

Vysoká škola ekonomická v Praze  
Fakulta informatiky a statistiky  
Vyšší odborná škola informačních služeb v Praze

Jiří Klíma

Databázová aplikace pro Call Centrum

Bakalářská práce

2007

Zadání:

Popište stávající SW užívaný v Call Centru CAC Leasing. Vymezte jeho hlavní funkční nedostatky. Navrhněte aplikaci, která by alespoň některé nedostatky funkcionality řešila.

Prohlášení

Prohlašuji, že jsem bakalářskou práci na téma „Databázová aplikace pro Call Centrum“ zpracoval samostatně a použil pouze zdrojů, které cituji a uvádím v seznamu použité literatury.

V Praze dne 25.května 2007

.....

Jiří Klíma

### Poděkování

Tímto děkuji vedoucí své bakalářské práce PhDr. Kateřině Julišové za odborné vedení, cenné rady a pomoc při jejím zpracování.

## Obsah:

Anotace.....	6
1. O společnosti.....	7
2. Call Centrum CAC Leasing .....	9
2.1 Call Centrum .....	9
2.2 Činnost Call Centra .....	10
2.3 Cílové skupiny Call Centra .....	10
2.4 Proces řízení hovoru s klientem .....	11
3. Analýza SW používaného v Call Centru .....	12
3.1 Obecné.....	12
3.2 Funkce aplikace.....	12
3.3 Práce s aplikací.....	13
3.4 Co databázová aplikace eviduje .....	17
3.5 Nedostatky aplikace .....	18
4. Požadavky na novou aplikaci.....	19
5. Použité technologie .....	20
6. Návrh databáze.....	22
6.1 SKUPINY.....	22
6.2 UŽIVATELÉ.....	22
6.3 Evidování hovorů .....	23
6.4 Oblasti, podtypy a specifika .....	24
7. Tvorba databáze .....	25
7.1 Datový model .....	25
7.2 Popis tabulek .....	26
7.3 Vazby .....	30
7.4 Práva.....	31
8. Aplikace .....	32
8.1 Komunikace MySQL a Borland Delphi.....	32
8.2 Přihlášení do aplikace.....	33
8.3 Hlavní formulář .....	34
8.4 Záložky hlavního formuláře .....	35
8.5 Export do Excelu.....	39

9.	Náklady na databázovou aplikaci .....	40
10.	Srovnání navrhnuté aplikace a původní aplikace .....	40
10.1	Práce s aplikací.....	40
10.2	Splnění požadavky na novou aplikaci .....	41
10.3	Přínosy navrhnuté aplikace .....	41
10.4	Srovnání stávající a navrhnuté aplikace .....	42
11.	Závěr.....	43
12.	Seznam obrázků a schémat .....	44
13.	Zdroje .....	45
13.1	Knížní dokumenty .....	45
13.2	Elektronické dokumenty .....	45

## Anotace

Tato bakalářská práce se zabývá aplikací pro evidenci hovorů Call Centra společnosti CAC Leasing. Část práce je věnována stávající aplikaci, která se používá v Call Centru. Analyzuje její užití, funkce, seznam položek, které eviduje, a specifikuje nevyhovující parametry a vlastnosti, které se negativně projevují v praxi.

Hlavní úkol práce je navrhnout novou databázovou aplikaci, která bude řešit zjištěné konkrétní nedostatky s ohledem na maximální zvýšení efektivity práce uživatelů a minimální náklady.

Nová databázová aplikace je rozdělena do tří částí. První část je „Návrh databáze“, kde jsou vymezeny skupiny uživatelů a položky, které chceme evidovat. Druhá část „Tvorba databáze“ se zabývá fyzickým uspořádáním dat v databázi, popisem jednotlivých tabulek a vzájemných vazeb mezi nimi. V poslední části „Aplikace“ je popsána aplikace pro komunikaci s databází. Prostřednictvím uživatelského rozhraní této aplikace mohou uživatelé do databáze zapisovat a vybírat konkrétní data, která mohou být exportována do Excelu.

# 1. O společnosti



*Obchodní jméno :* **CAC LEASING, a.s.**

*IČO :* **15886492**

*Sídlo :* **Praha 5, Radlická 14/3201, PSČ 15000**

*Datum vzniku :* **27.2.1991**

*Právní forma :* **Akciová společnost**

Společnost byla založena v roce 1991 Komerční bankou, a.s., největší bankou České republiky, a nejvýznamnější rakouskou bankou Creditanstalt Bankverein AG. Na základním jmění společnosti se Komerční banka podílí 50%, Bank Austria Creditanstalt International Wien AG 15% a její dceřinná společnost CA Leasing Wien GmbH 35%. CAC LEASING patří mezi stabilní leasingové společnosti s profesionálními přístupy a silným kapitálovým zázemím, vysokou úrovní produktivity práce a trvale zdravou ekonomickou situací s dostatkem odpovídajících rezerv a opravných položek.

Hlavním předmětem činnosti společnosti je finanční leasing a ostatní druhy leasingu. Poskytované služby zahrnují tři hlavní oblasti: leasing osobních, užitkových a nákladních automobilů a ostatní dopravní techniky, leasing strojů a zařízení a odbytový leasing. K úplné nabídce patří leasing nemovitostí, který je poskytován ve spolupráci s Bank Austria Creditanstalt Leasing Real Estate. Nabídka společnosti CAC LEASING, a.s. (dále jen CAC LEASING) je určena pro soukromé i právnické osoby.

Z forem leasingu mohou klienti využít klasického finančního leasingu, případně zpětného leasingu, pro nákup většího počtu vozidel, výpočetní techniky, kopírovacích strojů a dalších dobře prodejných komodit i operativního leasingu. Odbytový leasing poskytuje smluvním partnerům CAC LEASING, především výrobcům, případně prodejcům strojů a zařízení, možnost nabízet financování nákupu jejich produktů prostřednictvím leasingu jako samozřejmou součást nabídky. Společnost v současné době provozuje síť deseti poboček a šesti kanceláří. Tato struktura zaručuje klientům snadnou přístupnost a plné pokrytí z jakéhokoliv místa v České republice. Většina obchodů je uzavírána přímo na jednotlivých



pobočkách. Akciová společnost CAC LEASING je významným členem Asociace leasingových společností ČR, členem Svazu průmyslu a dopravy ČR, členem Hospodářské komory ČR, členem Česko-německé obchodní a průmyslové komory a zastupuje Českou republiku v mezinárodních leasingových klubech Multilease a Leaseclub. [27]

## 2. Call Centrum CAC Leasing

### 2.1 Call Centrum

Call Centrum společnosti CAC Leasing vzniklo v prosinci 2002. Důvodem vzniku bylo zlepšení kvality hovorů a dostupnosti na telefonních číslech, neboť každým rokem narůstá vytížení zákaznických linek. V roce 2005 mělo Call Centrum 4 operátory a 1 brigádníka, kteří vyřizovali pouze hovory týkající se linky CAC Leasing a linky Renault Leasing. Denně bylo vyřízeno přibližně 200 hovorů.

Od roku 2006 začalo Call Centrum sloužit i jako podpora pro klienty dceřinných společností CAC Leasing. Počet linek vzrostl na pět a počet hovorů přibližně na 400 denně. Počet zaměstnanců v Callcentru se zvýšil na 6 a 2 brigádníky.

Kontaktní centrum je jednotná komunikační brána do společnosti – jedná se o automatizovanou a integrovanou obsluhu všech komunikačních kanálů. Ať již zákazník do společnosti telefonuje, posílá e-mail, fax, dopis či přímo zajde na pobočku, bude vždy rychle a prvotřídně obsloužen, vždy mu budou podány stejné a konzistentní informace, jeho požadavky budou vždy vyřízeny. Z každého takového zákaznického kontaktu se také sama firma poučí – zjistí, jak často ji určití zákazníci kontaktují, jakými kanály, z jakého důvodu, s jakou úspěšností řešení atd. Díky tomuto automatickému procesu vzniká databáze těch nejcennějších zákaznických informací, které jsou velmi dobře využitelné při navazujících marketingových a prodejních procesech. Pokud tyto informace zanalyzujeme, můžeme kontaktní centrum používat i druhým směrem – tzn. při aktivní komunikaci ven, směrem k zákazníkům. Budeme vědět, co chtějí – jejich preference, budeme tedy vědět i čím je oslovit. Zjistíme, jaké preferují komunikační kanály a proto budeme schopni určit, zda jim zavolat, či poslat e-mail. Díky automatizaci pak budeme schopni tyto aktivity provozovat ven i dovnitř s velkým počtem zákazníků najednou.

## 2.2 Činnost Call Centra

Call Centrum slouží jako zdroj informací pro současné i budoucí klienty, externí firmy spolupracující s CAC Leasing i pro společnost samotnou. Pracovníci Call Centra denně zodpoví velké množství hovorů týkajících se leasingu (např. plateb, smluv, pojištění, nového leasingu atd. ) nebo sdělí kontakt na konkrétní osobu, která dokáže vyřešit problém klienta.

Tyto hovory jsou zaznamenávány a slouží jako cenný zdroj informací. Společnost tím získá informace o spokojenosti, či nespokojenosti zákazníků. Zjistí jejich přání a požadavky, na jejichž základě se může přizpůsobit potřebám klientů.

## 2.3 Cílové skupiny Call Centra

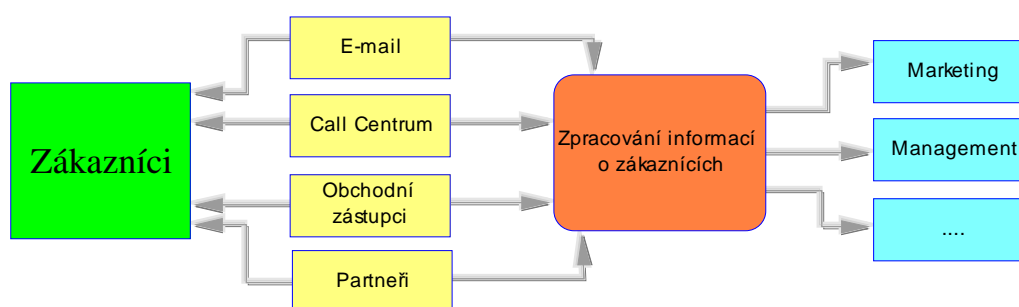
Vnější uživatelé:

- Současní zákazníci CAC Leasing
- Potenciální zákazníci CAC Leasing
- Partneři CAC Leasing

Vnitřní uživatelé:

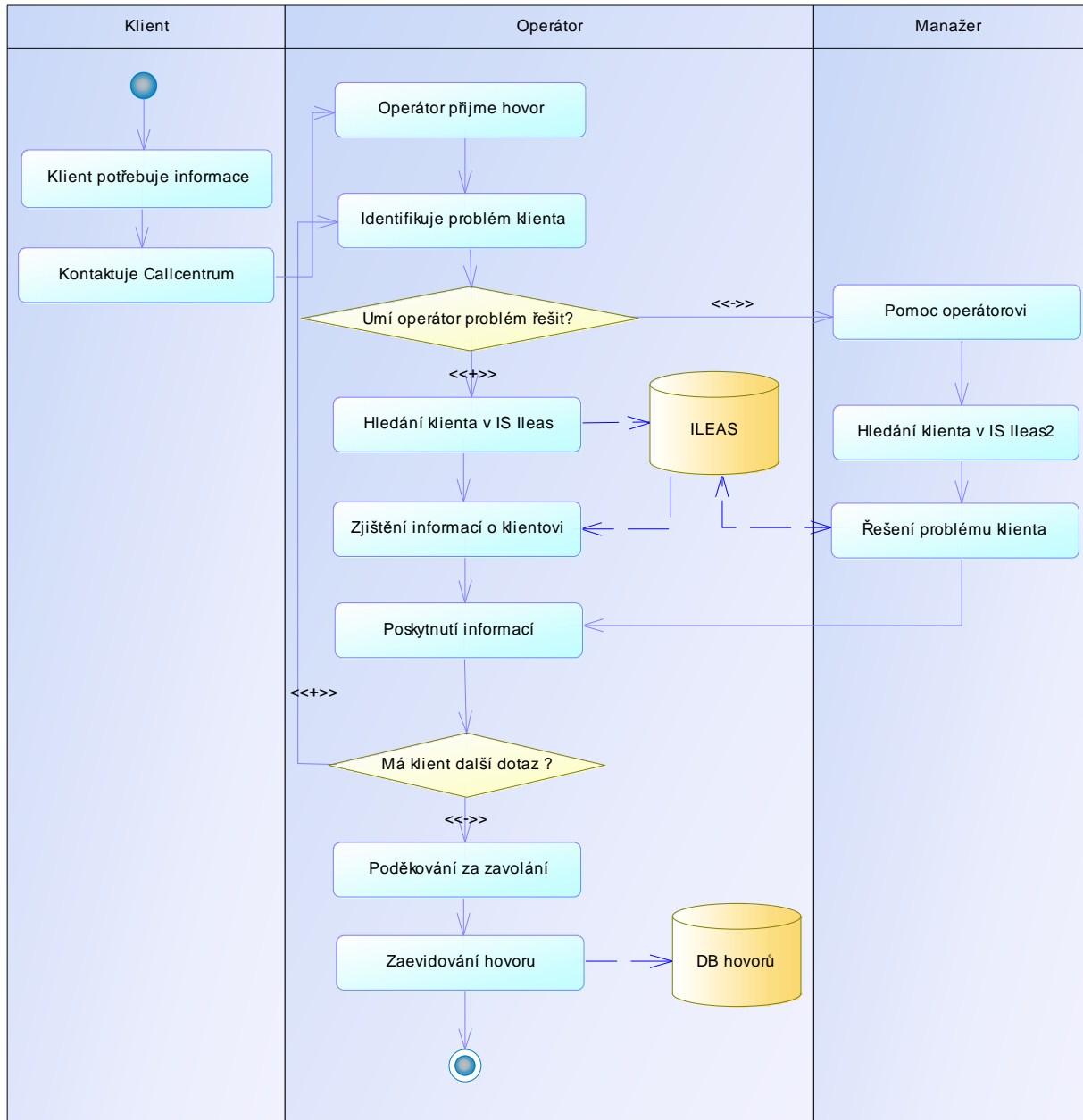
- Zaměstnanci CAC Leasing
- Dceřinné společnosti CAC Leasing

Zákazníci komunikují se společností pomocí e-mailu, telefonu, obchodních zástupců i partnerů. Všechna shromážděná data se zpracují do podoby, která je dále využitelná pro Marketing, Management nebo jiná oddělení.



Obrázek 2.1. Schéma procesu zákaznické péče a podpory

## 2.4 Proces řízení hovoru s klientem



Obrázek 2.2. Proces řízení hovoru s klientem

Klient má potřebu vyřešit nějaký problém nebo jen získat více informací. Kontaktuje zákaznické centrum a operátor se pokusí vyřešit jeho problém, případně podat co nejvíce informací. Pokud operátor nedokáže problém vyřešit nebo nemá dostatek informací, má možnost se zeptat manažera, který mu poradí nebo vyřeší problém za operátora. Po zodpovězení veškerých dotazů klienta, operátor hovor zaeviduje do databáze hovorů a vyčkává na další hovor nebo zodpovídá emailové dotazy.

## 3. Analýza SW používaného v Call Centru

### 3.1 Obecné

Typ databáze: DB2

Platforma: AIX

Aplikace komunikuje s databází po IP

Aplikace vznikla v roce 2003 a vytvořila ji firma Deloitte & Touche, která později zanikla. Od roku 2003 prošlo oddělení Call Centra mnoha změnami a aplikace se stala velice nepraktická. Nebyla od roku 2003 aktualizována a v dnešní podobě již nevyhovuje potřebám Call Centra a má mnoho nedostatků.

Na vlastní aplikaci se nevztahují žádné licenční poplatky.

### 3.2 Funkce aplikace

Funkce aplikace je možné rozdělit do několika okruhů:

- Služby pro uživatele

Zahrnují přihlášení a odhlášení uživatele včetně změny hesla. Dále obsahuje možnost simulace hovoru, což znamená průchod hovorem bez zákazníka.

- Podpora operátora

Podpora operátora je při hovoru se zákazníkem hlavním účelem celé aplikace. Operátor je přitom veden logikou scénářů podle reakcí zákazníka. Na konci záznamu hovoru provádí operátor vyhodnocení úspěšnosti kontaktu se zákazníkem i jednotlivých záležitostí, které byly řešeny.

- Přehledy

Přehledy poskytují seznamy přihlášených uživatelů, hovorů a jednotlivých případů včetně informací o nich. Informace o případě obsahuje atributy a parametry případu a kromě toho také průběh případu, což je seznam scénářů přesně v pořadí, v jakém byli zapisovány.

- Správa uživatelů

Tato část aplikace umožňuje vytváření a editaci uživatelů, včetně přiřazení přístupových práv. Do správy uživatelů má přístup pouze administrátor.

### 3.3 Práce s aplikací

Hlavním účelem aplikace je podpora operátora Call Centra při vedení telefonického hovoru se zákazníkem, který chce vyřešit určitý problém, případně i více problémů. Případy řeší operátoři se zákazníky a měli by postupovat podle pracovních postupů, které jsou implementovány v aplikaci jako scénáře. Vzhledem k neustálému rozvoji kvantity i kvality služeb společnosti CAC Leasing se dnes podle těchto scénářů už postupovat nedá. S každým klientem je třeba řešit problém individuálně a přesný postup daný scénáři je nepoužitelný. Operátoři řeší problémy díky svým odborným znalostem, podle konkrétních potřeb klienta. Bohužel, aby byl každý hovor správně zapsán do databáze, musí operátor projít celým scénářem a vybírat z obecných možností až se propracuje k určitému detailu, který potřebuje zaznamenat. Tento postup má jedinou výhodu a to je naprosto přesné popsání jak hovor postupoval a zaznamená velké množství detailů. Bohužel tyto detaily není třeba evidovat, jsou nadbytečné a operátory pouze časově zatěžují.

#### Přístupová práva

Přístupová práva jsou určena dvěma atributy:

- Zařazení do role
- Povolení vést hovory

V aplikaci jsou definovány následující role:

a) Administrátor

Má přístup do všech voleb aplikace bez omezení a v přehledech vidí všechny uživatele.

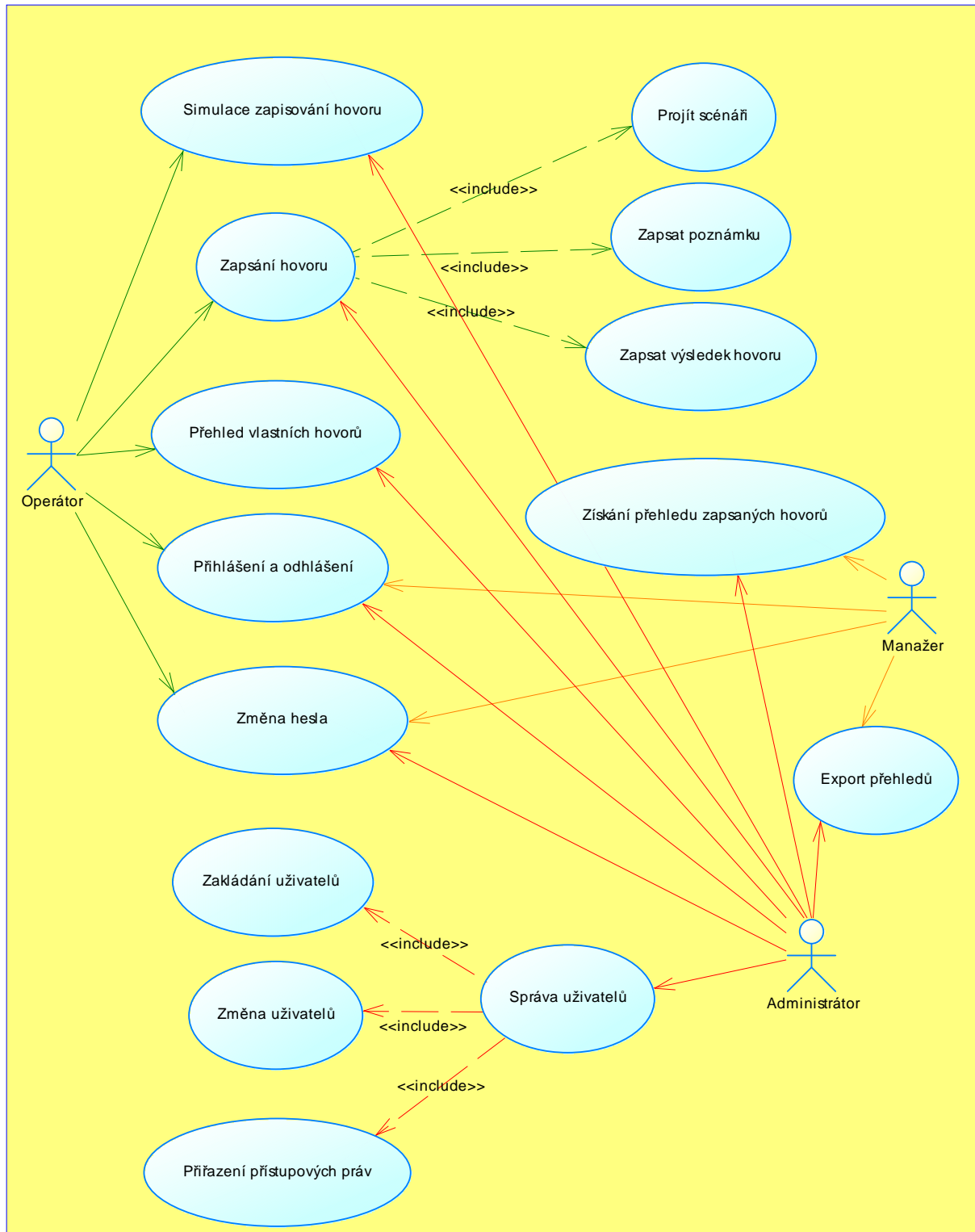
b) Manažer

Má přístup do všech voleb mimo správy uživatelů a zapisování hovorů.

c) Operátor

Má přístup pouze k záznamu hovorů a v přehledech vidí pouze data, která zapsal.

Každý uživatel Call Centra je zařazen do jedné z uvedených rolí. Přřazení přístupových práv a jejich aktualizaci provádí pouze administrátor.



Obrázek 3.1. Use Case diagram SW používaného v Call Centru

### Přihlášení

Uživatel, který byl vytvořen administrátorem, se musí přihlásit k aplikaci pod svým přihlašovacím jménem a heslem. Dále může užívat aplikaci pouze v rozsahu svých přístupových práv. Pokud s aplikací nepracuje déle než 10 minut, aplikace ho automaticky odhlásí a musí se přihlásit znovu.

### Proces zapisování hovoru

Operátor musí zjistit, jaká oblast problematiky bude v případě řešena. V každé z těchto oblastí se nachází seznam možností, ze kterých operátor vybere tu, která odpovídá dotazu zákazníka a dále se zobrazí seznam dalších konkrétnějších možností. Tímto způsobem operátor musí projít skrz jednotlivé scénáře hovoru až ke konkrétnímu detailu. Na závěr vyplní, zda byl klient spokojen a musí napsat poznámku k hovoru. Tuto poznámku musí zkopírovat do dalšího okna a kliknout na OK. Hovor se запиše do databáze. Při této operaci aplikace často vykazuje neznámou chybu a celý hovor je nutno zapsat znovu.

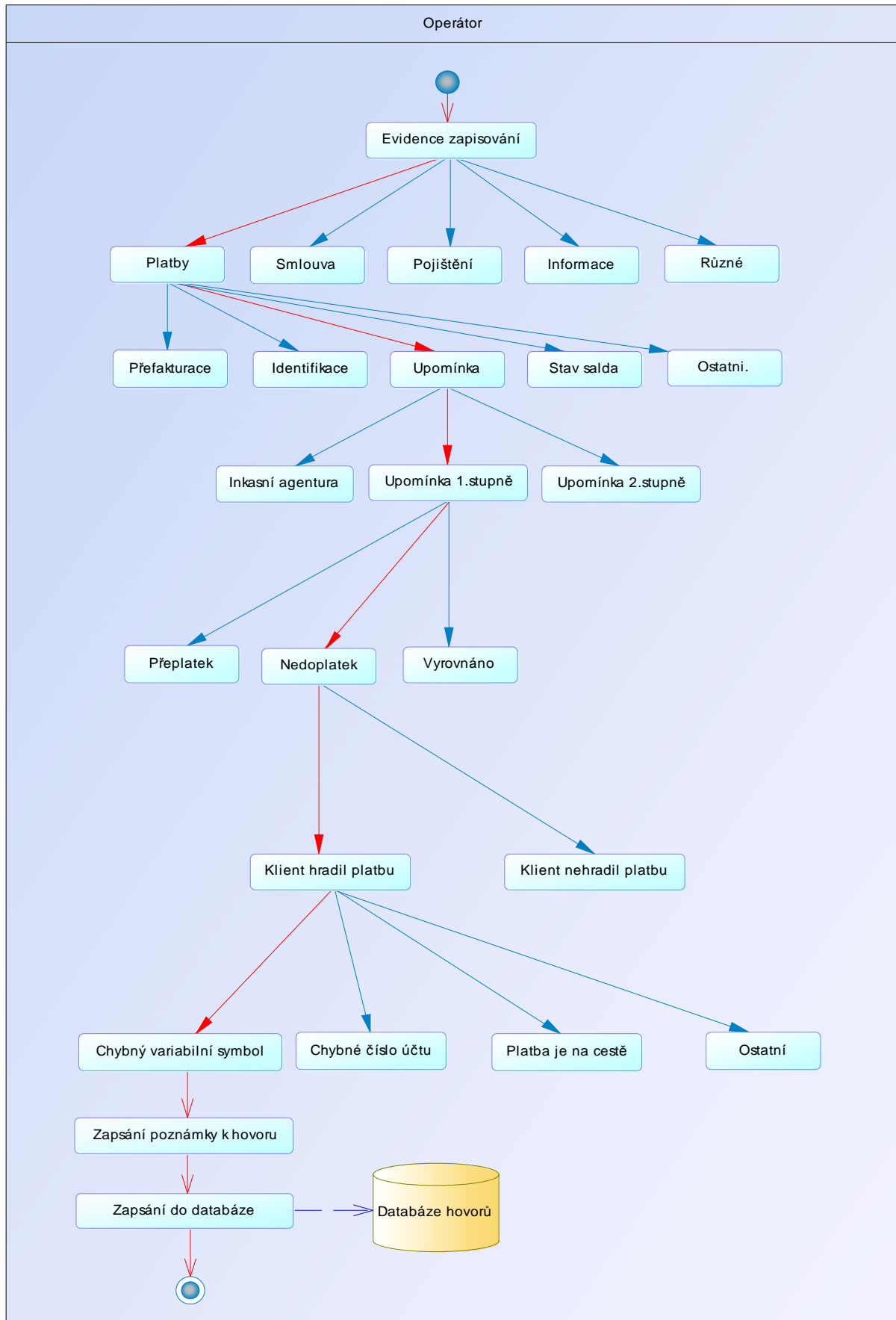
### Příklad hovoru a jeho zapsání

Klientovi přišla upomínka, protože neuhradil splátku. Operátor si ověřil stav v informačním systému a zjistil, že platba od klienta opravdu není na účtu společnosti. Klient poukazuje na to, že platbu uhradil a nadiktuje operátorovi všechny údaje z platebního příkazu. Po zkontrolování těchto údajů operátor zjistil, že klient platbu opravdu uhradil, ale pod chybným variabilním symbolem. Platba tedy je na účtu společnosti, ale není zapsána pod smlouvu klienta. Na základě těchto skutečností klient doloží potřebné doklady a platba bude převedena pod správný variabilní symbol.

Po skončení hovoru musel operátor hovor zapsat podle uvedeného obrázku 3.1. Nejdříve vybere, že se hovor týkal „Platby“ a načtou se další možnosti z kterých operátor vybere „Upomínka“. Takto postupuje až ke k poslední položce, která je „Chybný variabilní symbol“. Po kliknutí na poslední položku se objeví okno, kde musí operátor vyplnit poznámku a zkopírovat ji do druhého okna. Vyplní zda byl klient spokojen či nespokojen a klikne na tlačítko „OK“.

Průchod mezi jednotlivými možnostmi je proveden načtením stránky Exploreru a operátor musí na jejich načtení opakovaně čekat. Počet načítání stránky Exploreru se odvíjí od počtu úrovní zapisovaného hovoru a v uvedeném případě bylo třeba stránku aktualizovat devětkrát.





Obrázek 3.2. Postup zapisování hovoru v aplikaci

### Správa uživatelů

Administrátor má možnost vytvořit nové uživatele, zrušit uživatele a editaci uživatelů.

Atributy uživatele jsou:

- Login
- Jméno a příjmení uživatele
- Heslo
- Role
- Zapisování hovorů (ANO/NE)
- Přístup od – do
- Zakázán přístup (ANO/NE)

### Přehled hovorů

Každý uživatel se může podívat na přehled hovorů dle svého uživatelského oprávnění. Administrátor a manažer vidí všechny hovory na rozdíl od operátora, který vidí přehled jen svých vlastních hovorů.

#### 3.4 Co databázová aplikace eviduje

- Počet zapsaných hovorů
- Datum a čas
- Uživatele
- Číslo leasingové smlouvy
- Délku zapisování
- Nájemce
- Kompletní cestu zapisování hovoru
- Poznámku
- Spokojen / nespokojen

Není třeba zaznamenávat: nájemce (číslo smlouvy je dostatečné), délku zapisování, kompletní scénář hovoru.

### 3.5 Nedostatky aplikace

#### Přihlašování

- Pokud 10 minut uživatel nezapíše žádný hovor, aplikace ho automaticky odhlásí.

#### Zapisování hovorů

- Pro úspěšné zapsání musí uživatel projít skrz scénáře hovoru až k detailu hovoru.
- Aplikace eviduje velké množství detailů, které není třeba sledovat.
- S každým klientem se řeší problém individuálně a z tohoto důvodu jsou scénáře nevyhovující.
- Při zapisování hovoru aplikace vykazuje občasné chyby. Proč se v aplikaci vyskytují tyto chyby by pravděpodobně dokázal odpovědět pouze její, dnes už zaniklý, dodavatel. Pokud se tato chyba objeví, uživatel musí zapsat celý hovor znovu od začátku.
- Velké prodlevy při načítání jednotlivých webových stránek při průchodu scénáři.
- Na závěr hovoru musí být vyplněna poznámka a to i v případě, kdy je problém jednoznačně identifikován průběhem hovoru.
- Vyplněná poznámka se musí zkopírovat do druhého okna na stejné stránce. Proč je toto nutné by pravděpodobně věděl pouze autor.

#### Přehled hovorů

- Opět se občas objevuje chyba a načítání přehledu musí začít znovu.
- Načítání přehledu všech hovorů zapsaných za určité období je velmi dlouhé.

#### Úpravy aplikace

- Do aplikace není možné přidávat nové detaily hovoru a nelze odstranit ty, které se již nepoužívají.

Zapisování hovorů je nejběžnější činností operátorů. Jednotlivé hovory, které vyřídí, se uloží do databáze na serveru. Celý proces zapisování značně prodlužuje zaevidování hovoru a při větším počtu hovorů přicházejících rychle za sebou, není dostatek času na zapsání již vyřízených hovorů. Odstraněním těchto nedostatků by se zvýšila efektivita pracovníků Call Centra.

## 4. Požadavky na novou aplikaci

Při specifikaci požadavků na novou aplikaci vycházím z nedostatků stávající aplikace, které by měly být odstraněny.

- Aplikace nebude uživatele odhlašovat při delší nečinnosti.
- Zapsání hovoru by mělo být co nejrychlejší a jednoduché.
- Aplikace bude zaznamenávat v evidenci hovoru:
  - Datum
  - Čas
  - Uživatele
  - Skupinu do které uživatel patří
  - Číslo leasingové smlouvy
  - Oblast problému, typ problému a specifikaci problému
  - Poznámku (pokud je třeba)
  - Výsledek hovoru (Spokojen / nespokojen)
- Uživatelé budou mít přístup k přehledu hovorů, které sami zapsali.
- Administrátor bude mít přístup k celé databázi a bude moci vytvářet a blokovat uživatele
- Aplikace musí být odladěna takovým způsobem, aby se předešlo chybám.
- Načtení přehledu hovorů by mělo být provedeno během několika málo vteřin.
- Pomocí aplikace musí být možné jednoduše exportovat databázi nebo část databáze do Excelu.
- Náklady na databázovou aplikaci musí být minimální.
- Co nejrychlejší možnost zápisu hovorů, které se opakují mnohokrát za den.
- Aplikace musí být flexibilní, aby se dala kdykoli přizpůsobit požadovaným podmínkám. (tj. musí být možnost lehce přidávat či odebírat oblasti, typy i specifika)

## 5. Použité technologie

### MySQL 5.0

MySQL je Open Source systém, který vytvořila švédská firma MySQL AB, pro správu relační databáze. Popularita MySQL je obrovská a stále silně roste. Má řadu dobrých vlastností, mezi které patří například:

- *Rychlost*
- *Snadné používání (MySQL je vysoce výkonný, ale relativně jednoduchý databázový systém a jeho konfigurace a správa jsou mnohem méně složitější, než je tomu u velkých systémů.)*
- *Podpora dotazovacího jazyka (MySQL mluví jazykem SQL, což je jazyk, který si zvolily všechny moderní databázové systémy.*
- *Způsobnost (K serveru se současně může připojit mnoho klientů. Klienti mohou využívat simultánně několik databází. MySQL podporuje rozličná programovací rozhraní pro jazyky jako jsou C, Perl, Java, PHP a Python)*
- *Připojitelnost a zabezpečení( MySQL je plně síťový a k databázím lze přistupovat odkudkoli z internetu, takže je možné sdílet data kdekoli a s kýmkoli. MySQL podporuje zašifrované připojení pomocí protokolu SSL)*
- *Přenositelnost (MySQL běží na mnoha variantách UNIXu i na mnoha jiných systémech , než je UNIX, například Windows a OS/2.*
- *Má malé nároky na velikost pevného disku i paměti*
- *Dostupnost a cena (MySQL je Open Source, takže je dostupný zadarmo v licencích GPL GNU. Znamená to, že je MySQL zadarmo pro většinu použití)*

[7, s. 14]

Tuto databázi jsem zvolil, protože má výše zmíněné vlastnosti a nízké náklady. Tento databázový systém je zdarma, rychlý a stabilní. Na internetu je volně ke stažení MySQL Reference Manual, který obsahuje SQL příkazy včetně jejich vysvětlení a použití.

### MySQL administrator 5.0

Nástroj MySQL administrator je určen pro spravování databázového serveru (správa uživatelů, zálohování a obnovování databáze). Je možné ho zároveň použít pro vytváření databází a získání přehledu o databázi. Tímto programem můžeme vytvářet tabulky, klíče, vazby... v přijatelném rozhraní. Zároveň můžeme databázi naplnit daty. Při každé změně databáze program vygeneruje SQL příkaz a po potvrzení ho provede.

### Borland Delphi 7.0

Delphi je integrované grafické vývojové prostředí firmy Borland, určené pro tvorbu aplikací na platformě MS Windows v jazyce Object Pascal (objektové nástavbě Pascal). Obsahuje systém RAD (Rapid Application Development), který umožňuje vizuální návrh grafického uživatelského rozhraní, na jehož základě je automaticky vytvářena kostra zdrojového kódu, což výrazně urychluje vývojový cyklus. Programování v Delphi je z velké části založeno na použití komponent. [26]

### DAC for MySQL

*DAC for MySQL (MySQLDAC) je balík výkonných komponent pro tvorbu Delphi/C++Builder aplikací s přímým přístupem k databázím MySQL, bez vyžití služeb BDE a ODBC. Další přínos je podpora BLOB polí.[27]*

Propojení aplikace naprogramované v Borland Delphi a databáze MySQL je použit program DAC for MySQL od firmy microOLAP Technologies LTD. Při spuštění Delphi aplikace se aplikace automaticky připojí k databázi MySQL a prostřednictvím aplikace přenáší SQL příkazy do databáze, kde se provedou.

### OLE

„Object Linking and Embedding“ je technologie, která umožňuje vkládat a propojovat objekty. Pomocí OLE můžeme prostřednictvím jedné aplikace manipulovat s jinou aplikací. V aplikaci je tato technologie použita pro export dat do Excelu, kde je mohou uživatelé dále zpracovávat.

## 6. Návrh databáze

V databázi budeme chtít evidovat hovor vyřízený operátorem včetně konkrétních detailů, uživatele a skupinu do které uživatel patří.

### 6.1 SKUPINY

Každý uživatel musí patřit do konkrétní skupiny. Podle toho do jaké skupiny bude uživatel přidělen, taková bude mít práva. Skupiny jsou následující : OPERÁTOR, MANAŽER, ADMINISTRATOR. Jsou dané programátorem a nedají se měnit.

OPERÁTOR má oprávnění:

- zapisovat hovory do databáze
- podívat se na přehled hovorů, které zadal

MANAŽER má oprávnění:

- vyhledávat data v databázi
- exportovat vybraná data do Excelu

ADMINISTRÁTOR má oprávnění:

- manažera i operátora
- vytvářet uživatele
- blokovat uživatele
- přizpůsobovat databázi podle současných požadavků
- mazat vybrané záznamy z databáze

### 6.2 UŽIVATELÉ

U každého uživatele chceme evidovat :

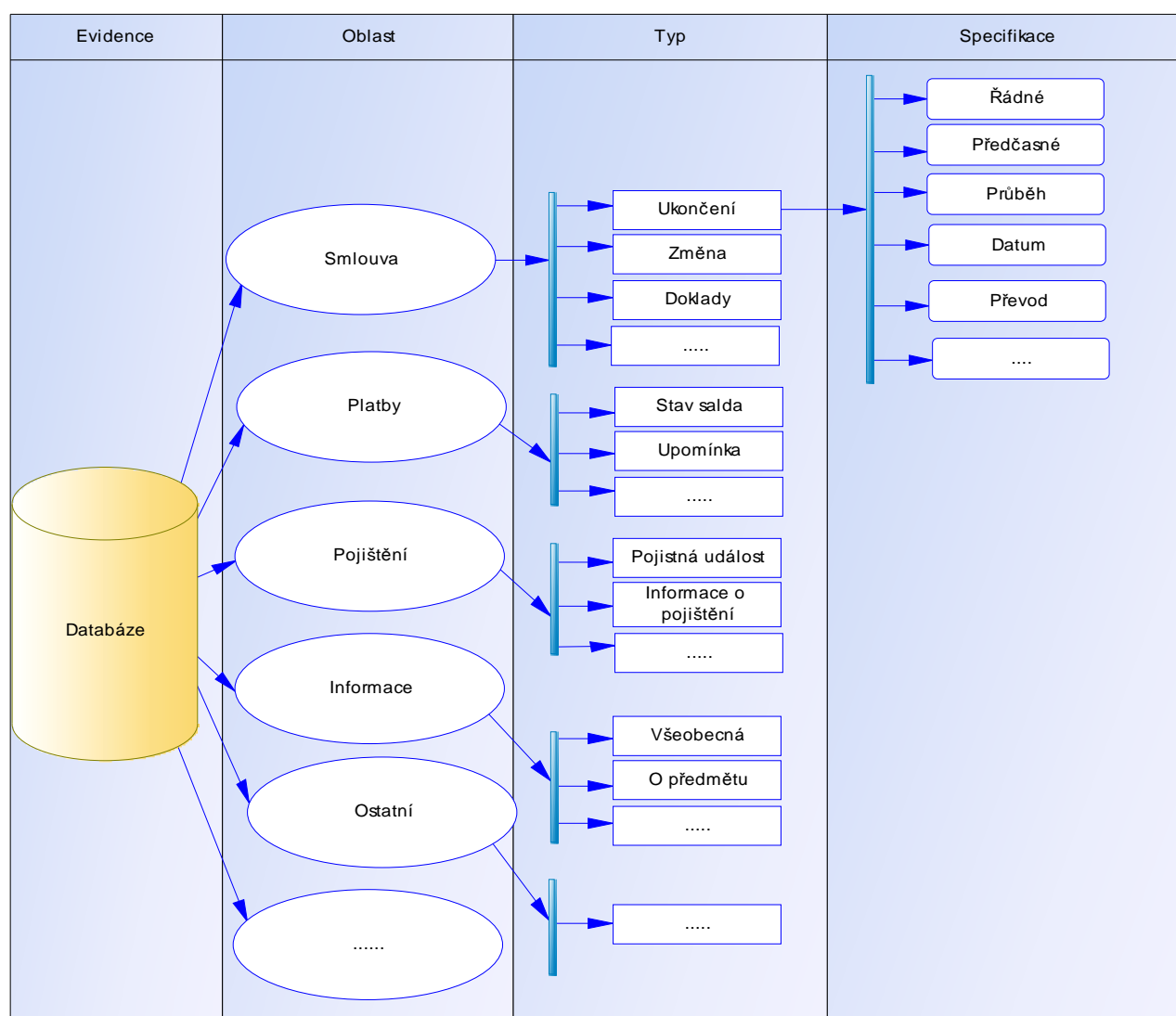
- jméno
- příjmení
- login
- heslo
- blokace (a/n)
- skupinu ve ke které patří
- hovory, které zapsal do databáze

### 6.3 Evidování hovorů

Operátor po skončení hovoru zaznamená do databáze všechny dotazy klienta. Úkolem databáze je zaznamenat dotazy rozdělené do oblastí, typů a specifík ke kterým patří.

Klient, který volá do společnosti, může mít jeden či více problémů, které se mohou týkat různých oblastí leasingu. Tyto oblasti se dále dělí na typy událostí hovoru a specifika hovoru.

Oblasti, typy i specifika může Administrátor libovolně přidávat, odebírat či editovat. Tímto by měla být zajištěna flexibilita databáze. Může být kdykoli přizpůsobena Administrátorem podle současných požadavků, které chceme evidovat.



Obrázek 6.1. Schéma evidovaných částí hovorů v databázi



## 6.4 Oblasti, podtypy a specifika

### Oblasti

Administrátor si vytvoří oblasti do kterých chce hovor dělit. Tyto oblasti mohou být například:

- SMLOUVA
- PLATBY
- POJIŠTĚNÍ
- INFORMACE
- OSTATNÍ
- ...

Pokud se administrátor rozhodne, že oblast už dále nebude používat, tak ji jednoduše zablokuje.

### Podtypy

Ke každé oblasti může administrátor vytvořit množství podtypů, na které se bude daná oblast dělit. U oblasti SMLOUVA mohou být podtypy například:

- UKONČENÍ SMLOUVY
- ZMĚNY
- DOKLADY KE SMLOUVĚ
- OSTATNÍ
- ...

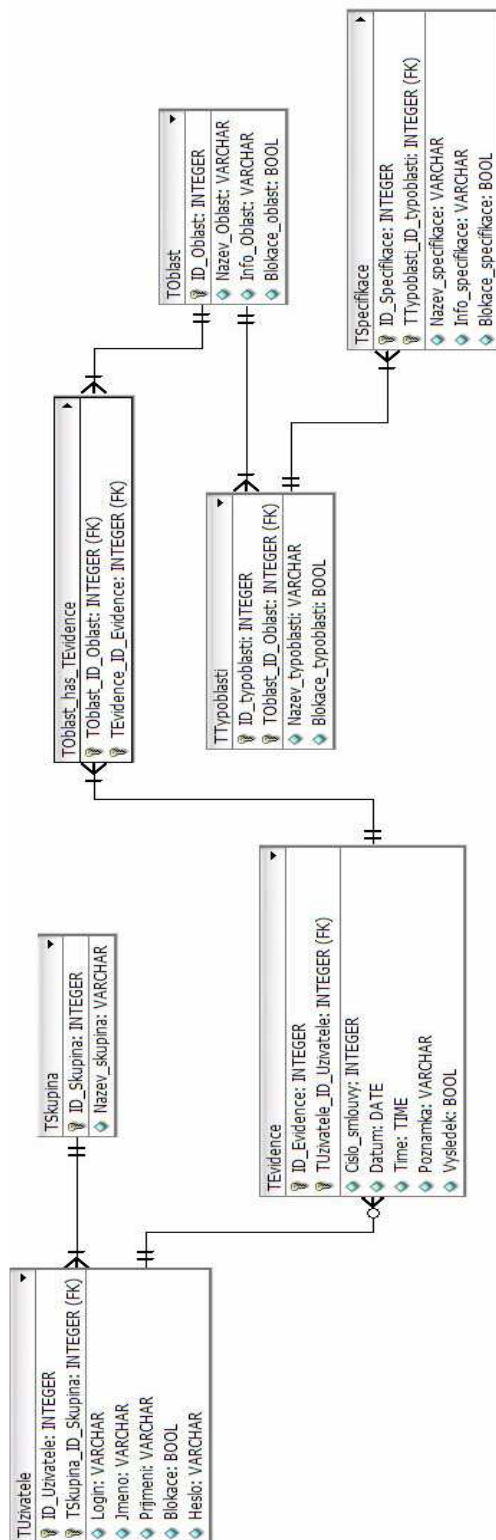
### Specifika

Každý podtyp se dále rozděluje na specifika. Administrátor vytvoří konkrétní specifika hovorů, které budou operátoři vybírat. U podtypu UKONČENÍ SMLOUVY například:

- PŘEDČASNÉ UKONČENÍ
- ŘÁDNÉ UKONČENÍ
- PRŮBĚH UKONČENÍ
- ....

# 7. Tvorba databáze

## 7.1 Datový model



Obrázek 7.1. Schéma datového modelu databáze

Z datového modelu na předchozí straně vidíme, že veškerá data potřebná pro aplikaci jsou uložena v 7 tabulkách, které jsou navzájem propojeny 6 vazbami.

Databáze se skládá se z hlavní tabulky tevidence na kterou jsou navázány další tabulky, které určují konkrétní oblasti, podtypy a specifikace jednotlivých hovorů. Na hlavní tabulku je dále navázána tabulka tuzivatele, která obsahuje všechny uživatele, včetně jejich detailů. Do jaké skupiny uživatel patří zjistíme pomocí tabulky tskupina, která je navázána na tabulku tuzivatele.

Seznam uživatelů, oblastí, podtypy a specifiky hovorů může editovat pouze uživatel, který je ve skupině administrátor. Operátoři pak budou pouze vybírat z předem nadefinovaných možností, které administrátor zadal do databáze. K jednomu hovoru může být zapsáno více detailů.

## 7.2 Popis tabulek

### tskupina

Column Name	Datatype	NOT NULL	AUTO INCR	Flags	Default Value	Comment
ID_Skupina	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
Nazev_skupina	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		

Obrázek 7.2. Schéma atributů tabulky tskupina

Tabulka tskupina eviduje vytvořené skupiny. Každý uživatel musí být přiřazen do jedné skupiny. Tabulka má primární klíč ID\_skupina a ten jednoznačně určuje každou skupinu. Dále obsahuje název skupiny, který je datového typu VARCHAR.

Do databáze jsou vloženy 3 skupiny prostřednictvím příkazů SQL – Administrátor, Manažer, Operátor. Tyto skupiny nemohou uživatelé měnit aplikací Delphi. Mohou se měnit pouze zásahem do databáze. Jsou dané programátorem z důvodů problematického určování nových práv v aplikaci.

tuzivatele

Column Name	Datatype	NOT NULL	AUTO INCR	Flags	Default Value	Comment
Id_tuzivatel	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
Login	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
Jmeno	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
Prijmeni	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
Heslo	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
Blokace	TINYINT(1)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
FK_ID_skupina	INTEGER	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		

Obrázek 7.3. Schéma atributů tabulky tuzivatel

Tabulka tuzivatele obsahuje primární klíč ID\_uzivatele a tento primární klíč je při záznamu v tabulce tevidence doplněn do cizího klíče FK\_Uzivatele. Obsahuje cizí klíč, který zajišťuje vazbu na tabulku tskupina. Další atributy jsou Login, Jmeno, Prijmeni, Heslo, které jsou datového typu Varchar a Blokaci datového typu boolean. Blokace slouží pro blokování uživatele, který už nemá mít přístup do aplikace. Uživatele nelze smazat, protože na něj jsou navázány záznamy v dalších tabulkách. Museli bychom smazat veškeré záznamy v databázi spojené s uživatelem, kterého chceme smazat. Protože je blokace typu boolean, může nabývat hodnoty pouze True/False. Pro vytvoření uživatele je třeba zadat všechny atributy.

tevidence

Column Name	Datatype	NOT NULL	AUTO INCR	Flags	Default Value	Comment
ID_Evidence	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
Cislo_smlouvy	INTEGER			<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
Datum	DATE	<input checked="" type="checkbox"/>				
Cas	TIME	<input checked="" type="checkbox"/>				
Poznamka	VARCHAR(150)			<input type="checkbox"/> BINARY	NULL	
Vysledek	TINYINT(1)			<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
FK_ID_Uzivatel	INTEGER	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		

Obrázek 7.4. Schéma atributů tabulky tevidence

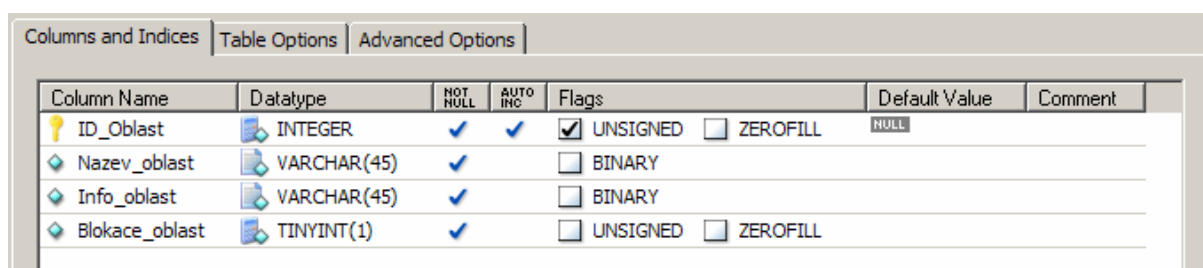
Hlavní tabulka evidence má primární klíč ID\_evidence. Pomocí tohoto klíče budou jednotlivé záznamy od sebe odlišeny i když by měli úplně stejné hodnoty.

Atribut číslo smlouvy vyplní operátor podle čísla smlouvy klienta. Každý klient nemusí mít číslo leasingové smlouvy a nebo ho nemusí vědět, proto tato hodnota může být prázdná(null). Datum a čas je aktuální datum a čas zápisu hovoru, které se vyplní automaticky při zápisu pomocí funkce MySQL.

Výsledek je třeba evidovat pouze spokojen/nespokojen/neutrální. Pokud by nebyl zaznamenán, zůstala by v poli výsledek hodnota Null a výsledek by byl brán jako neutrální. Proto tato hodnota je datového typu BOOLEAN a nemusí být vyplněna. Další atribut je Poznámka. Je to poznámka k celému hovoru a může být vyplněna, pokud ji chce operátor vyplnit. Je omezena na 150 znaků.

Tabulka obsahuje cizí klíč FK\_ID\_Uzivatel. Pomocí tohoto klíče je tabulka propojena vazbou 1:N do tabulky tuzivatele. Další vazba propojuje tabulku tevidence s tabulkou toblast. Protože je tato vazba typu M:N, je realizována prostřednictvím vazební tabulky toblast\_has\_tevidence. Bližší informace o této vazbě jsou v části 7.3. Vazby. Tato vazba je vytvořena z důvodu více specifických patřících k jednomu hovoru.

### toblast

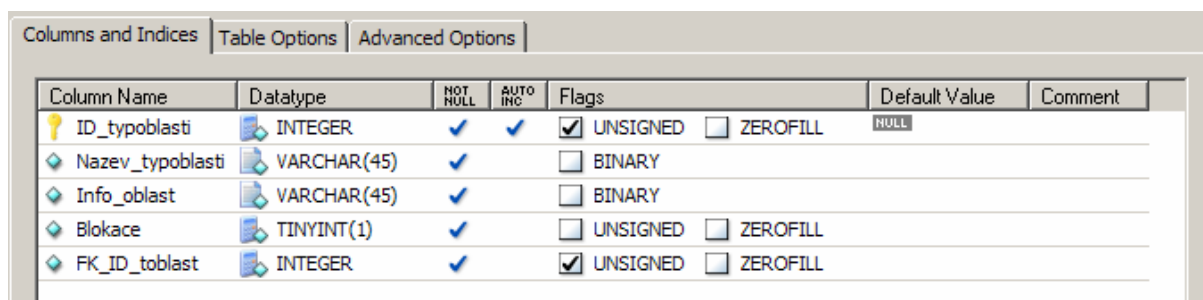


Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
ID_Oblast	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
Nazev_oblast	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
Info_oblast	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
Blokace_oblast	TINYINT(1)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		

Obrázek 7.5. Schéma atributů tabulky toblast

Tabulka toblast slouží pro ukládání oblastí hovoru. Obsahuje kromě primárního klíče ID\_Oblast také název oblasti, informace o oblasti a blokování oblasti. Tabulka obsahuje dvě vazby. První vazba M:N spojuje tuto tabulku s tabulkou tevidence a druhá vazba 1:N zajišťuje spojení s tabulkou TTypoblasti, která obsahuje jednotlivé typy této oblasti. Tabulka nemá žádné cizí klíče. Vazby s touto tabulkou jsou tvořeny pouze primárním klíčem této tabulky.

### ttypoblasti



Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
ID_ttypoblasti	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
Nazev_ttypoblasti	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
Info_oblast	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
Blokace	TINYINT(1)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
FK_ID_toblast	INTEGER	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		

Obrázek 7.6. Schéma atributů tabulky ttypoblasti

Tato tabulka obsahuje seznam typů dané oblasti. Její vlastnosti jsou název oblasti, informace o oblasti a blokování. Obsahuje cizí klíč sloužící k propojení této tabulky a tabulky specifikace, která obsahuje seznam specifikací daného typu u dané oblasti.

### tspecifikace

Column Name	Datatype	NOT NULL	AUTO INCR	Flags	Default Value	Comment
ID_Specifikace	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
Nazev_specifik...	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
Info_specifikace	VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY		
Blokace	TINYINT(1)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		
FK_ID_Typoblasti	INTEGER	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		

Obrázek 7.7. Schéma atributů tabulky tspecifikace

Tabulka specifikace je obdobná jako předchozí. Obsahuje cizí klíč FK\_ID\_Typoblasti, který propojuje tuto tabulku s tabulkou ttypoblasti.

### Datové typy

Ke každému názvu sloupce patří datový typ, který nám říká, zda bude hodnota číslo, text, datum nebo boolean.

Pro číselnou hodnotu je zvolen datový typ INTEGER, který reprezentuje rozsah -2147483648..2147483647. Pokud je zaškrtnut příznak UNSIGNED je rozsah 0.. 2147483647. Tento příznak je zaškrtnut u primárních a cizích klíčů, neboť nemohou nabývat záporných hodnot.

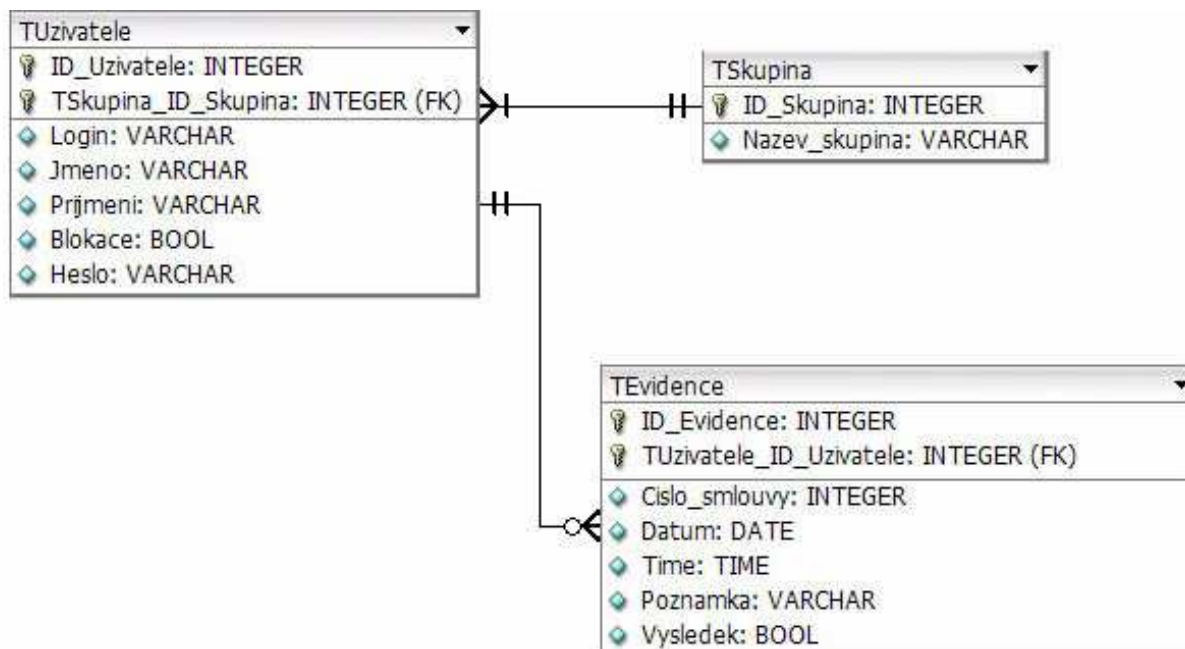
Pro text je zvolen datový typ VARCHAR, který umožňuje uložit text zadané délky. Tato délka je v závorce za datovým typem. V tabulkách musí být délka textových řetězců maximálně 45 znaků. Modifikátor BINARY určuje, že text bude uložen jako binární řetězec.

K uložení data je použit datový typ DATE. Datum je v databázi uloženo ve formátu YYYY-MM-DD. V databázi je datum automaticky vkládáno pomocí funkce CURRENT\_DATE, při vkládání záznamů do tabulky.

Poslední datový typ použitý v databázi je BOOLEAN(TINYINT(1)). Tento datový typ nabývá hodnoty 1 nebo 0. V databázi slouží například pro blokování uživatelských účtů nebo pro blokování dat, které už se nemají zaznamenávat.

### 7.3 Vazby

Následující část datového modelu zobrazuje vazby mezi skupinou, uživatelem a evidovanými hovory.

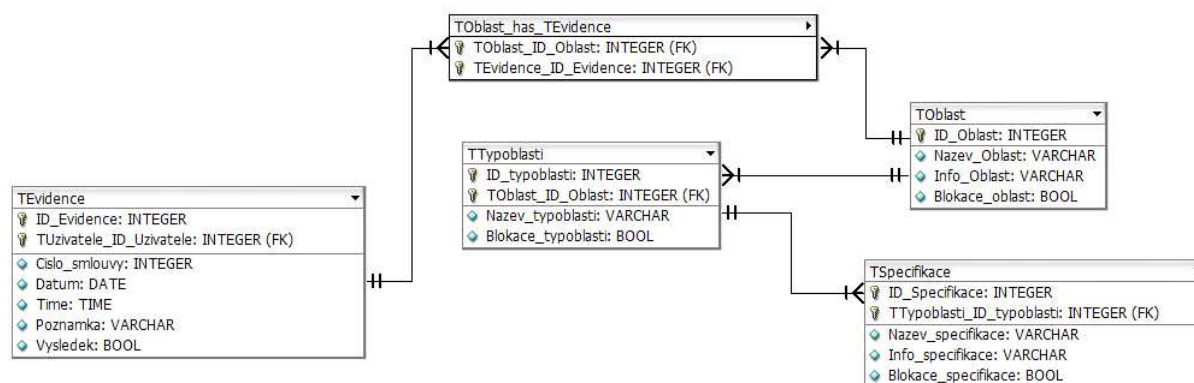


Obrázek 7.8. Schéma části datového modelu

Každý záznam tabulky tevidence obsahuje právě jednoho uživatele, který hovor do evidence zapisoval a ke každému uživateli může patřit 0 nebo více zapsaných hovorů. .

Primární klíč tabulky tuzivatele a cizí klíč FK\_ID\_Uzivatele v tabulce evidence nám určuje vazbu 1:N mezi těmito tabulkami. V praxi tato vazba určuje, že jeden uživatel může zapsat více hovorů a jeden hovor může být zapsán právě jedním uživatelem.

Tabulky tskupina a tuzivatele jsou propojeny cizím klíčem FK\_ID\_skupina tabulky tuzivatele a primárním klíčem ID\_skupina také vazbou 1:N. K jedné skupině může patřit více uživatelů a jeden uživatel musí patřit do právě jedné skupiny.



Obrázek 7.9. Část schématu databáze, v které se evidují oblasti, typy a specifikace hovoru

Tato část datového modelu zobrazuje vazby mezi tabulkami tevidence, toblast, ttypoblast a tspecifikace. Pro realizaci vazby M:N je třeba tabulka toblast\_has\_tevidence.

Z pravé strany tabulky tevidence je vazba M:N do tabulky toblast. Tato vazba je realizovaná pomocí vazební tabulky toblast\_has\_tevidence. Tato tabulka obsahuje primární klíče obou tabulek a každá z tabulek má vazbu typu 1:N s touto tabulkou. Dohromady tvoří vazbu M:N. V praxi to znamená, že k jednomu záznamu v tabulce tevidence patří více oblastí a zároveň jedna oblast může patřit k více záznamům v tabulce tevidence. Tato vazba je realizována z důvodu možnosti zapsat více specifik k jednomu hovoru.

Další vazby jsou vazby typu 1:N. Tabulky toblast, ttypoblast a tspecifikace jsou propojeny touto vazbou. Každá oblast se dělí na více typů oblastí a dále každý typ oblastí se dělí na více specifikací. Vazba je tvořena opět primárním a cizím klíčem.

## 7.4 Práva

Na úrovni databáze je vytvořen pouze administrátorský účet. Aplikace přistupuje k databázi pod tímto účtem a veškerá práva jsou řešena na úrovni aplikace. Aplikace zjistí, kdo se přihlásil, zda zadal správné heslo a do jaké skupiny patří. Na základě této skupiny zobrazí jen funkce přístupné pro danou skupinu. Uživatel se tak nedostane k funkcím, které jsou pro jinou skupinu.





## 8. Aplikace

Uživatel nebude vkládat data přímo do databáze MySQL, ale bude to za něj provádět aplikace. Nejdříve si spustí soubor a aplikace se okamžitě připojí k databázi. Uživatel se přihlásí a pracuje s aplikací. Administrátor bude mít přístupné všechny volby, manažer jen přehledové a operátor jen evidenční.

### 8.1 Komunikace MySQL a Borland Delphi

Aplikace naprogramovaná v Borland Delphi komunikuje s Databází MySQL pomocí DAC for MySQL. Tento program zároveň přidává do Delphi paletu komponent pro práci s MySQL. Pomocí těchto komponent můžeme komunikovat s databází. Z palety nových komponent budeme využívat pouze dvě: TmySQLDatabase a TmySQLQuery.

-  Komponenta TmySQLDatabase poskytuje kontrolu nad databází. Pomocí této komponenty se můžeme k databázi přihlásit přímo z aplikace. Obsahuje ve svých vlastnostech mimo jiných i administrátorské jméno a heslo pro přístup k databázi.
-  Komponenta TmySQLQuery je navázána na komponentu TmySQLDatabase a zároveň na datasource, pomocí které se data mohou zobrazit do tabulky DBGRID. Slouží k zadávání SQL příkazů. Do vlastnosti SQL této komponenty se mohou vkládat SQL příkazy pro ovládání databáze. [26]  
 Například : `Select login, datum, cas from tevidence,tuzivatel where ...` Do vlastností komponenty můžeme přidat SQL příkaz funkcí `add`.

```
Mysqlquery1.SQL.Add
```

```
('select login, datum, cas from tevidence where (login = '"+login+"' and datum=now());')
```

```
Mysqlquery1.ExecSQL;
```

Obrázek 8.1. Ukázka zdrojového kódu komunikace mezi databází a aplikací

Z této ukázky je zřejmé, že pokud chceme použít proměnnou deklarovanou v Delphi v SQL dotazu musíme jí dát mezi apostrofy a plusy.

## 8.2 Přihlášení do aplikace

Obrázek 8.2. Přihlašovací dialog

Přihlášení do aplikace probíhá pomocí samostatného formuláře, který po správném přihlášení zobrazí hlavní formulář.

V tomto formuláři je jeden Combobox, do kterého se načtou při spuštění aplikace všichni uživatelé z databáze, kteří nejsou blokováni. Načtení uživatelů do Comboboxu se provede událostí formuláře „OnCreate“. Uživatel si pouze vybere vlastní login, zadá heslo a klikne na tlačítko Přihlásit. Pokud je heslo správné otevře se hlavní formulář.

Při špatně zadaném hesle se objeví zpráva, že heslo bylo špatně zadáno a musí se zadat znovu. Pomocí Comboboxu je zajištěno snadné přihlašování. Uživatel nemusí vypisovat svůj login a řešit například velká, malá písmena nebo diakritiku. Tlačítko „Přihlásit“ má na starosti zobrazení hlavního formuláře, uložení Login uživatele z Comboboxu do globální proměnné a musí porovnat zadané heslo v editu s heslem v databázi. Heslo z databáze získá prostřednictvím SQL dotazu :

```
SELECT heslo from tuzivatele where nazev_uzivatel=' '+Login+' ';
```

Obrázek 8.3. Ukázka zdrojového kódu vybrání hesla dle zadaného Loginu

Login je globální proměnná, v které je načteno jméno uživatele, který se přihlašuje. Pokud je heslo uživatele v pořádku zobrazí se hlavní formulář. V opačném případě se objeví zpráva o špatně zadaném hesle.

### 8.3 Hlavní formulář

Hlavní formulář se skládá z části hlavičky formuláře a pracovní části formuláře.

#### Hlavička formuláře

Hlavička hlavního formuláře se skládá z loga, aktuálního data, loginu přihlášeného uživatele a tlačítka pro ukončení aplikace. Datum je vloženo prostřednictvím funkce Date. Uživatel je vloženo prostřednictvím globální proměnné, která se uložila při přihlášení.

Obrázek 8.4. Hlavička formuláře

#### Pracovní část formuláře

Druhou část hlavního formuláře tvoří 4 záložky. Každá záložka je zobrazena pouze uživateli, který na ní má práva. Všechny záložky jsou zobrazeny pouze skupině Administrátor. Uživatelé v této skupině mohou hovory zapisovat, vytvářet přehledy, vytvářet a blokovat uživatele a přizpůsobovat databázi. Uživatelé patřící do skupiny Manažer mají přístup pouze k záložce Přehledy a skupina Operátor má přístup pouze k záložce Evidence.

Při zobrazení hlavního formuláře (událost „OnShow“) aplikace zjistí v jaké je uživatel skupině. Aplikace pak zobrazí jen záložky, které mají být dané skupině zobrazeny. Standardně mají všechny záložky vlastnost Tabvisible nastavenou jako True. Záložce, která nemá být zobrazena aplikací nastaví vlastnost Tabvisible jako False.

uzivatel	datum	cas	smlouva	oblast	specifikum
PKleinova	19.4.2007	12:49:04	1502342141	smlouva	Řádné

Obrázek 8.5. Část hlavního formuláře při otevření uživatelem patřícím do skupiny Admin.

## 8.4 Záložky hlavního formuláře

### Evidence

Záložka Evidence slouží uživatelům skupiny Operátor k zaznamenávání hovorů. Uživatel vyplní hodnoty, které chceme evidovat, pokud je evidovaných částí hovoru více, klikne na tlačítko „Další záznam hovoru“ a postupuje obdobně. Nakonec klikne na tlačítko „Zapiš hovor“. Hovor se uloží do databáze s aktuálním datem a časem.

Při zobrazení formuláře se z tabulky TOblast v databázi načte seznam oblastí, které nejsou blokovány do comboboxu. Potom co uživatel vybere jednu z oblastí, načte se z tabulky TPodtyp seznam podtypů patřící k dané oblasti do dalšího comboboxu. Operátor vybere jeden podtyp a tím se zároveň načtou z databáze specifikace hovoru do dalšího comboboxu. Pokud klikne uživatel na další záznam k hovoru, comboboxy se uvedou do defaultního stavu a může vybrat další specifikace hovoru. Nakonec už stačí jen vyplnit poznámku, pokud je třeba, a kliknout na tlačítko „Zapiš hovor“. Pokud je checkbox určený pro poznámku prázdný, není okno pro poznámku zobrazeno. V opačném případě se zobrazí a operátor do něj může poznámku napsat. V okamžiku kliknutí na tlačítko „Zapiš hovor“ se provede SQL příkaz pro vložení informací o hovoru do databáze a zároveň se aktualizuje přehled zapsaných hovorů.

Databáze hovorů

CAC LEASING  
Člen skupiny NVM

Datum: 27.4.2007  
Uživatel: PKleinova

Konec

Evidence | Přehledy | Správa databáze | Správa uživatelů

Číslo smlouvy: 1502341412

S Poznámkou:

Oblast: Smlouva

Podtyp: Ukončení

Specifikace: Předčasné

Řádné  
Předčasné  
Průběh  
Datum  
Převod

Zapiš hovor

Výsledek

Zapsané hovory:

uživatel	datum	cas	smlouva	oblast	specifikum
PKleinova	19.4.2007	12:49:04	1502342141	smlouva	Řádné

Obrázek 8.6. Záložka Evidence

## Přehledy

Záložka přehledy slouží zejména uživateli ve skupině manažer nebo administrátor. Obsahuje filtry pro vybírání hovorů dle zadaných kritérií. Po vybrání konkrétního filtru a kliknutí na tlačítko „Vyhledat“ se provede SQL dotaz a vybraná data se zobrazí v tabulce v dolní části aplikace.

Filtry:

- Oblast (filtrování určité oblasti, vyhledá všechny záznamy týkající se dané oblasti)
- Podtyp (vyhledá všechny záznamy, které mají v evidenci vybraný podtyp )
- Specifikace (vyhledá všechny záznamy, které mají v evidenci vybranou specifikaci)
- Uživatele (vybere hovory zapsané daným operátorem)
- Datum (vybere celý záznam hovoru dle zadaného data)

Pro vyhledávání je možná kombinace filtru a data. Například filtr pro hovory, které obsahují oblast „Smlouva“ a jsou evidované od 1.5.2007 do 1.6.2007. Tlačítko „Export do Excelu“ veškerá data zobrazená v tabulce vloží je do Excelu. Bližší informace o exportování dat jsou v kapitole 8.5. Export do Excelu.

Obrázek 8.7. Záložka Přehledy

## Správa databáze

Záložka správa databáze slouží pouze uživateli, který je ve skupině Administrátor. Může zde přidávat OBLAST, PODTYP A SPECIFIKACI. A zároveň má možnost BLOKACE.

K přidání oblasti do databáze stačí pouze vyplnit oblast a kliknout na tlačítko „Přidej“. Nová oblast se zapíše do databáze, konkrétně do tabulky toblast. Pokud chce uživatel přidat podtyp, nejdříve musí vybrat oblast, pod kterou bude podtyp patřit. Aby se předešlo zbytečným chybám aplikace, edit pro přidání podtypu je přístupný pouze po vybrání oblasti. Je to řešeno vlastností „onchange“ comboboxu. Na tuto událost změny se zpřístupní edit. Přidání specifikace je provedeno obdobně jako podtyp. Nejdříve musí uživatel vybrat oblast, dále se zpřístupní podtyp a po vybrání podtypu může napsat konkrétní specifikaci.

Pro snadnější přehled je možné kdykoli z databáze vybrat seznamy oblastí, podtypů nebo specifikací pomocí tlačítek nad tabulkou. V těchto záznamech je uveden i stav blokace.

Blokace slouží pro blokování skutečností, které už nechceme sledovat. V comboboxech jsou načteny všechny oblasti, podtypy i specifikace. Pokud chce uživatel zablokovat určitou oblast, stačí ji vyhledat v comboboxu, zaškrtnout checkbox blokace a kliknout na tlačítko „Blokovat“. Tím se zablokují i veškeré podtypy a specifikace, které patří do této oblasti. V případě odblokování stačí pouze vyhledat oblast a odškrtnout checkbox blokace.

Obrázek 8.8. Záložka Správa databáze

## Správa uživatelů

Poslední záložka slouží také administrátorovi pro správu uživatelů. V pravé straně záložky je seznam uživatelů s jejich hesly a stavem blokace. V levé straně může uživatel přidat nové uživatele nebo editovat stávající uživatele.

Pokud chceme přidat uživatele musíme vyplnit Jméno, Heslo, Login a přiřadit ho do skupiny. Po kliknutí na tlačítko „Přidat“ se uživatel přidá do databáze a zároveň aktualizuje seznam uživatelů na pravé straně.

Pokud chceme editovat existujícího uživatele, musíme zadat Login a kliknout na tlačítko „Najdi uživatele“. Data nalezeného uživatele se načtou do příslušných editů a uživatel je může editovat. Po změně dat klikne na tlačítko „Editovat“ a změny se promítnou do databáze. Při blokování uživatelů skupiny Administrátor si aplikace zkontroluje, zda existuje ještě jiný uživatel skupiny Administrátor. Pokud neexistuje, tak blokace nebude provedena. Toto opatření má zajistit, aby uživatel nezablokoval omylem sám sebe jako jediného administrátora a pak už by nemohl nikdo databázi aplikačně editovat. Byl by nutný zásah prostřednictvím databázových příkazů SQL.

Databáze hovorů

CAC LEASING  
Člen skupiny NVR

Datum: 25.4.2007  
Uživatel: PKleinova

Konec

Evidence | Přehledy | Správa databáze | Správa uživatelů

Zadej LOGIN:  Najdi uživatele

**Uživatel :**

Jméno :

Příjmení :

Login :

Heslo :

Blokace :

Skupina :

Editovat Přidat

**Seznam uživatelů :**

login	jmeno	prijmeni	heslo	blokace
PKleinova	Petra	Kleinová	p1	0
MKlimova	Marie	Klímová	m1	1
IKotynkova	Iva	Kotýnková	i1	0

Obrázek 8.9. Záložka Správa uživatelů

## 8.5 Export do Excelu

Export je proveden prostřednictvím technologie OLE. Aplikace Delphi nejdříve otevře program Microsoft Excel a vytvoří nový list. Pomocí dvou FOR cyklů se načte hodnota z tabulky DBGRID a umístí se do buňky v Excelu. Souřadnice umístění hodnoty udávají hodnoty cells[x,y] ve FOR cyklu. Cyklus skončí po načtení a přenesení všech záznamů do Excelu.

```
procedure TForm.Button6Click(Sender: TObject);  
  
var excel:variant;  
    i,j:integer;  
    prenos:string;  
begin  
    excel:=CreateOleObject('Excel.application');  
    excel.visible:=true;  
    excel.caption:='MS Excel';  
    if VarIsEmpty(excel) then showmessage('Nelze připojit k excelu ');  
    excel.workbooks.add();  
  
for j:=0 to dbgrid1.DataSource.DataSet.RecordCount-1 do  
begin  
    for i:=0 to dbgrid1.DataSource.DataSet.FieldCount-1 do  
begin  
    prenos:=dbgrid1.DataSource.DataSet.Fields[i].AsString;  
    excel.workbooks[1].worksheets[1].cells[j+1,i+1]:=prenos;  
end;  
    dbgrid1.DataSource.DataSet.Next;  
end;
```

Obrázek 8.10. Ukázka zdrojového kódu otevření Excelu a vložení dat do nového listu



## 9. Náklady na databázovou aplikaci

Náklady na databázovou aplikaci jsou minimální díky použitým technologiím, které jsou téměř zdarma.

Databáze MySQL je zadarmo a tudíž nejsou žádné náklady na databázi.. Program pro komunikaci MySQL a Delphi (DAC for MySQL) platí stejně jako v případě Delphi pouze programátor, který ho používá. Za všechny programy které v tomto vývojovém prostředí programátor vytvoří a používají DAC for MySQL se nemusí odvádět licenční poplatky společnosti Microolap. Jediný náklad, který je na tuto databázovou aplikaci je cena, za kterou koupí společnost aplikaci od programátora. Žádné další licence ani poplatky nebudou.

Požadavky na místo na harddisku jsou zanedbatelné. Celá aplikace nebude mít větší velikost než 2MB. Přenosové cesty budou vytíženy pouze při zapisování do databáze a nebo vybírání dat z databáze. Na rozdíl od původní aplikace jsou přenosové cesty využity pouze při konečném zapsání hovoru do databáze jedním SQL příkazem a ne v celém průběhu zapisování hovoru.

## 10. Srovnání navrhnuté aplikace a původní aplikace

Navrhnutá databázová aplikace řeší řadu nedostatků původní aplikace a splňuje většinu požadavků na novou aplikaci. Je rychlá, flexibilní a pro uživatele snadno pochopitelná.

### 10.1 Práce s aplikací

Ve srovnání s původní aplikací je tato aplikace mnohem rychlejší. K zapsání hovorů nemusí operátor čekat, než se aktualizují stránky Exploreru, počká jen při zapsání hovoru do databáze, přičemž doba by měla být v řádu jednotek vteřin. Veškeré informace spojené s hovorem vybere v aplikaci a pak aplikace pouze odešle SQL příkaz k zapsání dat do databáze.

Jakákoli komunikace mezi aplikací a databází probíhá pomocí SQL příkazů. Tyto SQL příkazy by měli být v databázi promítnuty okamžitě nebo během několika málo vteřin.

## 10.2 Splněné požadavky na novou aplikaci

- Aplikace nebude uživatele odhlašovat při delší nečinnosti.
- Zapsání hovoru je jednoduché a rychlé.
- Aplikace zaznamenává pouze položky, které si administrátor sám zvolí.
- Poznámka nemusí být vyplněna.
- Uživatelé mají okamžitý přehled o zapsaných hovorech aktuálního dne.
- Administrátor má přístup k přizpůsobování databáze.
- Aplikace by neměla hlásit chyby.
- Načtení přehledu hovorů by mělo být provedeno během několika málo vteřin.
- Jakákoli vybraná data lze exportovat do Excelu.
- Náklady jsou minimální.

## 10.3 Přínosy navrhnuté aplikace

Největší přínos této aplikace je pro Operátora, který jí využívá celý den. Doba zapsání hovoru bude trvat v průměru 15 vteřin na rozdíl od původní aplikace, kde se doba pohybovala v průměru kolem 40 vteřin. Operátor tuto činnost provádí přibližně 100krát denně, podle počtu hovorů. Ušetří tak čas, který bude možno prospěšněji využít při odbavování hovorů přicházejících v rychlém sledu po sobě. Předmětem převážné části hovorů (cca 90%) jsou opakované záležitosti a zejména u těchto hovorů není třeba vyplňovat poznámku. Operátor tedy ušetří čas i nevyplněním poznámky.

Přínos aplikace pro Administrátora je ve snadném přizpůsobování databáze, dle požadavků společnosti. Rok od roku je situace jiná a administrátor může blokovat položky, které není třeba evidovat a vytvořit nové.

Přínos pro Manažera je v přehledných filtrech, jejichž prostřednictvím může snadno vyčlenit jakýkoli hovor a ten následně exportovat do Excelu.

## 10.4 Srovnání stávající a navrhnuté aplikace

Stávající databázová aplikace	Navrhnutá databázová aplikace	Vyřešeno
Aplikace uživatele ohlásí při 10 minutové nečinnosti.	Uživatel se přihlásí a zůstane přihlášen až do ukončení aplikace.	✓
Pro úspěšné zapsání hovoru musí uživatel projít skrz scénáře.	Uživateli stačí zapsat pouze Oblast, Typoblasti a Specifika.	✓
Aplikace eviduje velké množství detailů, které není třeba sledovat.	Aplikace eviduje pouze Administrátorem zvolená data.	✓
Výskyt chyb.	V aplikaci by se neměly vyskytovat chyby.	✓
Velké prodlevy při načítání jednotlivých webových stránek.	Vše je uspořádáno na jedné záložce. Uživatel čeká pouze při zapsání do databáze.	✓
Poznámka musí být vyplněna vždy a musí se zkopírovat do spodního okna.	Poznámka je vyplněna pouze pokud si to uživatel přeje a je vyplněna jen jednou.	✓
Do aplikace se nemohou přidávat nové detaily hovoru a staré nejdou odstranit.	Aplikaci si přizpůsobí administrátor podle momentálních potřeb.	✓
Načítání přehledů trvá velmi dlouhou dobu	Načítání přehledů by mělo trvat pouze několik málo vteřin.	✓
Aplikace nemá možnost rychlého zápisu hovorů, které se opakují mnohokrát denně.	Aplikace nemá možnost rychlého zápisu hovorů, které se opakují mnohokrát denně.	✗

Tabulka 1.1. Srovnání stávající a navrhnuté aplikace

## 11. Závěr

Cílem této bakalářské práce bylo popsat stávající aplikaci používanou v Call Centru společnosti CAC Leasing, vymežit nedostatky této aplikace a navrhnout aplikaci, která by tyto nedostatky řešila.

Analyzováním aplikace používané v Call Centru jsem popsal její funkce, nedostatky a práci s aplikací. Na základě této analýzy je navrhována nová databázová aplikace, která je tvořena databází MySQL a aplikací Delphi pro přístup k databázi. Tato nová databázová aplikace byla navržena s ohledem na maximální produktivitu práce uživatelů a minimální náklady. Při srovnání původní a navržené aplikace je nová aplikace rychlejší, flexibilní a nevyžaduje zvýšené finanční náklady.

Navrhnutá databázová aplikace řeší téměř všechny nedostatky původní aplikace. Pokud by byla implementována do systému společnosti, výrazně by se zvýšila efektivita pracovníků Call Centra.

## 12. Seznam obrázků a schémat

Obrázek 2.1. Schéma procesu zákaznické péče a podpory .....	10
Obrázek 2.2. Proces řízení hovoru s klientem.....	11
Obrázek 3.1. Use Case diagram SW používaného v Call Centru .....	14
Obrázek 3.2. Postup zapisování hovoru v aplikaci .....	16
Obrázek 6.1. Schéma evidovaných částí hovoru v databázi .....	23
Obrázek 7.1. Schéma datového modelu databáze .....	25
Obrázek 7.2. Schéma atributů tabulky tskupina.....	26
Obrázek 7.3. Schéma atributů tabulky tuzivatel .....	27
Obrázek 7.4. Schéma atributů tabulky tevidence .....	27
Obrázek 7.5. Schéma atributů tabulky toblast.....	28
Obrázek 7.6. Schéma atributů tabulky ttypoblasti .....	28
Obrázek 7.7. Schéma atributů tabulky tspecifikace .....	29
Obrázek 7.8. Schéma části datového modelu.....	30
Obrázek 7.9. Část schématu databáze, v které se evidují oblasti, typy a specifikace hovoru..	31
Obrázek 8.1. Ukázka zdrojového kódu komunikace mezi databází a aplikací .....	32
Obrázek 8.2. Přihlašovací dialog.....	33
Obrázek 8.3. Ukázka zdrojového kódu vybrání hesla dle zadaného Loginu .....	33
Obrázek 8.4. Hlavička formuláře .....	34
Obrázek 8.5. Část hlavního formuláře při otevření uživatelem patřícím do skupiny Admin. .	34
Obrázek 8.6. Záložka Evidence.....	35
Obrázek 8.7. Záložka Přehledy .....	36
Obrázek 8.8. Záložka Správa databáze .....	37
Obrázek 8.9. Záložka Správa uživatelů.....	38
Obrázek 8.10. Ukázka zdrojového kódu otevření Excelu a vložení dat do nového listu.....	39
Tabulka 1.1. Srovnání stávající a navržené aplikace .....	42

## 13. Zdroje

### 13.1 Knížní dokumenty

1. BUCHALCEVOVÁ, Alena. Vývoj aplikací v Delphi. Praha: Vysoká škola ekonomická v Praze, 2001, 209 s.  
ISBN: 80-245-0150-3
2. LACKO, Luboslav. SQL – Hotová řešení. Brno: Computer Press, 2003, 296 s.  
ISBN: 80-7226-975-5
3. PÍSEK, Slavoj. Začínáme programovat v Delphi : Podrobný průvodce začínajícího uživatele. Praha: Grada Publishing, 2000, 303 s.  
ISBN: 80-247-9008-4
4. WELLING, Luke, THOMSON, Laura. MySQL : Průvodce základy databázového systému. Brno: CP Books, 2005, 255 s.  
ISBN: 80-251-0671-3
5. HERNANDEZ, J. Michael. Návrh databází. Praha: Grada Publishing, 2006, 408 s.  
ISBN: 80-247-0900-7
6. HOULETTE, Forrest. SQL – Příručka programátora. Praha: Softpress, 2001, 382 s.  
ISBN: 80-86497-14-3
7. DuBois,Paul. MySQL profesionálně. programátora. Brno: Mobil Media, 2003, 1071 s.  
ISBN: 80-86593-41-x

### 13.2 Elektronické dokumenty

8. DOBROVOLNÝ, Aleš: Jak na databáze 15.10.2005 [web site] Dostupné z [http://adobrovolny.net/uvod/124/jak\\_na\\_databaze\\_I](http://adobrovolny.net/uvod/124/jak_na_databaze_I)  
Článek o databázích.
9. DOBROVOLNÝ, Aleš: Normalizace databází 24.9.2006 [web site] Dostupné z [http://adobrovolny.net/uvod/204/normalizace\\_databazi](http://adobrovolny.net/uvod/204/normalizace_databazi)  
Článek o normalizaci databází.
10. JAKEL, Milan : Uživatelé a přístupová práva v MySQL 19.4.2002 [web site].  
Dostupné z <http://interval.cz/clanky/nebojte-se-formatu-datumu-a-casu-v-mysql/> .  
Článek o formátu data a času v MySQL.
11. JAKEL, Milan : Uživatelé a přístupová práva v MySQL 11.12.2001 [web site].  
Dostupné z <http://interval.cz/clanky/instalujeme-mysql-na-windows/> .  
Článek o instalaci MySQL ve Windows.
12. JAKEL, Milan : Uživatelé a přístupová práva v MySQL 21.12.2001 [web site].  
Dostupné z <http://interval.cz/clanky/uzivatele-a-pristupova-prava-v-mysql/> .  
Článek o přístupových právech v MySQL.

13. KADLEC Václav. Článek Umíme to s Delphi: 17.díl – databáze v delphi 16.7.2001 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=100888>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.
14. KADLEC, Václav. Článek Umíme to s Delphi: 18.díl – databáze v delphi, pokračování 23.7.2001 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=100961>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.
15. KADLEC, Václav. Článek Umíme to s Delphi: 34.díl – práce se soubory podrobněji, 1.část 3.9.2001 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=101674>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.
16. KADLEC, Václav. Článek Umíme to s Delphi: 37.díl – datum a čas je nade vše: výber data a času 3.12.2001 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=103200>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.
17. KADLEC, Václav. Článek Umíme to s Delphi: 38.díl – technologie OLE jasně a srozumitelně , 1.část 10.12.2001 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=103309>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.
18. KADLEC, Václav. Článek Umíme to s Delphi: 39.díl – technologie OLE jasně a srozumitelně , 2.část 17.12.2001 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=103365>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.
19. KADLEC, Václav. Článek Umíme to s Delphi: 40.díl – technologie OLE jasně a srozumitelně , dokončení 7.1.2002 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=103483>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.
20. KADLEC, Václav. Článek Umíme to s Delphi: 49.díl – pohrajte si s formulářem 11.3.2002 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=104610>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.
21. KADLEC, Václav. Článek Umíme to s Delphi: 75.díl – databáze v Delphi, obecný úvod 17.2.2003 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=109410>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.

22. KADLEC, Václav. Článek Umíme to s Delphi: 76.díl – databáze v Delphi, navrhujeme databázi 24.2.2003 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=109532>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.
23. KADLEC, Václav. Článek Umíme to s Delphi: 77.díl – databáze v Delphi, jaké máme možnosti 3.3.2003 [web site].  
Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=109629>.  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) zabývající se programováním v delphi.
24. MAKOWSKI, Lukáš : MySQL 16 - Tvorba databázového systému 6.9.2002 [web site]. Dostupné z <http://www.zive.cz/h/Programovani/AR.asp?ARI=107361> .  
Seriál, který vychází na [www.zive.cz](http://www.zive.cz) o SQL/MySQL.
25. TRUNDA, Michal : Jak nakonfigurovat ODBC pro MySQL 30.3.2004 [web site].  
Dostupné z <http://www.dbsvet.cz/view.php?cislocclanku=2004033001> .  
Článek o konfiguraci ODBC.
26. Internetová encyklopedie. [web site]. Dostupné z <http://cs.wikipedia.org/wiki/Delphi>  
Článek o Borland Delphi.
27. Studna – internetový obchod [web site].  
Dostupné z <http://www.studna.cz/4276/vyvoj-software/delphi/dac-for-mysql>  
Informace o softwaru
28. Homepage firmy Microolap. Produkt DAC for MySQL™ [web site]. Dostupné z <http://www.microolap.com/products/connectivity/mysqldac/> .  
Článek o konfiguraci ODBC.
29. Výroční zprávy CAC LEASING.  
Dostupné z <http://www.cac.cz/cz/spolecnost/vyrocní-zpravy.html>