díky API funkci. Praktické řešení této úlohy je popsáno níže. Na začátku modulu je umístěna deklarace v této podobě:

```
1 Declare Function wu_GetUserName Lib "advapi32.dll" Alias _
2   "GetUserNameA" (ByVal lpBuffer As String, nSize As Long) _
3   As Long
```

Vlastní API funkce pak přiřadí návratovou hodnotu do předem připravené řetězcové konstanty. Vzhledem k tomu, že API doplňuje prázdná místa nulami, bylo nutno tento řetězec zkrátit po poslední nenulový znak před předáním funkci VBA ve formě návratové hodnoty.

```
1 Function ap_GetUserName() As Variant
2   Dim strUser As String
3   Dim lngLength As Long
4   Dim lngResult As Long
5   '-- Připrav retezcovou promennou pro ulozeni jmena
6   strUser = String$(255, 0)
7   lngLength = 255
8   '-- Volej rutinu API
9   lngResult = wu_GetUserName(strUser, lngLength)
10  '-- Prirad navratovou hodnotu
11  ap_GetUserName = Left(strUser, InStr(1, strUser, Chr(0)) - 1)
```

```
12 End Function
```

Tohoto řešení bylo možno využít díky velmi přísně definovaným pravidlům pro přihlašování do Windows.

Parametry hesla:

- Minimálně 6 znaků obsahujících alespoň tři z těchto čtyř kategorií znaků: velké písmeno, malé písmeno, číslo a speciální znak.
- Maximálně 2 opakující se znaky.
- Minimálně 3 znaky nenacházející se v minulém heslu.
- Heslo se nesmí opakovat v posledních 50 případech.
- Doba platnosti maximálně 6 měsíců.

Doplňkem zvyšujícím bezpečnost je automatické uzamykání stanice při nečinnosti delší než 10 minut.

Následující část kódu zapíše právě přihlášeného uživatele s časem otevření souboru do tabulky *"T98UzivatelePristup"*. Tímto způsobem je možno archivovat každé otevření a zavření souboru, což umožní zjištění aktuálně přihlášených uživatelů i dobu práce každého z nich.

```
1  UzivatelZkratka = ap_GetUserName()  
2  '-- Otevri tabulku  
3  Set rs = db.OpenRecordset("T98UzivatelePristup")  
4  '-- Zapis vstup uzivatele do tabulky  
5      With rs  
6          .AddNew          '-- Pridej zaznam  
7          !Uzivatel = UzivatelZkratka  
8          !DatOtev = Now  
9          .Update          '-- Uloz zmeny  
10         .Bookmark = rs.LastModified  '-- Bez na novy zaznam  
11         Pristup = !IDpristup  
12     End With  
13 rs.Close
```

Kontrola oprávněnosti přístupu je zajištěna porovnáním údajů zjištěných o přihlášeném uživateli z Windows se seznamem oprávněných uživatelů v tabulce *"T99uzivatel"*. V této tabulce je po ověření nalezeno skutečné jméno uživatele, které je zapsáno do tabulky

"T99aplikace" pro další použití. Pokud uživatel v seznamu není, je zapsán prázdný řetězec, dojde k uložení informací o jeho neúspěšném přihlášení a následuje ukončení aplikace.

```

1     UzivatelJmeno = DLookup("[AktUzivatelJmeno]", "T99aplikace")
2     If Len(UzivatelJmeno) > 0 Then
3         '-- Uzivatele znam, pokračuj
4     Else
5         Application.Quit acQuitSaveAll '-- Ulož a zavři aplikaci
6     End If

```

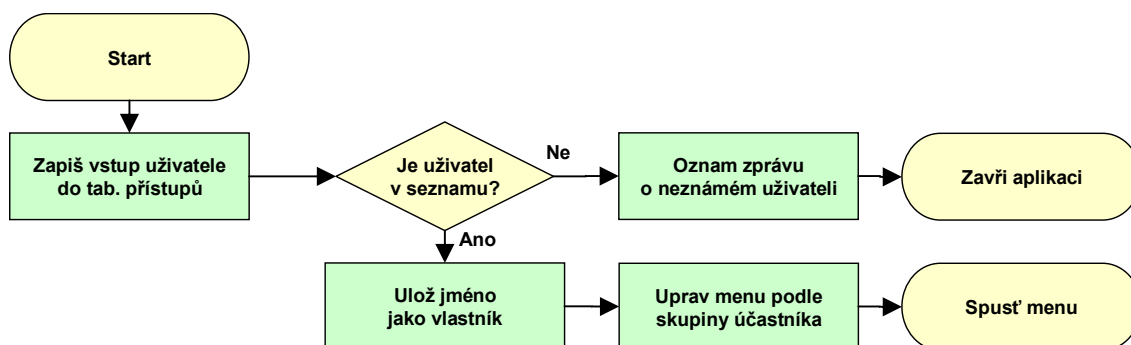
Z uživatelského hlediska je popsání řešení příjemnější, neboť každý uživatel využívá několik systémů vyžadujících různá hesla. Při větším počtu těchto hesel je náročné si všechna pamatovat, pak jsou volena buď všude stejná, nebo jsou napsána na papírku u počítače.

5.2 Úprava uživatelských práv a vzhledu aplikace

Jestliže systém identifikuje uživatele, který je v seznamu uživatelů, přiřadí mu práva podle zařazení do jedné ze skupin. Technicky není problém nadefinovat každému uživateli samostatný profil a práva, ale údržba pak není snadná ani účelná. Skupina referentů vzdělávání je navíc poměrně stejnorodá z pohledu pracovních úkolů a tedy i formy přístupu. Postačí tedy pouze tři typy práv:

- Administrátor.
- Uživatel.
- Host.

Obrázek 14 znázorňuje celý proces zápisu dat od přístupu, přes ověření uživatele a úpravu práv v uživatelském prostředí až po spuštění hlavního menu. Všechny tyto procesy jsou úzce spojené.



Obr. 14. Úprava aplikace podle přihlášeného uživatele

Výsledkem je pak možnost poskytnout přístup k datům dalším uživatelům mimo referenty vzdělávání. Ti mohou mít přístup ze skupiny host a tedy právo pouze pro čtení. Všichni referenti mají přístup uživatelský, který umožní editaci dat pro všechny standardní operace. Nejvyšší stupeň oprávnění je administrátorský, který má vedoucí vzdělávání a administrátor systému. Ten je oprávněn měnit číselníky, které jsou jinak zadány jako "pevné" a provádět editaci dat, která jsou jinak již uzamčena pro další úpravy.

5.3 Sledování změn klíčových položek

Ani nejdokonalější systém nezabrání nechtěným chybám, odstranění dat, nebo dokonce záměrnému útoku na data. Následky těchto rizik minimalizuje uchovávání historických dat v podobě před změnou, datum a identifikaci uživatele, který změnu provedl. Tímto způsobem je možno obnovit řadu chyb a vysledovat chování uživatelů. Základním pravidlem je nemazat data při požadavku uživatele na odstranění záznamu. V takovém případě je vhodnější vložit do tabulky k záznamu atribut datum odstranění. Pro uživatele již takový záznam jakoby neexistuje, neboť podkladový dotaz tento záznam vyfiltruje. Ve skutečnosti je však uložen jak záznam, tak údaj o datu a osobě, která odstranění provedla.

Pro sledování změn je situace složitější, neboť je nutno zachovat také historickou hodnotu daného atributu. K tomu bylo využito tabulek, do kterých jsou zapisovány změny. Pro přesnou identifikaci místa změny je nutno zaevidovat jak číslo záznamu dle jeho primárního klíče, tak název pole a tabulky.

Tab. 6. Ukázka tabulky zaznamenávající změny

Kdy	Kdo	Tabulka	Pole	IDzaznamu	PredZmenou
15.4.2006	admin	T11Zamest	Jmeno	123456	Honza
15.4.2006	admin	T11Zamest	Adresa	123456	Dlouhá 254
16.4.2006	mechl	T21Skoleni	NazevKurzu	25487	Elektro – vyhláška 50

Ne všechny změny je však nutné monitorovat, např. změny telefonních čísel nejsou pro běh aplikace rozhodující a tak jsou archivovány pouze změny vybraných položek. Tabulky změn při větší frekvenci oprav úctyhodně narůstají, proto bylo nutno omezit archi-

vaci na poslední měsíc a provádět pravidelnou údržbu, která starší záznamy změn po vyhodnocení automaticky odstraní.

6 PROPOJENÍ SE STÁVAJÍCÍM IS

Základním pilířem všech změn je propojení doposud samostatné databáze do IS firmy (důvody popsány ve zdrojích [7] a [16]).

V praxi nebylo snadné zajistit přímé propojení na "dospělé" databáze typu DB2 nebo Oracle. Důvodem byla především bezpečnost a také stabilita systému. Z těchto důvodů došlo ke kompromisu ve formě exportu dat z těchto systémů a import těchto dat do databáze.

Tento způsob neumožňuje zpětnou vazbu, což je pochopitelně nedostatkem, který pro daný systém není omezující, neboť získaná data není třeba měnit, stačí je využívat.

V opačném směru toku dat je nejvyužívanějším komunikačním kanálem pro předávání vybraných aktuálních informací intranet.

Technické řešení jednotlivých úloh je popsáno v následujících kapitolách.

6.1 Export dat z IS

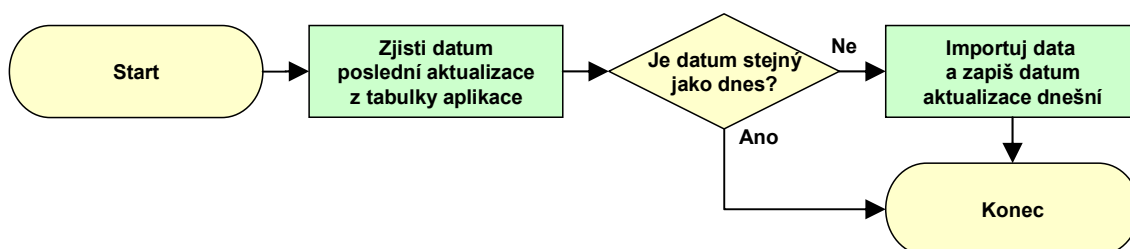
Jedním ze záměrů této práce je minimalizace zadávání dat do PC. První oblastí, kde se tento záměr podařilo uskutečnit je doplňování informací o zaměstnancích. Bylo využito seznamu zaměstnanců, který obhospodařuje personální divize ve své databázi. Z bezpečnostních důvodů nebyl umožněn přímý přístup k datům. Vybraná data jsou však automaticky exportována vždy v noci do textového souboru na serveru. K tomuto textovému souboru už má aplikace přístup a vždy ráno při prvním spuštění dojde k importu do interní tabulky, aby byla data pohotově k dispozici. Denní aktualizace je vzhledem k povaze a stálosti dat naprosto dostatečná.

Druhou oblastí je seznam oddělení zadavatele, který je přebírán ze zdroje v MS Excel. Zajímavostí je zde transformace struktury dat z tabulkové na hierarchickou.

6.1.1 Seznam zaměstnanců

Export dat z personální databáze zaměstnanců je řešen jednou denně formou předem definovaných výstupů do textových souborů. Soubory mají standardní strukturu, mění se pouze počet záznamů, tzn. řádků textu. Export je automaticky spuštěn vždy v noci, aby data byla připravena na ranní aktualizaci. Tyto textové soubory by mohly být importovány přímo do MS Access, ale při tomto způsobu docházelo k chybám při exportu způsobeným

špatným datovým formátem nebo neplatnou hodnotou. Z tohoto důvodu byl vytvořen excelovský sešit, který tyto textové soubory importuje, upraví všechny sloupce do požadovaných formátů a opraví některá rozdílně interpretovaná data, například datum. Celá procedura importu a úprav je uložena v MS Excel a spouštěna z MS Access. Uživatel tedy vůbec nepozná, co všechno při prvním spuštění aplikace v každém dnu probíhá. Z denní aktualizace samozřejmě vyplývá další nedostatek a tím je neaktuálnost způsobená tím, že úpravy dat v aktuálním dnu se neprojeví okamžitě, ale až následující den. Tady je nutno upozornit, že data o zaměstnancích se v čase nijak dramaticky nemění.



Obr. 15. Schéma importu dat z podnikového IS

Prvním krokem je ověření data poslední aktualizace. Tato informace je uložena v tabulce "T99Aplikace" a je tedy snadné ji porovnat s aktuálním datem. V případě stejné hodnoty procedura končí, v opačném případě je potřeba data aktualizovat. MS Access je schopen data importovat z textových souborů přímo, ale v praxi docházelo k nepřesnostem ve formátech polí, kde nebylo podle struktury dat jasné, zda se jedná o textový řetězec, číslo, nebo datum. Ne zrovna elegantním, ale spolehlivě funkčním řešením se ukázal import do MS Excel, úprava dat na jeho listu a následně import do MS Access. Následující kód spouští na pozadí proceduru VBA uloženou v MS Excel.

```

1 Dim spojeni As New ADODB.Connection
2 Dim appExcel, sExcel As Excel.Workbook, listExcel As Excel.Worksheet
3 Set appExcel = Excel.Application '-- Otevri Excel
4 Set sExcel = appExcel.Workbooks.Open(FileName:=Adresar() &
  "\ZamestnanciTabulky.xls", ReadOnly:=False)
5 Set listExcel = sExcel.Worksheets(1)
6 appExcel.Visible = False '-- Excel na pozadi
7 '-- Zavolej makro v danem sesitu excelu
8 appExcel.Run "ZamestnanciTabulky.xls!Zamestnanci"
9 '-- Uklid z pameti
10 Excel.Application.DisplayAlerts = False
11 sExcel.Close True
12 Set sExcel = Nothing
13 appExcel.Quit
  
```



```
14      Set appExcel = Nothing
```

Okamžitě po dokončení procedury v MS Excel jsou odstraněna data ze stávající tabulky a následně je spuštěn přidávací dotaz uložený v kódu VBA, který doplní aktuální data do tabulky "T11Zaměstn". Díky tomuto seznamu stačí pro identifikaci zaměstnance znát jeho firemní osobní číslo. Všechny ostatní údaje je možno doplnit z tohoto zdroje. Identifikace na základě příjmení a jména není možná, neboť se tyto vyskytují duplicitně.

Při aktualizaci dochází k zálohování posledně použitých dat do tabulky "T12ZamestnVcera". Hlavním důvodem tohoto kroku je funkce, která porovnává tabulku dnešní se včerejší a vypisuje všechny rozdíly do samostatné tabulky změn. Přehled těchto změn slouží pro přesnější kontrolu. Záložní data samozřejmě umožňují vrátit se ke včerejším datům, pokud by se import nezdařil.

```
15      DoCmd.SetWarnings False '-- Ignoruj varovne hlasky
16 '-- Vymaz zalozni tabulku
17      DoCmd.RunSQL "DELETE T12ZamestVcera.* FROM T12ZamestVcera;"
18 '-- Prekopiruj vcerejsi data do zalozni tabulky
19      DoCmd.RunSQL "INSERT INTO T12ZamestVcera SELECT T11Zamest.* FROM
T11Zamest;"
20 '-- Vymaz aktualni tabulku
21      DoCmd.RunSQL "DELETE T11Zamest.* FROM T11Zamest;"
22 '-- Vloz data z excelu do aktualni tabulky accessu
23      DoCmd.RunSQL "INSERT INTO T11Zamest SELECT Zamestnanci.* FROM
Zamestnanci;"
```

6.1.2 Seznam oddělení

Export oddělení a divizí, tedy firemní struktury je řešen obdobným způsobem jako seznam zaměstnanců, s tím rozdílem, že aktualizace probíhá pouze tehdy, pokud byl podkladový vyexportovaný soubor změněn. V praxi dochází k úpravám přibližně 4krát ročně. Z uvedeného je zřejmé, že nemá smysl aktualizovat tabulku oddělení každý den, ale pouze při změně zdrojových dat, která jsou na serveru v souboru MS Excel. Následující část kódu porovnává datum poslední aktualizace s datem uložení zdrojového souboru na serveru.

```
1 Dim Oddel As String, Datum As Date
2      Oddeleni = Adresar & "\Strediska\ImportStredisek.xls"
3      If DLookup("[ImportNS]", "T99Aplikace") = FileDateTime(Oddel)
4      Then
5          '-- Tabulka je aktualni
6      Else
7          '-- Je nutna aktualizace
```

```
7      End If
```

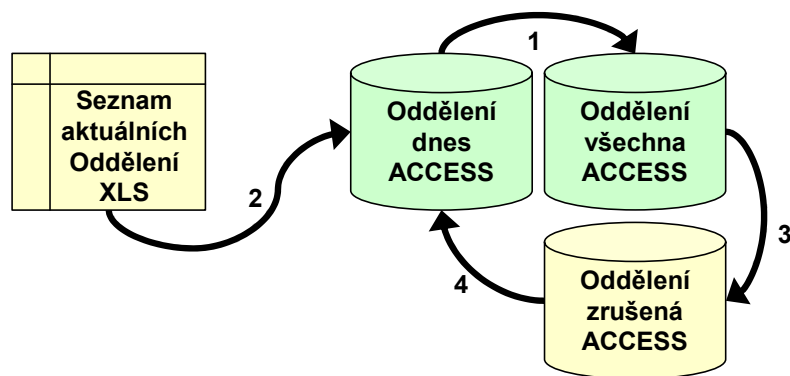
Přímý import nebo propojení zdrojové tabulky není možné, neboť struktura oddělení je ve skutečnosti stromová. Podkladová data je nutno hierarchicky rozdělit do dvou tabulek *Divize* a *Oddělení*. Tato úprava dat probíhá v tabulkách MS Excel, ve kterých je umístěna procedura VBA spouštěná při aktualizaci z MS ACCESS.

Aktualizace probíhá stejným způsobem, jako u zaměstnanců, tedy odstraňovacím a přidávacím dotazem spouštěným z modulu kódu. Na závěr každé aktualizace je nutno zapsat datum do tabulky "*T99Aplikace*" pro porovnávání aktuálnosti dat při příštím spuštění.

```
1 Dim db As Database, rs As Recordset
2     Set db = CurrentDb
3 '-- Otevri tabulku
4     Set rs = db.OpenRecordset("T99Aplikace")
5 '-- Zapis datum do tabulky
6     With rs
7         .Edit                '-- Uprav zaznam
8         !ImportNS = FileDateTime(Oddeleni)
9         .Update              '-- Uloz zaznam
10    End With
11    rs.Close
12 End If
```

6.2 Archiv historických stavů

Tím, že dochází k importu kompletní sady záznamů, může nastat situace, že historicky platná data se už v aktuální sadě neobjeví. Požadavek však zněl zachovat historická data pro archivaci, ale pro nové záznamy nabízet pouze aktuální sadu dat. Z tohoto důvodu byly rozšířeny tabulky o pole zaznamenávající datum odstranění záznamu. Záznamy tedy nejsou odstraněny fyzicky, ale pouze označeny tímto datem a výběrový dotaz testující vyplnění tohoto pole zobrazuje pouze aktuálně platné záznamy. Komplikovanější situace nastala v případě importovaných tabulek. Ty obsahují vždy pouze aktuální data a historii neposkytují. Tento nedostatek byl vyřešen přidáním pomocné tabulky zachovávající poslední uložený stav, jejímž porovnáním s aktuálním stavem je možno získat rozdíly, které jsou uloženy do tabulky historie. Příkladem je aktualizace číselníku oddělení.



Obr. 16. Aktualizace s uchováním historických dat

Postup při aktualizaci:

- Zálohování dat do tabulky *Oddělení všechna*.
- Import aktuálních dat do tabulky *Oddělení dnes*.
- Vyfiltrování historických záznamů, která nejsou obsažena v aktuální sadě dat do dočasné tabulky *Oddělení zrušená* a jejich označení příznakem "zrušeno".
- Přidání vyfiltrovaných záznamů do tabulky *Oddělení dnes*.

Historická data není nutno archivovat trvale. V případě povinných školení je minimální nezbytná doba archivace po dobu platnosti daného oprávnění. Požadavkem zadavatele je archivace dat po dobu pěti let. Ani po této době nebudou data odstraněna nenávratně, ale dojde k jejich přesunu do záložních tabulek mimo vlastní systém. V případě potřeby pak bude možnost tyto data znovu "oživit".

V průběhu času data ztrácí na aktuálnosti, ale jejich cena vzrůstá právě s dobou, po kterou jsou sbírána. Na významu pak získává a statistický pohled na data. Pro tyto účely byly vytvořeny souhrnné tabulky, které po uzavření roku seskupí data tak, aby statistické výpočty netrvaly dlouho a byly pohotově k dispozici (Více se tomuto tématu věnuje kapitola 4.4).

6.3 Komunikace přes intranet

Některé úpravy systému přinesly změny patrné i mimo oddělení vzdělávání. Široce diskutována byla forma distribuce informací. Některé dokumenty musí být z pochopitelných důvodů tištěné a verifikované, např. smlouvy nebo objednávky. Pro přenos informací v elektronické podobě se nabízejí dva nejdůležitější kanály: elektronická pošta a intranet. Rozesílání e-mailu je osobní, cílené a s možností rychlé zpětné vazby. Intranet naproti tomu přináší standardní schránku pro ukládání dat ve formě souborů. Taková schránka zna-

mená, že zaměstnanci vědí, kde aktuální data hledat a pro jejich aktualizaci stačí obnovit jediný soubor. O aktualizaci samozřejmě uživatelé nejsou informováni, změnu poznají, až budou danou informací znovu hledat. Další předností intranetu je poskytování jednotného přístupu k různým dokumentům, které byly vytvořeny různými aplikacemi.

Veřejný přístup do databázových aplikací není možný ani žádoucí. Vybraná data je však možno prezentovat formou dynamických HTML stránek na intranetu. Pro distribuci informací bylo využito výhod obou technologií.

Elektronická pošta byla využita pro rozesílání pozvánek a seznamů pro hromadná školení. Většinu ostatních úkolů převzal firemní intranet. Využívání této formy komunikace stále narůstá a tak je tento krok logickým vyústěním zdokonalování informačního systému. Využití intranetu má dvě podoby. Jednak zveřejnění údajů, které byly posílány elektronickou poštou, nebo sdělovány telefonicky a za druhé příjem informací doposud probíhající pouze papírovou formou. Nespornou výhodou oproti e-mailu je obousměrný strukturovaný dialog prostřednictvím HTML formulářů založených na databázových tabulkách. Nejdůležitější úpravy jsou zmíněny v následujících odstavcích (další postupy ve zdroji [4]).

6.3.1 Informace o volných místech

Snadným, ale velmi prospěšným krokem bylo zveřejnění informací o probíhajících školeních včetně termínů s aktuálním počtem volných míst. Tato stránka slouží pro rychlý přehled aktuálně probíhajících kurzů. Je zpracována jako statická, exportována denně.

Záměrem bylo zpřístupnit velkému počtu zájemců informace přehledně, jednoduše a s krátkou dobou odezvy. Při exportu je využito standardní funkce MS Access exportující předem připravenou sestavu do formátu HTML.

```
1 DoCmd.OutputTo _  
2     ObjectType:=acOutputReport, _  
3     ObjectName:="R11DivizeOddel", _  
4     OutputFormat:=acFormatHTML, _  
5     OutputFile:=Application.CurrentProject.Path & "\Akce.html"
```

6.3.2 Přehled o termínech vypršení platnosti oprávnění

Díky údajům o účasti na školení v minulých letech a znalosti doby platnosti oprávnění je možnost vyhledat termín vypršení platnosti, případně zobrazit seznam všech řidičů, kterým

už platnost vypršela. Tyto kontrolní mechanismy usnadní přehled o vzdělávacích potřebách.

Vzhledem k velkému množství údajů by zobrazení běžného seznamu výrazně snížilo přehlednost a tedy i účelnost takové stránky. Proto bylo rozhodnuto exportovat nezbytná data do tabulky mimo systém a tuto tabulku použít jako zdroj pro dynamickou stránku HTML, která umožňuje filtrovat data na základě zvoleného typu školení a čísla oddělení. Tímto způsobem získá nadřizený přehled o potřebách opakovaných školení svých zaměstnanců.

6.3.3 Možnost objednávat se automaticky

Zatímco předchozí stránky jsou pouze informativní, logickým důsledkem obousměrné komunikace přes intranet je formulář pro vstup dat. Na základě termínu vypršení platnosti a volných místech je možno okamžitě přihlásit zaměstnance do kurzu. Objednání probíhá na základě vyplnění osobního čísla, stejně jako případné stornování přihlášky. Na seznamu se pro kontrolu zobrazují pouze jména přihlášených. Tím je zamezeno neoprávněné manipulaci v podobě odstraňování přihlášených zaměstnanců. Pro případné ověření přihlášení a následné odhlášení zaměstnance jsou tyto informace archivovány s datem ohlášení stejně jako všechny ostatní odstraňované záznamy.

6.4 Export souhrnných dat pro prezentace

Všechny rutinní i většina rozhodovacích operací jsou prováděny v rámci systému. Jsou však situace, kdy je třeba informace prezentovat mimo systém, např. managementu firmy. Pak je nejvhodnější vybrat potřebná data a ta vhodně graficky upravit, podpořit grafy a vše vložit do prezentačního software, v tomto případě MS PowerPoint. Pokud jde o standardní výstupy typu roční uzávěrky, je celá operace automatizována. Komplikovanější situace je, pokud není předem jasné, jaká data bude potřeba. MS Access je velmi pružná platforma, ale uživatel musí mít dostatečné znalosti, aby ji mohl dostatečně využít. Požadavek zadavatele byl nalézt maximum standardních výstupů, aby byla jejich struktura jasná a ověřená dostatečně brzy před přechodem na nový systém na platformě Oracle.

Pro nestandardní výstupy a "hledání" optimálních pohledů na data byly zapojeni všichni uživatelé. Aby se reálně mohli zapojit, byl jim poskytnut automatizovaný výstup do tabulek MS Excel, který všichni ovládají. Tímto byl položen základ pro širší diskusi o nejvhodnějším způsobu prezentace a distribuce informací.

Jádrum exportního modulu je procedura VBA, která vyexportuje data do konkrétní pojmenované oblasti konkrétního listu daného souboru MS Excel.

```
1 Function KopirujTabDoBunekXLS(Tabulka As String, Sesit As String,  
  List As String, Oblast As String)  
2 Dim objXL As Excel.Application  
3 Dim objWkb As Excel.Workbook  
4 Dim objSht As Excel.Worksheet  
5 Dim db As Database  
6 Dim rs As Recordset  
7 Const conMAX_ROWS = 20000  
8  
9 Set db = CurrentDb  
10 Set objXL = New Excel.Application  
11 Set rs = db.OpenRecordset(Tabulka, dbOpenSnapshot)  
12 With objXL  
13     .Visible = True  
14     Set objWkb = .Workbooks.Open(Sesit)  
15     On Error Resume Next  
16     Set objSht = objWkb.Worksheets(List)  
17     If Not Err.Number = 0 Then  
18         Set objSht = objWkb.Worksheets.Add  
19         objSht.Name = List  
20     End If  
21     Err.Clear  
22     On Error GoTo 0  
23     objSht.Range(Oblast).CopyFromRecordset rs  
24 End With  
25 Set objSht = Nothing  
26 Set objWkb = Nothing  
27 Set objXL = Nothing  
28 Set rs = Nothing  
29 Set db = Nothing  
30 End Function
```

6.5 Zahrnutí všech opakovaných povinných školení

Řidiči motorových vozíků jsou pouze jednou, i když nejrozsáhlejší oblastí povinných školení. Po úspěšném zpracování této agendy databázi bylo rozhodnuto o rozšíření na všechny oblasti povinných školení. Tímto krokem dojde ke zjednodušení řízení povinných školení a také žádoucí unifikačnosti. Všechny typy školení tak budou organizovány i objedná-

vány stejným způsobem, což jistě přivítají nejen uživatelé (garanti vzdělávání), ale také účastníci školení.

Do databáze musel být zakomponován seznam povinných školení včetně periody opakování, aby mohly být stejným způsobem zpracovány i školení opakující se po 3 nebo 5 letech.

ZÁVĚR

Hlavním cílem této diplomové práce je vytvoření aplikace pro organizaci povinných školení v Barum Continental spol. s r.o. na základě databáze Řidičů manipulačních vozíků zpracované v rámci bakalářské práce. Klíčovým úkolem je zakomponování celé aplikace do existujícího informačního systému firmy tak, aby přehledněji poskytovala více informací při nižší časové náročnosti na obsluhu a umožnila jednoduchou kontrolu nad průběhem vzdělávací akce.

Propojením na stávající informační systém je umožněno získání aktuálních informací bez nutnosti všechna data zadávat. Žádaným zdokonalením je dopracování objednávkového systému přes intranet, což přináší snížení nároků na vypisování údajů. Velká péče je věnována automatizaci a standardizaci úloh tak, aby byly minimalizovány zásahy uživatele.

Za zkvalitnění služeb směrem k interním zákazníkům je možno považovat generování přesných podkladů pro fakturaci.

Aplikace nyní zahrnuje všechny oblasti povinných školení realizovaných u zadavatele. Využití všech možností aplikace přináší efektivnější využití zdrojů a zkrácení doby obsluhy. Průběžně budou vyhodnocována další zadavatelem stanovená kritéria. Následovat budou moduly zaměřující se na vyhodnocování efektů vzdělávacích akcí.

Řešení využívá programu MS Access a podnikové sítě intranet. Přestože jde o sofistikované řešení s řadou bezpečnostních prvků, stále je to "jen" krátkodobý funkční model, který slouží pro testování funkcí v praxi a vyškolení uživatelů. Celý projekt směřuje k převedení tohoto modelu na platformu "dospělá" databáze Oracle.

SEZNAM POUŽITÉ LITERATURY

Monografie:

- [1] BARKER, F., Scott. *Microsoft Access 2002 – Programování databázových aplikací*. Praha: Computer Press, 2002. ISBN 80-7226-611-X
- [2] BERKA, Petr. *Dobývání znalostí z databází*. Praha: Academia, 2003. ISBN 80-200-1062-9
- [3] DOBTA, Luboš. *Ochrana dat v informačních systémech*. Praha: Grada Publishing, 1998. ISBN 80-7169-479-7.
- [4] LACKO, Luboslav. *Web a databáze*. Praha: Computer Press, 2001. ISBN 80-7226-555-5
- [5] MECHL, Viktor. *Systém kompletního vedení agendy řidičů MV*. Zlín: UTB, 2004. Bakalářská práce.
- [6] MOLNÁR, Zdeněk. *Efektivnost informačních systémů*. Praha: Grada Publishing, 2000. ISBN 80-7169-410-X.
- [7] POKORNÝ, J. *Databázové systémy a jejich použití v informačních systémech*. Praha: Academia, 1992. ISBN 80-200-0177-8
- [8] POKORNÝ, J. *Spolupráce aplikací MS Office*. České Budějovice: KOPP, 1997. ISBN 80-85828-81-2
- [9] POKORNÝ, Jaroslav. *Visual Basic pro aplikace Accessu 97*. České Budějovice: Kopp, 1997. ISBN 80-7232-003-3
- [10] POLÁK, J., MERUNKA, V. a CARDA, A. *Umění systémového návrhu*. Praha: Grada Publishing, 2003. ISBN 80-247-0424-2
- [11] ŘEPA, V. *Analýza a návrh informačních systémů*. Praha: Ekopress, 1999. ISBN 80-86119-13-0
- [12] SOLOMON, Christiane. *Tvorba aplikací v MS Office 97 pomocí jazyka VB*, Praha: Computer Press 1998. ISBN 80-7226-025-1
- [13] STEVEN, Roman. *Microsoft Access Návrh a programování databází*. O'Reilly, Brno: Computer Press 1999. ISBN 80-7226-134-7

- [14] STINSON, Craig, DODGE, Mark. *Mistrovství v Microsoft Office Excel 2003*. Brno: CP Books, 2005. ISBN 80-251-0669-1
- [15] STONEBRAKER, M., BROWN, P. *Objektově-relační SŘBD, analýza příští velké vlny*. BEN, 2000, Praha. ISBN 80-86056-94-5
- [16] VOŘÍŠEK, J. *Strategické řízení informačního systému a systémová integrace*. Praha: Management Press, 1997. ISBN 80-85943-40-9
- [17] VRANA, Ivan, RICHTA, Karel. *Zásady a postupy zavádění podnikových informačních systémů*. Praha: Grada Publishing, 2005. ISBN 80-247-1103-6
- [18] ŽID, N., BENÁČANOVÁ, H., KUNSTOVÁ, R. a SVOBODA, J. *Orientace ve světě informatiky*. Praha: Management Press, 1998. ISBN 80-85943-58-1

Internet:

- [19] *Certifikát počítačové gramotnosti ECDL* [online]. [cit. 2006-04-15] dostupný z WWW: <<http://www.gopas.cz/Clanek.aspx?IDClanku=193>>
- [20] FARANA, Radim. *Řešené problémy v MS-Access*. [online]. [cit. 2006-02-12] Dostupný z WWW: <<http://www.fs.vsb.cz/books/DBAcc/Kap01.html>>
- [21] *Informační zdroje a komunikace (kap.5 Bezpečnost dat)* [online]. [cit. 2005-12-05] Dostupný z WWW: <http://elis.mendelu.cz/metodika/ukazky/kapitola5.html>
- [22] KUČERA, Milan. *Bezpečnost Data Warehouse* [online]. [cit. 2005-12-05]. Dostupný z WWW: <<http://www.systemonline.cz/site/data-warehousing/bezpdw.htm>>
- [23] SOCHOR, J. *Analýza a návrh systémů* [online]. (Studijní materiály k předmětu PB007, podzim 2003). Fakulta informatiky MU Brno. [cit. 2006-01-22]. Dostupné na WWW: <<http://www.fi.muni.cz/~sochor/PB007/>>
- [24] URBÁNEK, Miloš. *IT SYSTÉM – Interní bezpečnost* [online]. 7-8/2002 Dostupný z WWW: <<http://www.systemonline.cz/site/bezpecnost/intbezp3.htm>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ADO	Active Data Object
API	Application Programming Interface
BaCo	Barum Continental spol. s r.o.
CASE	Computer Aided System Engineering
CD-ROM	Compact Disc Read-Only Memory
DAO	Data Access Objects
DB	DataBáze
DLL	Dynamic Linked Library
DW	DataWare
GUI	Graphical User Interface
HR	Human Resource
HRD	Human Resource Development
HTML	Hyper Text Markup Language
HW	HardWare
IS	Informační Systém
IT	Informační Technologie
MS	Microsoft
ODBC	Open Database Connectivity
OLE	Object Linking and Embedding
OW	OrgWare
PC	Personal Computer
PIN	Personal Identification number
PW	PeopleWare
SQL	Structured Query Language
SŘBD	Systém Řízení Báze Dat

SW	SoftWare
USB	Universal Serial Bus
URL	Uniform Resources Locator
VB	Visual Basic
VBA	Visual Basic for Application

SEZNAM OBRÁZKŮ

Obr. 1. Certifikát ECDL ve formátu kreditní karty	11
Obr. 2. Smart Card a iKey	30
Obr. 3. Struktura a návaznost jednotlivých částí projektu.....	34
Obr. 4. Struktura a umístění jednotlivých komponent aplikace	38
Obr. 5. Kontrola přístupnosti dat na serveru.....	39
Obr. 6. Doplnění osobních údajů s ověřením existence osobního čísla	41
Obr. 7. Zápis požadavku nového kurzu	44
Obr. 8. Datové toky přihlášky do nového kurzu.....	45
Obr. 9. Zápis do opakovaného kurzu přes intranet.....	46
Obr. 10. Datové toky přihlášky do volně přístupného kurzu.....	46
Obr. 11. Postup upřesňování seznamu účastníků	47
Obr. 12. Vztah mezi daty, informacemi a znalostmi	48
Obr. 13. IS jako podpora jednotlivých úrovní řízení.....	49
Obr. 14. Úprava aplikace podle přihlášeného uživatele	52
Obr. 15. Schéma importu dat z podnikového IS.....	56
Obr. 16. Aktualizace s uchováním historických dat	59

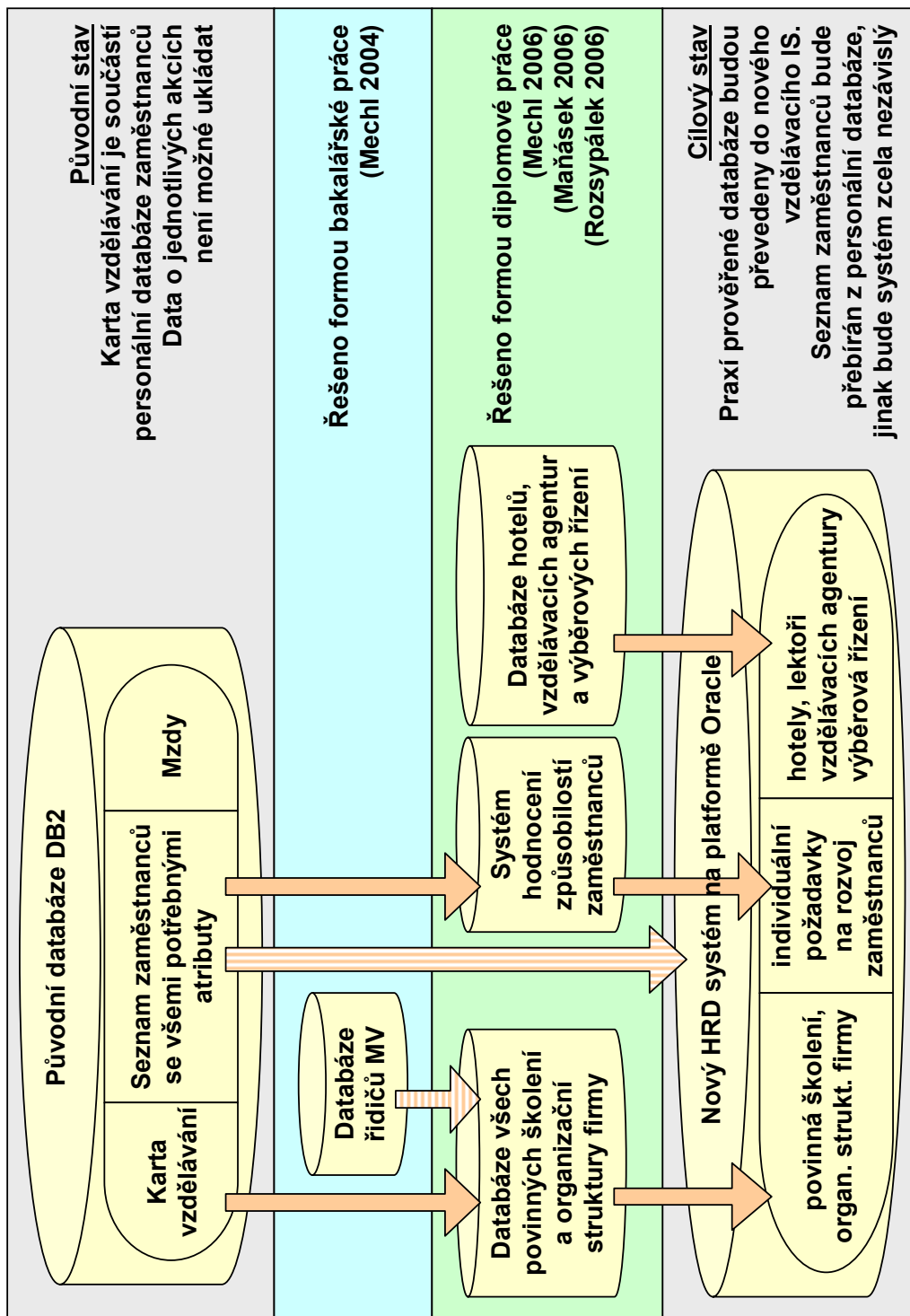
SEZNAM TABULEK

Tab. 1. Výhody a nevýhody obou přístupů k programování	16
Tab. 2. Konstanty parametrů příkazu DoCMD.OutputTo	20
Tab. 3. Konstanty pro parametry zprávy	22
Tab. 4. Návratové hodnoty po stisku tlačítka	22
Tab. 5. Časová osa projektu.....	36
Tab. 6. Ukázka tabulky zaznamenávající změny.....	53

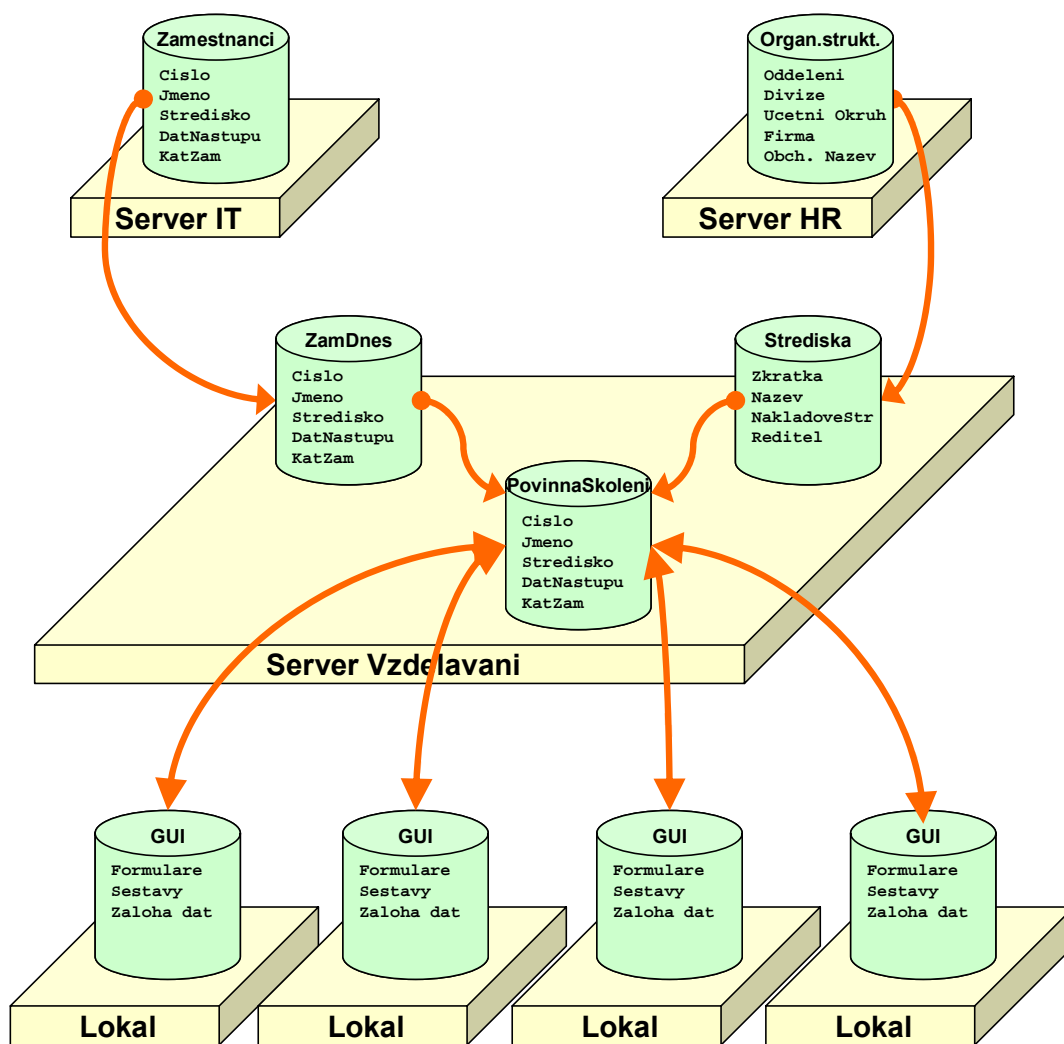
SEZNAM PŘÍLOH

Příloha P I:	Schématický přehled projektu	72
Příloha P II:	Schéma propojení dat z IS	73
Příloha P III:	Datová struktura jádra databáze	74
Příloha P IV:	Organizace Opakovaných školení	75
Příloha P V:	Přehled oblastí zahrnutých školení	76
Příloha P VI:	Úvodní procedury probíhající po spuštění aplikace	77
Příloha P VII:	Hlavní přepínací panel	78
Příloha P VIII:	Karta absolvovaných školení	79
Příloha P IX:	Karta výběru zaměstnance	80
Příloha P X:	Karta nominace do kurzu	81

PŘÍLOHA P I: SCHÉMATICKÝ PŘEHLED PROJEKTU

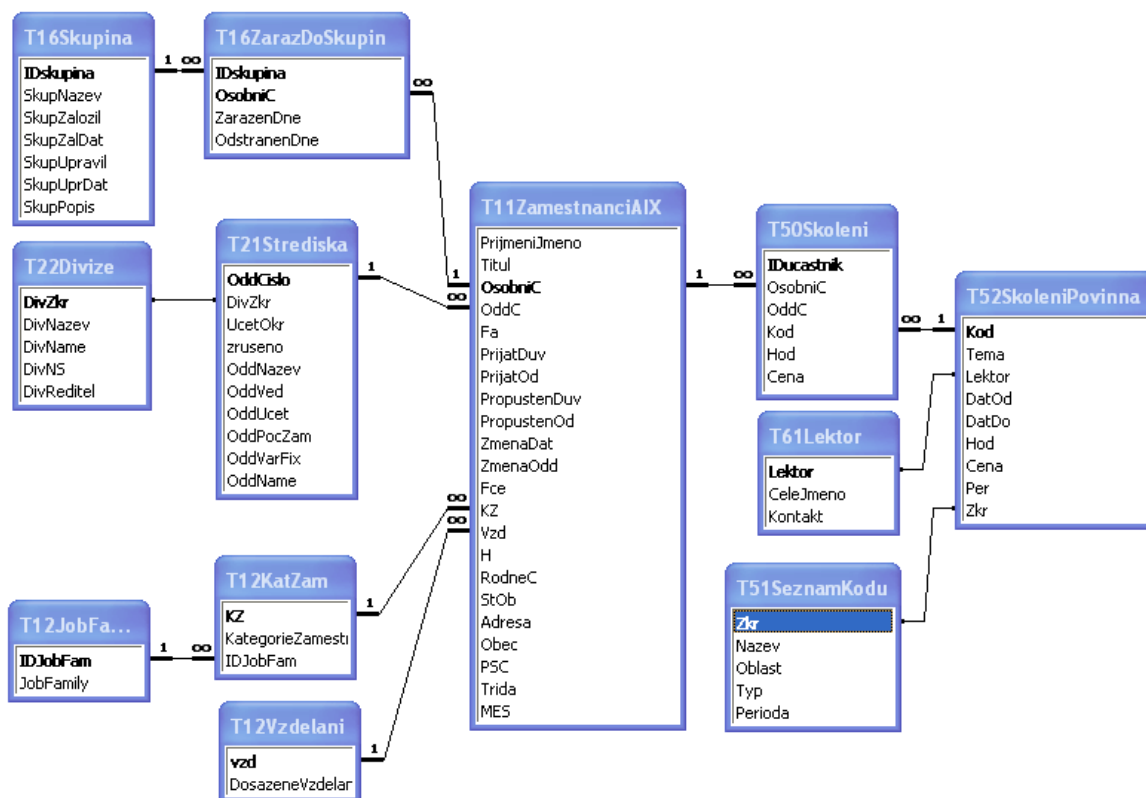


PŘÍLOHA P II: SCHÉMA PROPOJENÍ DAT Z IS



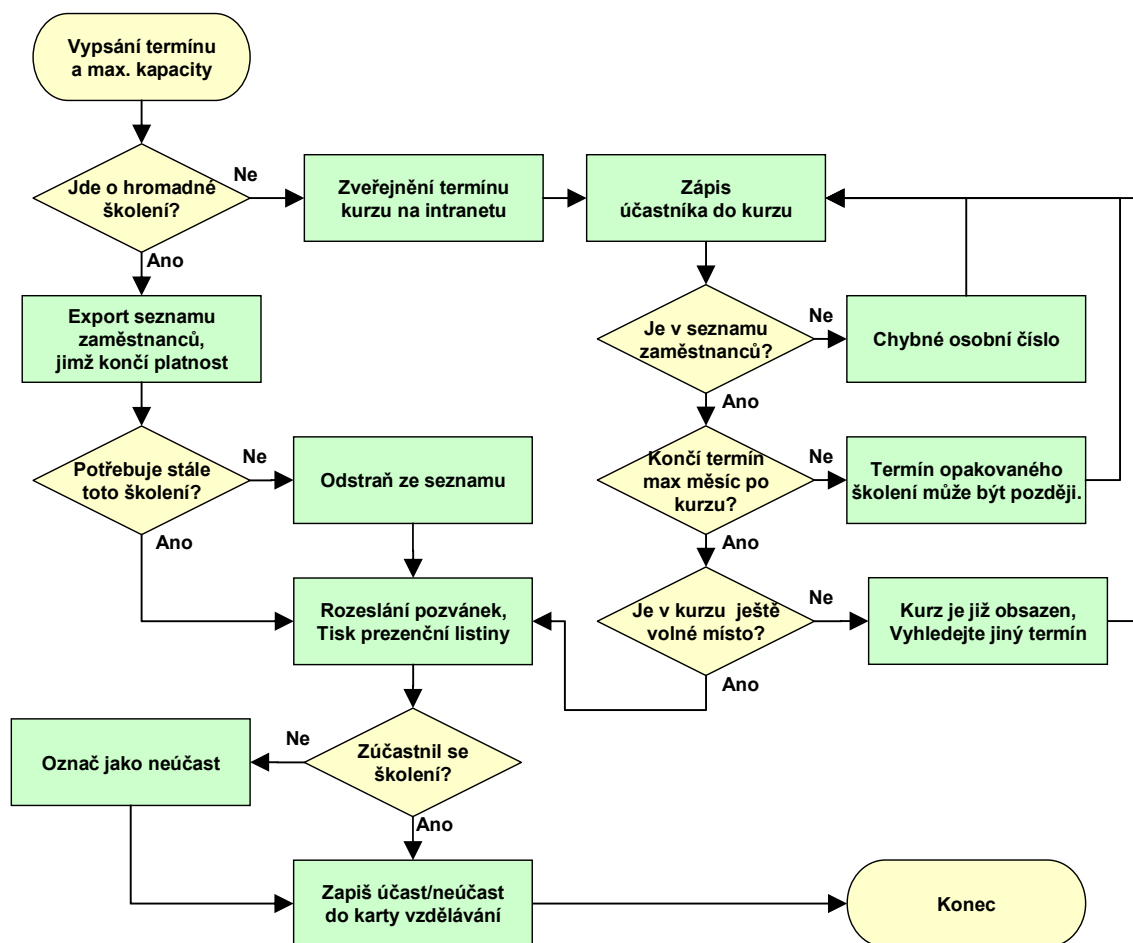
PŘÍLOHA P III: DATOVÁ STRUKTURA JÁDRA DATABÁZE

Zobrazeny jsou pouze tabulky trvale propojené relacemi (primární klíč je vyznačen tučně).



PŘÍLOHA P IV: ORGANIZACE OPAKOVANÝCH ŠKOLENÍ

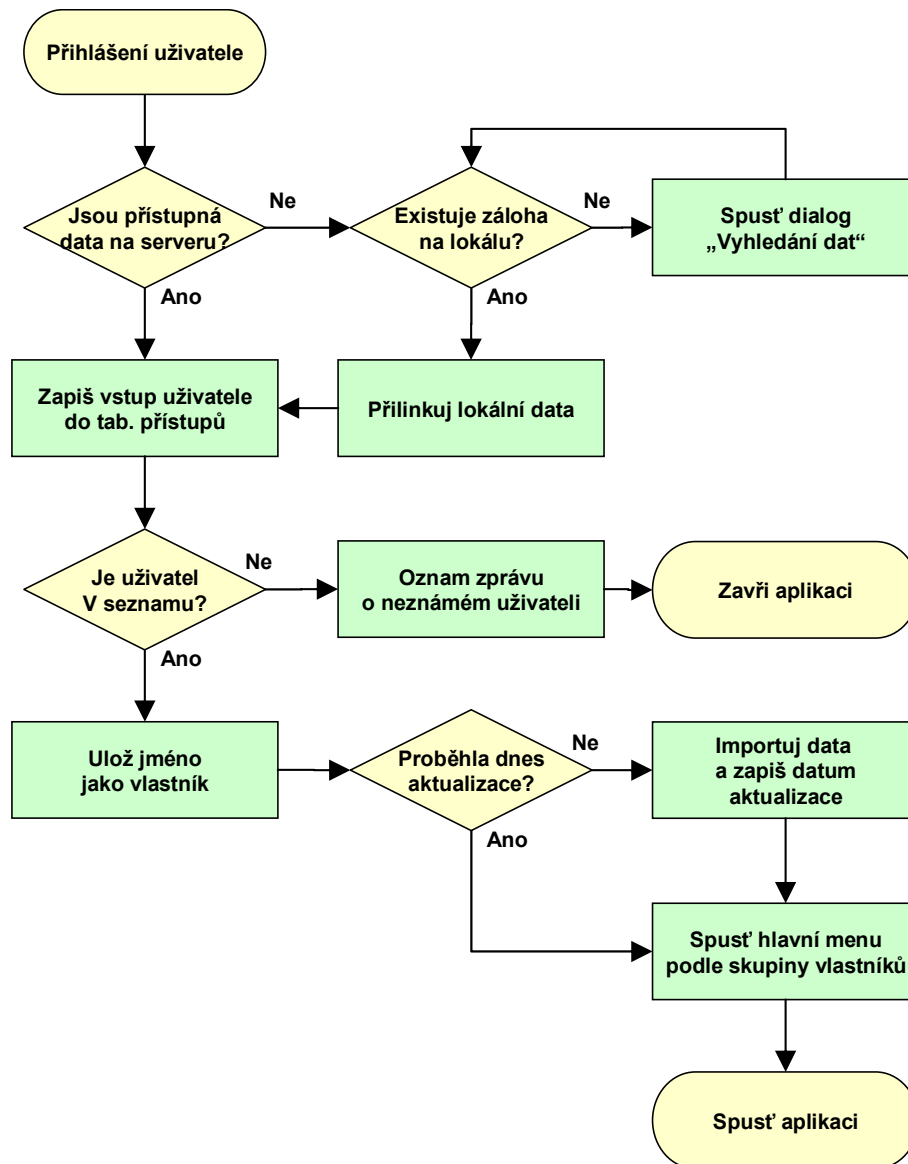
Postup od založení kurzu po evidenci účasti.



PŘÍLOHA P V: PŘEHLED OBLASTÍ ZAHRNUTÝCH ŠKOLENÍ

Kód	Oblast	Název	Perioda
Ao	Tlakové nádoby	Obsluhovatelé	3
Au	Tlakové nádoby	Uživatelé	3
Bn	Bezpečnost práce	Práce v prostředí nebezpečí výbuchu	3
Bo	Bezpečnost práce	Normy BHP (OHSAZ)	3
Bv	Bezpečnost práce	BHP pro vedoucí zaměstnance	3
Eh	Ekologie	Povodně a havarijní plán	1
En	Ekologie	Nebezpečné látky	1
Ep	Ekologie	Přeprava nebezp.látek a manipulace	1
Er	Ekologie	Likvidace úniku ropných látek	1
Ev	Ekologie	Ekologie pro vedoucí zaměstnance	1
Go	RTG	Obsluhy RTG a zářičů	3
Gr	RTG	Radiační ochrana	1
Ha	Hasiči	HZSP - Taktické řízení	5
Hh	Hasiči	Požární hlídky	1
Ht	Hasiči	HZSP - Hasičská technika	5
Hv	Hasiči	HZSP pro vedoucí zaměstnance	3
Lo	Kalandry	Školení kalandristů - opakované + přezk.	1
Lz	Kalandry	Školení kalandristů - základní	
Pp	Ostatní povinná	První pomoc	
Ps	Ostatní povinná	Svářeči	2
Pv	Ostatní povinná	Vstřelovači	3
Qm	Kvalita	Statistické metody (SPC, DOE)	1
Ra	ŘMV	Řidiči referentských vozidel	1
Ro	ŘMV	Opakované školení a přezkoušení	1
Ru	ŘMV	Uživatelé MV	2
Rz	ŘMV	Základní kurz ŘMV	1
Uv	Elektro	Elektro vyhláška 50/78 - vedoucí	3
Yp	Plyny	Technické plyny	
Yo	Plyny	Plynová zařízení - obsluhy	3
Zd	Zvedací zařízení	Dozorce výtahu	2
Zj	Zvedací zařízení	Jeřábník	2
Zp	Zvedací zařízení	Obsluha pracovních plošin	2
Zr	Zvedací zařízení	Řidič výtahu	2
Zu	Zvedací zařízení	Provozovatel (uživatel, pověř.osoba)	3
Zv	Zvedací zařízení	Vazač	2

PŘÍLOHA P VI: ÚVODNÍ PROCEDURY PROBÍHAJÍCÍ PO SPUŠTĚNÍ APLIKACE



PŘÍLOHA P VII:HLAVNÍ PŘEPÍNAČÍ PANEL

Nejdůležitější informace o aktuálně probíhajících kurzech a odkazy na jednotlivé oblasti a formuláře.

The screenshot displays the Microsoft Access application window titled "Microsoft Access - [Úvod Povinná školení BaCo verze 1.2 Povinná školení BaCo verze 1.2]". The interface is divided into several sections:

- Přehled oblastí školení**: A table listing training areas and their corresponding number of actions.
- Školení**: A panel with buttons for "Aktuální akce", "Všechny akce", "Proškolených měsíčně", and "Nejškolenější zaměstnanci".
- Tisk**: A panel with buttons for "Přehled školení", "Systémové formuláře", and "Zavřít formulář".
- Zaměstnanci**: A panel with buttons for "Organizační členění", "Seznam zaměstnanců", "Vyhledávání zaměstnanců", "Příjem a propuštění", and "Skupiny".
- Aktuální akce začínající do 24.5.2006**: A table showing current training actions starting by a specific date.
- Konec aplikace**: A button located at the bottom right of the dashboard area.

The date **10.5.2006** is displayed at the bottom of the dashboard. The Windows taskbar at the bottom shows the Start button, several open folders, and the Microsoft Access application window.

Kód	Oblast	Počet akcí
A	Tlakové nádoby	2
B	Bezpečnost práce	1
E	EŠŘ	17
G	RTG	0
H	Hasiči	0
L	Kalandry	6
N	Nový příjem	4
P	Ostatní povinná	17
Q	Kvalita	0
R	ŘMV	12
U	Elektro	0
Y	Plyny	0
Z	Zvedací zařízení	12

Kód	Téma	Lektor	Datum Od	Datum Do	Zkr.
Ro08	Opakované školení řidičů	Hlaváč	11.5.2006		Ro
Ps06	Svářeči	Horký	13.5.2006		Ps
Pp08	První pomoc	Zdravá	22.5.2006		Pp
Ro09	Opakované školení řidičů	Hlaváč	22.5.2006		Ro

PŘÍLOHA P VIII: KARTA ABSOLVOVANÝCH ŠKOLENÍ

Přehledné informace o absolvovaných školeních s údajem o vypršení platnosti (vyznačeno barevně).

Microsoft Access - [T11ZamestnanciAIX]

Soubor Úpravy Zobrazit Vložit Formát Záznamy Nástroje Okno Nápověda Základní kurzy Opakované školení Katalog MV

Nápověda – zadejte dotaz

Karta absolvovaných školení zaměstnance

Zaměstnanec	Novák Jan	Odděl. č.	24432	Přijat	1.5.1974
		Osobní č.	111127	Propuštěn	
Adresa		Kategorie	dělník variabilní		
Obec		PSC	Vzdělání	vyučen s výučním listem	

Datum	Kód	Téma	účasť	Hod	Cena	Oddělení	Pozv.	Lektor	Perioda	Platnost
12.2.2005	Lo01	Školení kalandristů - opakované	<input type="checkbox"/>			24432		Navrátil	1	12.2.2006
1.5.2005	Np01	Školení nových příjmů	<input type="checkbox"/>			24432		Světlíková		
3.4.2005	Pp03	První pomoc	<input type="checkbox"/>			24432		Zdravá		
16.2.2006	Qn02	Statistické metody (SPC, DOE)	<input type="checkbox"/>			24432		Přesný		
22.5.2006	Ro09	Opakované školení řidičů	<input type="checkbox"/>			24511		Hlaváč		
2.3.2006	Hh10	Požární hlídky	<input type="checkbox"/>	0	0 Kč			Hoflávý	1	2.3.2007

*

Formulářové zobrazení

FLTR NUM

Tisk Karty zaměstnance Export Karty zaměstnance

Start I:\#\Prenos\Muj... D:\Data vse\vikt... D:\Data vse\vikt... barevne schema... Microsoft Access ... CS 22:09

PŘÍLOHA P IX: KARTA VÝBĚRU ZAMĚSTNANCE

Na základě uvedených kritérií jsou vyhledány vyhovující záznamy. Kterýkoliv záznam je možno uložit do skupiny pro pozdější hromadné použití.

The screenshot shows the Microsoft Access application window titled "Výběr zaměstnanců Povinná školení BaCo verze 1,2". The interface includes a menu bar, a toolbar, and a main workspace divided into several sections.

1-tady zadejte kritéria výběru

Osobní číslo: [text box]
Příjmení Jméno: [text box]
Oddělení: 24401 [dropdown] 24426 [dropdown]
Přiját od/od: [text box]
Propuštěn od/do: [text box]

2-tady se zobrazí vyhovující záznamy (zaměstnanci)

Zaměstnanec	Středisko	Osobní č.	Přiját	Propuštěn
Burda Miroslav	24414	111253	1.11.1984	
Coufal Roman	24411	111113	1.3.1973	1.1.2002
Dobrovský Jindřich	24412	111251	1.9.1984	
Elefteriadu Jiří	24413	111115	1.5.1973	1.1.2002
Fusek Michal	24413	111252	1.10.1984	
Gottwald Pavel	24401	111112	1.2.1973	1.1.2002
Hrubý Miroslav	24412	111114	1.4.1973	1.1.2002
Ivánek Tomáš	24421	111391	1.5.1996	
Knitl Michal	24414	111390	1.4.1996	
Kretovič Milan	24401	111386	1.12.1995	
Lahola Lumír	24412	111388	1.2.1996	
Lhotský Tomáš	24401	111249	1.7.1984	
Mašláň Miroslav	24422	111118	1.8.1973	
Matulík Jan	24425	111120	1.10.1973	
Mazur Václav	24424	111119	1.9.1973	
Nemochovský František	24411	111387	1.1.1996	
Pavelka Pavel	24426	111121	1.11.1973	
Pelikán Martin	24421	111254	1.12.1984	
Pláňava Rostislav	24413	111389	1.3.1996	

Přehled členů skupiny

Detail	Zaměstnanec	Středisko	Dat. zařaz.	Dat.odstran
Detaily	Štolfa Jaroslav	26 100	3.5.2005	
Detaily	Vrzala Ladislav	28 200	3.5.2005	24.3.2006
Detaily	Škrabal Jaroslav	40 601	24.3.2006	8.5.2006
Detaily	Plichta Antonín	40 851	3.5.2005	8.5.2006
Detaily	Žůrek Vlastimil	40 821	24.3.2006	
Detaily	Knitl Michal	24 414	8.5.2006	
Detaily	Pelikán Martin	24 421	8.5.2006	
Detaily	Lhotský Tomáš	24 401	8.5.2006	
Detaily	Dobrovský Jindřich	24 412	8.5.2006	
Detaily	Elefteriadu Jiří	24 413	8.5.2006	

Buttons: Zařadit do skupiny, Zobraz detaily, Exportuj skupinu, Exportuj vyber, Založ skupinu

PŘÍLOHA P X: KARTA NOMINACE DO KURZU

Nominace probíhá jednotlivě, nebo hromadně na základě výběru skupiny. V některých případech je možno využít předem připravených seznamů požadavků získaných z intranetu.

Microsoft Access - [Povinná školení Povinná školení BaCo verze 1,2]

Soubor Úpravy Zobrazit Vložit Formát Záznamy Nástroje Okno Nápověda Základní kurzy Opakované školení Katalog MV Nápověda – zadejte dotaz

Tahoma 10 B I U

Povinná školení Eh

Kod	Eh09	Tema	Povodně a havarijní plán	DatOd	11.3.2006
Hod	8	Lektor	Plaváček	Cena	520,00 Kč
Perioda		DatDo			

Zaměstnanec	Oddělení	Osobní č.	hod	Kč	Pozv	Účast
► Kadavý Ladislav	27260	111444			ne	<input type="checkbox"/>
Čevela Milan	40701	111606			ne	<input type="checkbox"/>
Věrný Jan	40516	111343			ne	<input type="checkbox"/>
Suč Michal	26961	111297			ne	<input type="checkbox"/>
Dobrovský Jindřich	24412	111251			ne	<input type="checkbox"/>
Hubík Zdeněk	40515	111205			ne	<input type="checkbox"/>
Klabačka Miroslav	26960	111159			ne	<input type="checkbox"/>
Coufal Roman	24411	111113			ne	<input type="checkbox"/>
Horsák Petr	25240	111138			ne	<input type="checkbox"/>
Krýsa Jan	40418	111560			ne	<input type="checkbox"/>
Hruška Petr	40413	111468			ne	<input type="checkbox"/>
Mikel Jaroslav	26500	111420			ne	<input type="checkbox"/>
Košina Ivan	24427	111396			ne	<input type="checkbox"/>
Mikšík David	50418	111372			ne	<input type="checkbox"/>
Dvořák Stanislav	40712	111348			ne	<input type="checkbox"/>
Košina Jiří	28800	111324			ne	<input type="checkbox"/>
Zicháček Ondřej	26990	111300			ne	<input type="checkbox"/>
Dvorník Jaroslav	50425	111514			ne	<input type="checkbox"/>
*						<input type="checkbox"/>

Formulářové zobrazení

Nové školení Duplikuj školení Ulož akci Tiskni prezenčku Ulož účast Exportuj

FLTR NUM

Start I:\#\Prenos... D:\Data vse... D:\Data vse... barevne sch... Microsoft Ac... Adobe Phot... CS 21:40