Univerzita Karlova v Praze
1. lékařská fakulta

BAKALÁŘSKÁ PRÁCE

Zdeněk Telička

Databázová aplikace pro výzkum nemocí štítné žlázy

3. interní klinika 1. lékařské fakulty Univerzity Karlovy
a Všeobecné fakultní nemocnice v Praze

Vedoucí bakalářské práce: MUDr. Ing. Daniel Smutek, PhD.

Praha, červen 2006

Čestné prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a s použitím pramenů uvedených v seznamu literatury.

V Praze dne 8. června 2006 Zdeněk Telička

Poděkování

Chtěl bych poděkovat svému zaměstnavateli, společnosti Actinet Informační systémy, za výborné pracovní podmínky, které mi umožňují studovat a pracovat na dalších zajímavých projektech.

Abstrakt

Tato bakalářská práce popisuje vývoj databázové aplikace pro 3. interní kliniku 1. lékařské fakulty Univerzity Karlovy a Všeobecné fakultní nemocnice v Praze. Databáze je zaměřena na ukládání podrobných údajů o vyšetřeních a léčbě pacientů s nemocemi štítné žlázy. Smyslem aplikace je usnadnit zpracování dat pro klinické výzkumy, které je velmi obtížné, jelikož informace ukládané do nemocničního informačního systému ve formě lékařských zpráv neobsahují podrobné údaje o provedených vyšetřeních. A pokud je obsahují, vyhledávání jednotlivých údajů je náročné na čas. Aplikace rovněž umožňuje import výsledků biochemických vyšetření z nemocničního systému, což poskytuje lékařům široké možnosti ve zpracovávání a porovnávání těchto výsledků ve vztahu k léčbě pacienta. Příjemnou vlastností této aplikace je urychlení psaní lékařských zpráv do nemocničního informačního sytému, protože aplikace umí tyto zprávy automaticky generovat z dat uložených v databázi.

Abstract

The Bachelor thesis “Database Application for Research on Thyroid Gland Diseases” deals with development of database application for 3rd Medical Department, 1st Faculty of Medicine, Charles University in Prague. Application's aim is to store detailed data of patients’ examinations and treatments with thyroid gland diseases. Purpose of the application is to facilitate data processing for clinical research, which is very difficult, because information stored in hospital information system doesn't contain detailed records from medical examinations. Also, searching for several entries is slow. As well, application allows import biochemical results from hospital information system. This function provides physicians wide range of possibilities in data processing and comparison in relation to treatments. Effective feature of this application is faster writing medical reports stored in information system because they are automatically generated from application's database.

Obsah

1. Úvod

Tato bakalářská práce vznikla jako součást mé práce na výzkumných projektech na 3. interní klinice 1. lékařské fakulty Univerzity Karlovy a Všeobecné fakultní nemocnice v Praze. Na klinice jsem začal spolupracovat s MUDr. Ing. Danielem Smutkem, PhD. původně na projektu počítačové analýzy digitálních ultrazvukových snímků štítné žlázy. Ultrazvukové vyšetření štítné žlázy je dnes nejčastější volbou pro diagnostiku a screening zánětů štítné žlázy [1; 2; 3]. Jedna z nejčastějších tyreopatií je Hashimotova tyreoiditida, chronický zánět štítné žlázy [4]. Zánět ve štítné žláze způsobuje změny struktury tkáně, které jsou difúzní a nepostihují pouze samotnou žlázu, a proto mohou být detekovány při ultrazvukového zobrazení [1; 3]. Stanovení nálezů z ultrazvukových snímků při vyšetřování štítné žlázy je velmi obtížné [5], protože je založeno na zkušenostech lékaře. Pro stanovení diagnózy využívají lékaři kvalitativního hodnocení velikosti vyšetřované žlázy, její perfúze, struktury a echogenity parenchymu bez možnosti poskytnutí kvantifikovatelných ukazatelů. Zmíněný projekt se zabývá automatickou analýzou textury digitálních sonografických snímků štítné žlázy zobrazených v B-módu. Tyto snímky jsou zpracovávány algoritmy naprogramovanými v aplikaci Matlab, které analyzují kombinace příznaků založených na vlastnostech textury sonogramu a klasifikují sonogram do dvou tříd podle toho, rozpozná-li algoritmus sonogram zdravé štítné žlázy nebo žlázy postižené chronickým autoimunitním zánětem. Správnost klasifikace pak byla ověřována aspirační cytologií. Aplikací uvedených postupů bylo ověřeno, že počítačovou analýzou ultrasonografického obrazu lze odlišit chronický autoimunitní zánět od zdravé tkáně se 100% úspěšností [6].

Při práci na popisovaném projektu vznikla potřeba naprogramovat databázovou aplikaci, jež by poskytovala prostředky k zaznamenávání podrobných dat o onemocněních pacientů, jejich vyšetřeních a léčbě. Například při sonografickém vyšetřování štítné žlázy by bylo možné ukládat veškeré zjišťované parametry, což by zjednodušilo následné porovnávání změny parametrů štítné žlázy v závislosti na léčbě, apod. Vývoj této databázové aplikace se stal mým úkolem a jeho popisu se věnuji v této

bakalářské práci. Využití aplikace bylo během jejího vývoje rozšířeno i na jiné studie, které se zabývají nemocemi štítné žlázy.

Úkolem této aplikace je shromažďovat od lékařů požadované informace o diagnostice nemocí a léčbě pacientů tak, aby bylo snadné následně zpracovávat uložené údaje pro výzkumné účely. Lékaři do této aplikace zadávají informace již během vyšetřování pacientů a poté uložené informace exportují ve formě zpráv do nemocničního informačního systému. Další funkcí vyvinuté aplikace je import výsledků biochemických vyšetření z NIS. Lékaři si mohou do databáze ukládat volitelně takové výsledky, které potřebují zpracovávat ve svých klinických studiích. Uložené výsledky jsou v aplikaci kdykoliv dostupné k automatizovanému zpracování například pro statistické účely. Lékařům to ušetří čas a práci, jež by jinak musela být vynaložena k ručnímu přepisování údajů z NIS do ručně připravených tabulek.

Popisovaná databázová aplikace řeší nedostatky nemocničního informačního systému obsahující informace o diagnostice nemocí a léčbě pacientů ve formě textových zpráv, z kterých se zpětně získávají konkrétní údaje jen velmi zdlouhavě a složitě. Navíc, do informačního systému lékaři zapisují pouze určité informace, a tak dochází ke ztrátám mnoha údajů, které by lékaři chtěli sledovat ve svých studiích.

2. Analýza databázové aplikace

2.1 Současný způsob zpracování dat pro klinické studie

Na 3. interní klinice Všeobecné fakultní nemocnice v Praze se používá k ukládání dat o pacientech nemocniční informační systém Medea společnosti Stapro. Do tohoto systému lékaři zapisují zprávy o diagnostice nemocí, vyšetřeních a léčbě pacienta. Dále se do něj z centrální laboratoře ukládají zpracované výsledky biochemických vyšetření, které mají lékaři přístupné. Mnoho lékařů potřebuje nejrůznější data o pacientech zpracovávat do svých výzkumů a studií, avšak zde nastává problém se získáním požadovaných dat, jelikož do NIS se ukládají jen důležité informace nutné pro sledování postupu léčby pacienta. Proto zde lékaři nenaleznou

detailní informace o provedených vyšetřeních. Navíc, každý lékař do své studie zaznamenává data, která mohou být pro ostatní lékaře nepodstatná a do NIS je nepíší.

Pokud lékař začne vyhledávat některé údaje v nemocničním informačním systému, čeká jej časově náročná práce při prohledávání jednotlivých údajů, které jsou skryté ve zprávách z vyšetření pacienta. Rovněž vyhledání požadovaných výsledků biochemických vyšetření vyžaduje velkou časovou náročnost. Další částí práce lékařů je ruční opisování údajů do připravených tabulek. Situaci mnozí lékaři řeší tak, že si vytvářejí tabulky, do kterých si zapisují potřebné údaje již během vyšetření pacienta. Mají tak možnost sami si určit sledované údaje a navrhnout tabulku dle svých představ. Toto řešení však v sobě skrývá mnoho následujících nevýhod:

a) Vyšší nároky na čas nutný pro vyšetření pacienta, neboť lékař zapisuje všechny údaje pacienta dvakrát: nejdříve do NIS a poté do své tabulky.

b) Pro každou novou studii vzniká nová tabulka, která má svůj vlastní seznam pacientů. Není tedy možnost sdílet společnou databázi pacientů mezi několika studiemi.

c) Složitá spolupráce mezi lékaři na výzkumných pracích. Pro jednotlivé studie je často nutné získat informace z vyšetření od velkého počtu pacientů. Lékaři se tak musí dohodnout ke spolupráci individuálně a seznámit se s tím, co je třeba do tabulek zapisovat.

Dle výše popsaných zkušeností se začalo uvažovat o vývoji databázové aplikace, která by situaci se zadáváním a sledováním údajů o vyšetřeních pacientů řešila takovým způsobem, aby byl pro všechny lékaře snadný a časově nenáročný.

2.2 Požadavky na databázovou aplikaci

Při vývoji aplikace byl kladen největší důraz na snadnost jejího ovládání a rychlost vkládání dat do databáze. Práce s aplikací se nesměla stát další zdržující činností, neboť aplikace by se pak brzy přestala používat a bránilo by jí to v dalším rozšíření na ostatní pracoviště kliniky. Snažil jsem se při vývoji uživatelského rozhraní

postupovat tak, aby zadávání jednotlivých údajů do databáze bylo ještě rychlejší, než zdlouhavé psaní zpráv do nemocničního informačního systému. Aplikaci jsem se společně s lékaři snažil navrhnout do takové podoby, která zaručuje intuitivní ovládání bez větších nároků na její studování. Aby aplikace neobtěžovala uživatele zbytečnými chybovými dialogy a nenutila je dělat úkony, které vůbec nepožadují, mnoho údajů není povinné do databáze zadávat.

Aplikace musela obsahovat jednotnou databázi pacientů společnou pro všechna vyšetření, při kterých se bude používat. S tím souvisí i další požadavek na možnost rychlého vyhledávání pacientů, kteří již nějaký záznam v databázi mají uložený. Podařilo se mi aplikaci naprogramovat tak, že lékaři se nezatěžují prohledáváním databáze, zda-li se v ní již hledaný pacient nenachází, ale po zadání rodného čísla pacienta mají možnost ihned zobrazit jeho identifikační údaje, pokud bylo zadané rodné číslo nalezeno.

Velmi významnou roli při návrhu databáze sehrálo stanovení kompromisu mezi množstvím údajů, které se do databáze ukládají a časovou náročností pro tyto úkony. Lékaři požadují při vyšetření pacientů ukládání co největšího počtu parametrů, ale nesmí to překročit jistou mez, která by zbytečně zvýšila nároky na jejich čas. Mohlo by pak dojít ke snížení přehlednosti aplikace a zvýšení náročnosti na ovládání. Nesměla se rovněž stát situace, aby si některý z lékařů pracujících s databází musel dodatečně vytvářet vlastní poznámky pro sledování parametrů, které v databázi nemůže najít.

Aby byla práce s aplikací komfortní, zvolil jsem pro zadávání údajů do databáze takové ovládací prvky, které umožňují volbu údaje z předem definovaných hodnot. Tímto jsem dosáhl významného zrychlení práce s databází, neboť lékaři jen vyberou jednu z předdefinovaných hodnot a ručně vepisují údaje pouze v případech, pokud chtějí zprávu ukládanou do nemocničního systému doplnit o údaje, které se do databáze neukládají. K uživatelskému rozhraní jsem naprogramoval i další doplňkové nástroje, které lékařům dovolují rozšiřovat databázi aplikace o předem definované hodnoty pro jejich opakované použití. Samozřejmě, každou hodnotu lze do databáze vepsat ručně a uživatelé nejsou nuceni vybírat jen z uvedených předvoleb.

Důležitým kritériem při návrhu aplikace byla její modulárnost. Aplikace je dělena na jednotlivé moduly, které odpovídají pracovištím kliniky nebo různým typům vyšetření pacienta. Podrobnému popisu funkce každého modulu se věnuji v 5. kapitole. Vývoj databázové aplikace je v současné době soustředěn na pracoviště zabývající se nemocemi štítné žlázy. V databázi proto musí být snadná možnost pro doplnění dalších sledovaných parametrů v jednotlivých modulech, pokud se k jejímu používání připojí lékaři s doplňujícími požadavky nebo z jiných pracovišť kliniky. Požadavky na databázi shrnuji v tabulce č.1.

Přehled základních požadavků na databázovou aplikaci	
1.	Jednotná databáze pacientů využitelná pro všechna vyšetření.
2.	Dostatečné množství sledovaných parametrů.
3.	Jednoduchost a přehlednost grafického rozhraní.
4.	Nezvýšení časové náročnosti na vyšetření pacienta.
5.	Modulárnost a přizpůsobitelnost dalším požadavkům lékařů.

Tabulka č.1 Přehled požadavků na databázi.

2.3 Volba vývojového nástroje

K vývoji databázové aplikace byl zvolen Microsoft® Access 2003, jelikož je součásti verze Microsoft® Office, která je dostupná na všech pracovištích kliniky, ale také v celé fakultní nemocnici. Tímto byly eliminovány dodatečné finanční náklady na pořízení jiného vývojového nástroje, například Borland® Delphi® nebo Microsoft® Visual Studio®. Access má v sobě implementovány veškeré prostředky potřebné pro vývoj databáze a jejího uživatelského rozhraní. Vývoj aplikace v Accessu je také mnohem rychlejší, protože mnoho funkcí za programátora obsluhuje přímo rozhraní Accessu a programátor se tak může soustředit pouze na tvorbu databáze. Microsoft se snaží do Accessu alespoň částečně implementovat podporu pro XML. Uživatel má možnost importovat a exportovat tabulky do souboru XML a tím zvýšit použitelnost této aplikace ve spolupráci s jinými aplikacemi [7]. Access má v sobě implementovány následující základní funkce:

- Návrh databázových tabulek a relací mezi nimi.
- Návrh SQL dotazů pro různé uživatelské pohledy na uložená data nebo pro manipulaci s daty.
- Tvorbu grafického uživatelského prostředí (GUI) pomocí formulářů umožňující snadnou uživatelskou práci v databázové aplikací.
- Programovací jazyk Visual Basic® for Applications dovolující vývojáři naprogramování automatizovaných funkcí nebo interakcí mezi uživatelem a databázovou aplikací a nebo mezi databázovou aplikací a operačním systémem.
- Tvorba sestav pro tiskové výstupy.
- Podpůrné nástroje pro rozdělení databáze na dva soubory. První, který je zdrojem dat a druhý, který má v sobě uloženo grafické prostředí (formuláře) a makra (kód ve Visual Basicu).
- Zálohování, případně replikování databáze pro možnost automatického udržování několika kopií databáze.
- Převod databáze na SQL server. Databázi lze vytvořit nebo převést na Microsoft® SQL Server nebo se lze připojit k existující databázi kteréhokoliv databázového stroje, jež má ovladač pro ODBC. V tomto případě bude soubor databáze sloužit pouze jako uživatelské rozraní. ODBC je standardní aplikační programátorské rozhraní poskytující jednotný přístup k různým databázovým systémům. Access pro svou databázi také zprostředkovává ovladač ODBC a tím je možné zvolit opačný postup: připojit se například z aplikace naprogramované v Delphi k databázi vytvořené v Accessu [8].
- Možnost vytvoření tzv. MDE souboru, který je určen pouze pro uživatelskou práci s databází. V tomto souboru není možné měnit struktury databáze nebo formulářů a obsahuje pouze zkompilovaný kód Visual Basicu, formuláře a sestavy. Databázová aplikace ke svému spuštění bude nadále vyžadovat nainstalovaný Microsoft Access.

- Možnost tvorby přístupových práv pro různé uživatele nebo jejich skupiny.

2.4 Výhody a nevýhody řešení pomocí Accessu

Velkou výhodou Accessu je jeho dostupnost v celé fakultní nemocnici a na lékařské fakultě. Access také nabízí velice rychlý vývoj databáze a následně uživatelského rozhraní. V tomto nástroji je přístupných velké množství funkcí ke snadnému naprogramování databáze, které by jinak musel vývojář v jiných vývojových nástrojích dodatečně programovat.

Nevýhodou databáze vytvořené v Accessu je nutnost mít dostupný program Microsoft Access z každého počítače, na kterém hodláme aplikaci spouštět (např. síťová instalace balíku Office).

3. Popis použitých funkcí Accessu

3.1 Nástroje pro tvorbu tabulek

Pro tvorbu tabulek bylo využito návrhové zobrazení, které poskytuje standardní funkce ve volbách vlastností pole. Patří sem volba datového typu, volitelně jeho rozsahu, nastavení indexace, výchozí hodnoty a nastavení pole jako primárního klíče. V každé tabulce jsem vytvořil identifikační pole s primárním klíčem, jež zaručuje jednoznačnost každého záznamu. Identifikační pole jsou pak využita v relacích mezi společnými sloupci souvisejících tabulek. Jelikož Access integruje databázový stroj a grafické rozhraní k databázi v jedinou aplikaci, lze díky tomu u každého pole již při návrhu tabulky zvolit formát zobrazení polí, který pak automaticky přejímají nově vytvořené ovládací prvky ve formulářích. Další výhodou integrace je možnost nastavení vyhledávání. To je funkce, která doplní ovládací prvek o možnost výběru ze seznamu předdefinovaných hodnot. Ve formuláři je prvek zobrazen například jako rozbalovací seznam a uživatel místo ručního vepisování hodnoty do pole může vybrat hodnotu z nabídnutého seznamu. Obsahem vyhledávacího seznamu může být nejen textový řetězec s hodnotami, ale i výsledek vyhledávacího SQL dotazu nebo přímo pole jiné tabulky. Této funkce je možné využít díky relacím mezi souvisejícími poli tabulek.

3.2 Nástroje pro tvorbu relací mezi tabulkami

Pro návrh relací je v Accessu určeno samostatné dialogové okno, jež zobrazuje diagram databázových tabulek a jejich relací. Relace slouží k zamezení výskytu redundantních dat. Data se ukládají dle svého druhu a významu do samostatných tabulek, jejichž určitá související pole jsou provázána relacemi. Relace vysvětlím na příkladu, který bude využívat jedné tabulky pacientů a druhé tabulky ošetřujících lékařů. Bez použití databázových relací by každý záznam, kromě dat pacienta, obsahoval i jméno, příjmení, adresu ordinace lékaře, apod. A to i v případě, že by různí pacienti měli stejného lékaře. Tímto dochází k redundanci dat, každý lékař by se v tabulce pacientů musel vyskytovat tolikrát, kolik má pacientů. To přináší další problém s aktualizacemi údajů o lékařích. V případě jakékoliv změny v údaji o lékaři by se musel aktualizovat každý záznam pacienta, který daného lékaře obsahuje. Vývojář databáze by tak musel čelit nekonzistenci, což je nejednotnost záznamů v tabulce, která mají být shodná.

Relace umožňují rozdělit záznamy pacientů a jejich lékařů do samostatných tabulek a tím docílit toho, že každý lékař, ale i pacient se bude v příslušné tabulce vyskytovat pouze jedenkrát. Jednoznačnost každého záznamu v tabulce určuje pole, které nazýváme primárním klíčem. Vezmeme-li v úvahu výše popsaný příklad, pak do tabulky pacientů přidáme navíc ještě jedno pole, tzv. pole s cizím klíčem, jehož obsah bude totožný s primárním klíčem záznamu v tabulce lékařů. Tato dvě pole budou propojena relací typu 1:N, protože primární klíč jednoho lékaře se může vyskytovat v několika záznamech pacientů jako cizí klíč. Existují ještě další typy relací, jejich výčet nyní uvedu:

- **1:N** Popis viz. výše. Počet záznamů s cizím klíčem odkazující se na shodný primární klíč záleží pouze na rozsahu datového typu pole a schopnosti databázového stroje ukládat potřebný počet záznamů (maximální velikost databáze).

- **M:N** V relaci M:N může počet záznamů jedné tabulky odpovídat více záznamům jiné tabulky. Pro vytvoření tohoto typu relace je nutné vytvořit třetí tabulku, kterou nazýváme spojená tabulka, jejíž primární klíč se skládá z cizích

klíčů dvou tabulek, které chceme relací M:N propojit. Tato tabulka je tedy jakýmsi prostředníkem mezi těmito tabulkami.

- **1:1** Relace, která dovoluje vytvořit ke každému záznamu první tabulky pouze jediný záznam tabulky druhé a naopak. Tato relace se v praxi nepoužívá příliš často, ale v mé databázi našla své typické využití. Tato relace se totiž používá v případě, kdy požadujeme rozdělení velkého množství dat (sloupců), které spolu jedinečně souvisí, do více tabulek. Například tehdy, očekáváme-li, že data v druhé tabulce budou obsahovat menší počet záznamů, než tabulka první a budeme tím chtít ušetřit diskovou kapacitu.

V Accessu je možné ke každé relaci nastavit ještě referenční integritu. Referenční integrita zajistí platnost relací mezi záznamy v souvisejících tabulkách. To znamená, že pomocí systému pravidel se zamezí případům, kdy by se nějaká relace v určitém záznamu odkazovala na záznam související tabulky, který by již nemusel existovat. Doplňkově lze k této funkci nastavit automatické aktualizace nebo odstraňování souvisejících polí v kaskádě. Při pokusu uživatele o odstranění nebo aktualizaci záznamu v primární tabulce (v popisovaném příkladu tabulka lékařů), dojde k odstranění nebo aktualizaci všech záznamů ve vedlejší tabulce (tabulka pacientů). S touto volbou je třeba pracovat opatrně, aby nedošlo k nechtěnému odstranění nebo přepsání dat.

Nedostatkem Microsoft Accessu v návrhu relací je absence nástroje umožňujícího rychlý vizuální návrh a modelování databáze v jediném pracovním okně. Z relace je možné se přepnout do návrhu tabulky, ale už není možné v okně s diagramem relací vytvářet nové tabulky. Jednou ze zajímavých aplikací, která umožňuje komplexní vizuální návrh databází v E-R diagramech, je DBDesigner. V této Open Source aplikaci (aplikace s otevřenými zdrojovými kódy) lze po grafickém návrhu databáze vygenerovat SQL příkazy, které automaticky vytvoří celou databázi. Přestože nejde o aplikaci komerční firmy, má vysoký komfort ovládání a jsou s ní velmi dobré zkušenosti i při návrhu složitějších databází [9]. Jejím dalším přínosem jsou tiskové výstupy s přehledným rozvržením databáze, včetně zobrazení datových typů polí.

Pro úplnost bych rád doplnil, že název „relace“ pro vztahy mezi databázovými tabulkami není zcela správný. Toto označení bylo v poslední době zřejmě převzato z pojmu „relační databáze“, ale slovo „relační“ nepředstavuje vztahy mezi tabulkami. Pojem se týká tabulek jako relací (matematicky lze tabulku chápat jako relaci) a zejména relačního kalkulu, na kterém je postaven dotazovací jazyk, dnešní SQL [10]. Konceptem relačního modelu pro databáze se jako první zabýval v 70. letech minulého století Edgar F. Codd, britský počítačový vědec. Jeho myšlenky byly společností IBM, jeho tehdejším zaměstnavatelem, zpočátku odmítány, a proto je Edgar. F. Codd zveřejnil. IBM začala vyvíjet databáze založené na principech relačního modelu až v době, kdy tuto technologii začaly využívat konkurenční firmy [11]. Přesto v této bakalářské práci pro vztahy mezi tabulkami používám pojem „relace“, který je použit v české verzi Microsoft Access a jeho dokumentaci.

3.3 SQL dotazy

Jazyk SQL je dotazovacím jazykem pro práci s relačními databázemi, jehož prvky, syntaxe a chování byly definovány v rámci mezinárodních ANSI a ISO standardů [12]. Standardní jazyk SQL není procedurálním jazykem, což znamená, že je orientován přímo na výsledky jednotlivých příkazů místo na formalizaci postupů k dosažení výsledků [13]. Access 2003 dodržuje v současné době velmi používaný standard jazyka SQL označovaný pojmem SQL-92 [7]. Zatím nejnovějším standardem je SQL-99, který reaguje na potřeby nejmodernějších databází s objektovými prvky.

SQL dotazy lze použít jako datový zdroj formulářů a tím zvýšit možnosti uživatele, jak si data zobrazí (pořadí záznamů, filtrování dle obsahu polí, apod.). Další možností je využití dotazů na jednorázové manipulace s daty, například aktualizace obsahu některých polí, apod. V této aplikaci budou v budoucím vývoji SQL dotazy využity k zobrazení uložených dat podle nejrůznějších rozsáhlých kritérií stanovených uživateli této aplikace.

3.4 Formuláře

Pro tvorbu uživatelského prostředí v Accessu slouží formuláře, jež jsou pro uživatele rozhraním k prezentaci dat a mohou přijímat uživatelský vstup nebo na něj reagovat. V návrhovém režimu se při jejich vývoji postupuje podobným způsobem, jaký je obvyklý v běžných vývojových nástrojích (Delphi, Visual Studio .NET apod.). Programátor má k dispozici sadu ovládacích prvků, které vkládá na formuláře v návrhovém zobrazení. Každý ovládací prvek může mít určen datový zdroj, který dle potřeb programátora odpovídá zvolenému poli zdrojové tabulky.

3.5 Visual Basic for Applications

Visual Basic for Applications je programovací jazyk odvozený z Visual Basicu společnosti Microsoft a je s ním syntakticky shodný. Tento jazyk byl implementován do Accessu a jiných aplikací sady Office proto, aby se ještě zvýšily možnosti interakce dokumentů, sešitů a databází s uživatelem nebo operačním systémem. Jazyk VBA dává programátorovi široké možnosti v objektově orientovaném programování v rámci databázové aplikace [14].

V této aplikaci jsem VBA používal ve formulářích k naprogramování funkcí, které v sobě Access nemá přímo implementovány nebo byly použity k výměně informací mezi touto aplikací a NIS Medea. Díky tomuto programovacímu jazyku může být databázová aplikace v Accessu pro uživatele stejně komfortně ovladatelná jako jiné aplikace vyvinuté v některém jiném profesionálním vývojovém prostředí. Součásti editoru jazyka VBA jsou prostředky k ladění aplikace, tzv. debugger [14].

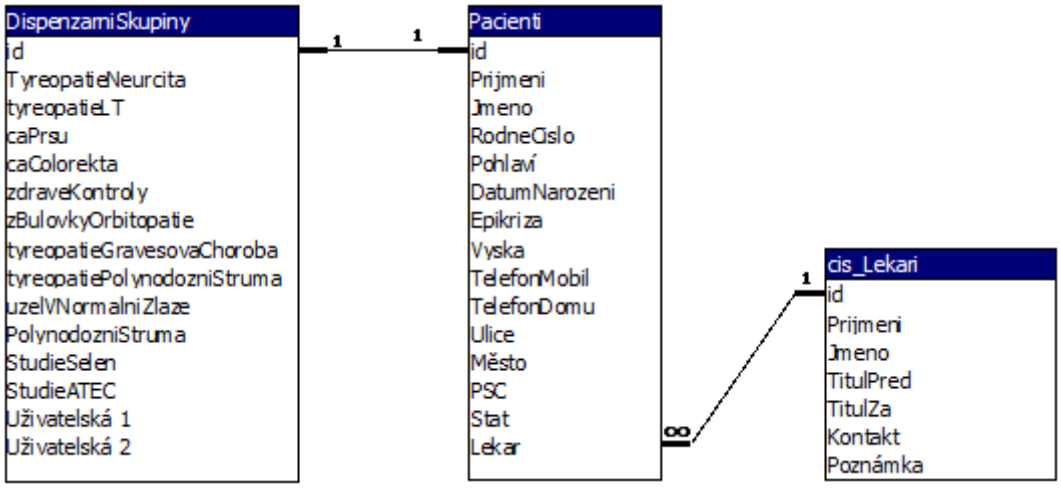
4. Návrh tabulek a jejich relací

4.1 Tabulky pacientů, lékařů a dispenzárních skupin

Tabulka pacientů obsahuje veškerá pole důležitá pro identifikaci pacienta. Jako její primární klíč není použito rodné číslo, ale pole `ID` s celočíselným datovým typem. Dovoluje nám to ukládat záznamy pacientů ze zahraniční, jež rodná čísla nemají nebo

pořizovat záznamy z vyšetření anonymních pacientů. Kromě identifikačních údajů pacienta, adresy a jiných kontaktních údajů se v tabulce nachází pole `Epikriza`, jehož obsah lékaři požadují zobrazit téměř v každém formuláři aplikace. Toto pole poskytuje rychlý přehled o tom, s jakými obtížemi pacient přišel na vyšetření a jak postupuje jeho léčba. K této tabulce bylo doplněno pole `Vyska` pro ukládání tělesné hmotnosti. Hodnota do tohoto pole se zapisuje jen jednou a proto bylo zvoleno jeho umístění právě do tabulky pacientů. Eliminovala se tak situace, kdy by se toto pole v databázi vyskytovalo ve více tabulkách, které jsou součásti formulářů pro odlišná vyšetření. Pole `Lekar` obsahuje cizí klíč tabulky `cis_Lekari`. Je to lékař, který doporučil vyšetření pacienta na 3. interní klinice. Můžeme tak sledovat, od jakých lékařů pacienti na vyšetření přicházejí nebo to lze využít pro další komunikaci. Pole `Pacienti.Lekar` a `cis_Lekari.ID` jsou svázána pomocí relace typu 1:N.

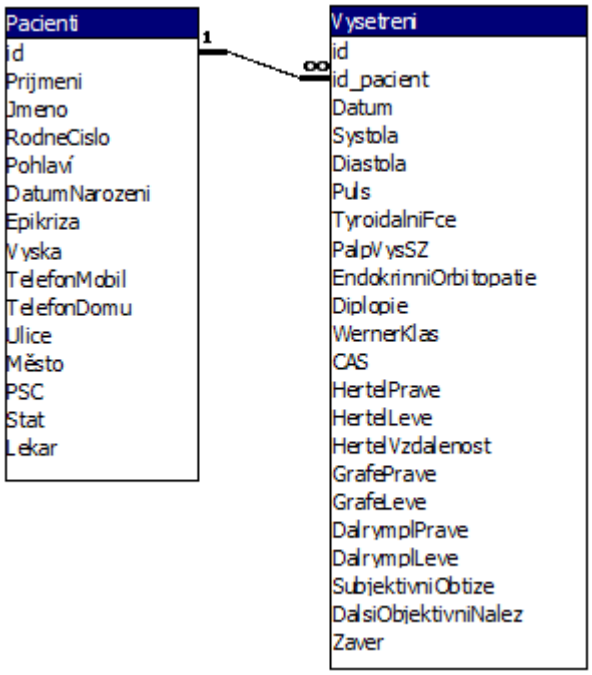
S tabulkou pacientů úzce souvisí tabulka dispenzárních skupin pojmenovaná `DispenzarniSkupiny`. Dispenzární skupiny jsou s tabulkou pacientů spojeny relací typu 1:N. Toto řešení bylo zvoleno z důvodů zachování přehlednosti tabulky pacientů a také z důvodu úspory ukládaných dat, jelikož ne každý pacient je v nějaké dispenzární skupině. Názvy polí tabulky `DispenzarniSkupiny` odpovídají dispenzářům, které mají lékaři vytvořený v NIS Medea. Pacienti se zařazují do dispenzárních skupin buď podle svých onemocnění nebo podle studie, ve které jsou jejich údaje využívány. Každý pacient může být součásti několika dispenzárních skupin, jelikož jednotlivé záznamy dispenzárních skupin se vzájemně nemusí vylučovat. Vyhledávání pacientů dle dispenzárních skupin významným způsobem v budoucnu ulehčí práci lékařů s touto aplikací, neboť si lékaři budou moci ke svému zpracování dat zobrazit pouze data pacientů, která jsou součástí studií, kterými se právě zabývají. Seznam polí popisovaných tabulek a jejich relací je patrný z obrázku č. 1.



Obrázek č. 1 Tabulky pacientů, lékařů, dispenzárních skupin a relace mezi nimi.

4.2 Tabulka klinických vyšetření

Tabulka klinických vyšetření s názvem `Vysetreni` byla navržena s ohledem na pokrytí co největšího počtu ukládaných atributů při vyšetření pacienta, aby nedocházelo ke komplikacím s absencí některých dat při exportu zpráv do NIS Medea nebo při následném vyhledávání údajů pro studie. K identifikaci jednotlivých záznamů slouží pole `ID` s primárním klíčem a pole `ID_Pacient` s cizím klíčem, jehož číslo odpovídá primárnímu klíčí pacienta v tabulce `Pacienti`. Pole `Pacienti.ID` a `Vysetreni.ID_Pacient` jsou svázána relací 1:N. Pole obsažená v tabulce klinických vyšetření jsou patrná z obrázku č. 2. V následujících odstavcích popíši nejzajímavější pole této tabulky.



Obrázek č. 2 Seznam polí v tabulce klinických vyšetření.

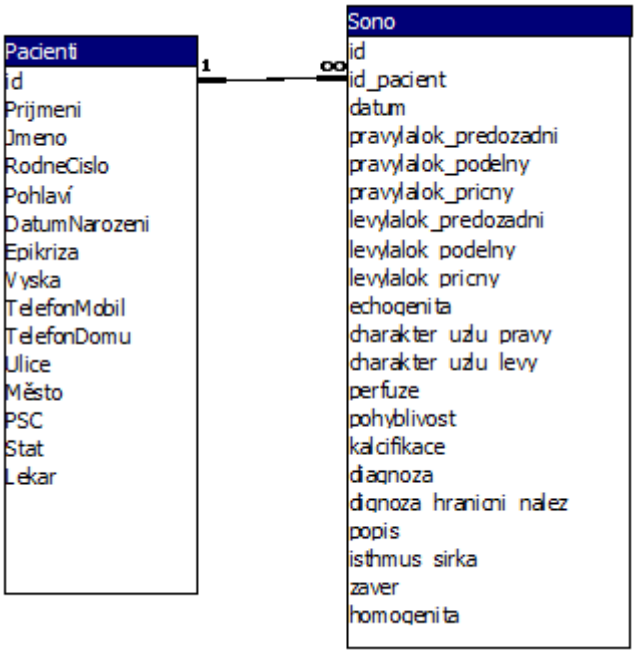
První skupinu polí tvoří `datum` vyšetření, `systola`, `diastola` a `puls`, které lékař vyplní na začátku vyšetření pacienta. Pole `SubjektivniObtize` slouží k zapsání významných subjektivních obtíží, jež chce lékař zapsat do zprávy. V současné době nám toto řešení neumožňuje v databázi sledovat výskyt konkrétních subjektivních obtíží a jejich zlepšování nebo zhoršování v čase. Proto má toto pole zatím význam pouze pro export informací do NIS Medea. Vyřešení této funkce ztěžuje fakt, že obtíží je několik desítek a v současné době není zcela rozhodnuto, jaké subjektivní obtíže by lékaři chtěli sledovat. Navíc, každý lékař do svých studií vyžaduje sledování odlišných obtíží a jejich předem definovaný seznam by znemožňoval dokonalé využití potenciálů této databázové aplikace. Začal jsem proto v současné době vyvíjet novou verzi tabulky a formuláře pro klinické vyšetření, jež by lékařům s dostatečným komfortem nabízela vytváření vlastního seznamu subjektivních obtíží a jejich volitelné přidávání do jednotlivých klinických vyšetření. Tyto parametry dosud nebyly kvůli své složitosti podrobně sledovány v žádné studii, a proto tato funkcionalita může významně přispět do studií, v nichž bude tato aplikace využita. Pole `TyroidalniFce` a `PalpVysSZ` slouží k uložení základních údajů o štítné žláze – o její funkci a o jejím palpačním

vyšetření. K zaznamenání podrobných informací o štítné žláze, které přináší ultrazvukové vyšetření, se věnuji v následující kapitole 4.3.

Významnou skupinu polí tvoří pole pro ukládání informací o endokrinní orbitopatii. Pole `EndokrinniOrbitopatie` slouží pouze k zaznamenání, má-li pacient endokrinní orbitopatii. Pokud ano, může lékař vyplnit ostatní parametry, které k ní náleží. Jsou to pole `WernerKlas` (Wernerova klasifikace), `CAS` (Clinical activity score), `HertelPrave`, `HertelLeve` (Hertel pro pravé a levé oko), `HertelVzdalenost` (vzdálenost v milimetrech pro pozorování Herteleho příznaku), `GrafePrave`, `GrafeLeve` (Grafeho příznak na pravém a levém oku), `DalrymplPrave`, `DalrymplLeve` (Dalrymplův příznak pro pravé a levé oko). Pole `DalsiObjektivniNalez` a `Zaver` slouží k uložení informací, které se pouze exportují do NIS. Neobsahují informace, jež by bylo nutné rozdělovat do speciálních polí, abych musel zajistit jejich snadné zpracování do statistik.

4.3 Tabulka sonografických vyšetření

Tabulka sonografických vyšetření `Sono` obsahuje údaje získané v ordinaci při ultrazvukovém vyšetření štítné žlázy. Jednotlivá pole tabulky se dají rozdělit na tři větší skupiny. První skupinu tvoří textové hodnoty pro posouzení parametrů štítné žlázy (`homogenita`, `echogenita`, `perfuze`, `charakter_uzlu_pravy`, `charakter_uzlu_levy`, `pohyblivost` a stanovení diagnózy v poli `diagnoza`. Je-li diagnóza hraniční, použije lékař pole `diagnoza_hranicni_nalez`. Druhou skupinou jsou pole pro zadání rozměrů štítné žlázy, které lékař změří pomocí speciální funkce na displeji ultrazvukového přístroje. Jde konkrétně o rozměry příčné (`pravylalok_pricny`, `levylalok_pricny`), předozadní (`pravylalok_predozadni`, `levylakok_predozadni`) a podélné (`pravylalok_podelny`, `levylalok_podelny`) pravého a levého laloku a šířky isthmu (`isthmus_sirka`). Všechny rozměry se zadávají v milimetrech. Zbývající textová pole `Popis` a `Zaver` slouží k uložení ostatních informací pro export do NIS. Seznam všech polí v tabulce `SONO` je vyjmenován v obrázku č. 3.



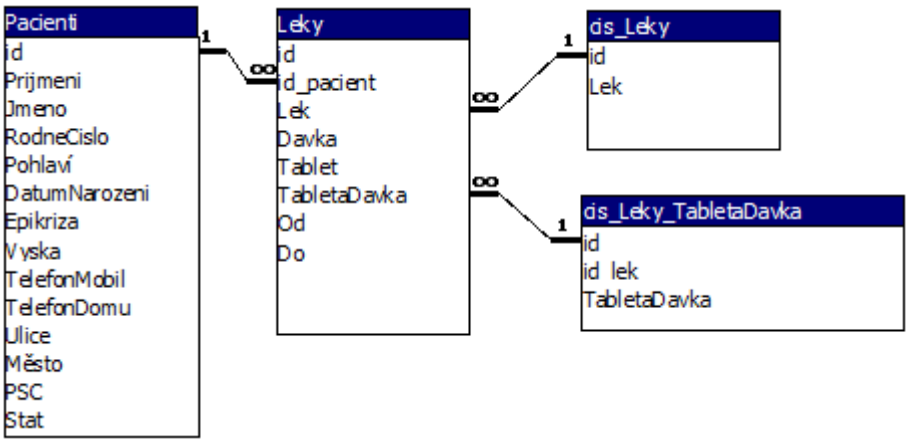
Obrázek č. 3 Seznam polí tabulky sonografických vyšetření.

4.4 Tabulky pro doporučenou medikaci

Tabulky pro doporučenou medikaci vznikly z toho důvodu, že lékaři požadují ve svých studiích sledovat léky předepsané pacientům. Do databáze se ukládají pouze generické názvy léčiv, aby se předešlo duplicitě záznamů. Hlavní tabulkou pro doporučenou medikaci je tabulka `Leky`, jež obsahuje základní pole `Lek`, které je cizím klíčem pole `ID` tabulky `cis_Leky`. Pole `Davka` obsahuje hmotnost předepsané dávky léku. Jednotka hmotnosti dávky není předem dána vzhledem k řádovým rozdílům předepsaných dávek léků. Většina se jich předepisuje buď v mikrogramech nebo v miligramech. Jednotka hmotnosti je součástí názvu léku `Lek` v tabulce `cis_Leky`. Další pole `Leky.Od` a `Leky.Do` obsahují datum, kdy bylo započato u ukončeno braní předepsaného léku. Když je právě lék předepsán, nezapisuje se předem datum ukončení, aby lékaři měli přehled o lécích, které pacient v současné době užívá. Dojde-li pak ke změně dávky léku, zapíše lékař datum ukončení braní léku a vytvoří nový záznam. Tabulku doplňují ještě dvě nepovinná pole `Tablet` a `TabletaDavka`, která lékařům zjednodušují práci s aplikací, aby nemuseli sami přepočítávat celkovou hmotnost předepsané dávky.

Tabulka `cis_Leky` obsahuje seznam léků, které lze přidávat do tabulky `Leky`. Povinný formát názvu léku obsahuje na konci v hranaté závorce jednotku obvyklé předepisované dávky. Toto je ošetřeno ve formuláři pro tuto tabulku, kterým se zabývám v kapitole 5.4.

Tabulka `cis_leky_TabletaDavka` má za účel opět zjednodušit lékařům práci s databázovou aplikací. Obsahuje v poli `TabletaDavka` obvyklé dávky léků v jedné tabletě. Toto pole pak slouží jako zdroj řádků v poli `TabletaDavka` tabulky `Leky`. Aby se v tabulce `Leky` zobrazily pouze dávky v tabletách, které náleží k určitým lékům, doplňuje tabulku `cis_leky_TabletaDavka` pole `id_Lek` s cizím klíčem z tabulky `Leky`. Toto pole slouží jako filtr, aby se lékařům v nabídce dávek v tabletě zobrazovaly pouze dávky v tabletách, které k vybranému léku patří. Jednotka dávky v tabletě musí odpovídat jednotce, která je uvedena v názvu léku. Návrh popisovaných tabulek a relací mezi nimi je zobrazen na obrázku č. 6.



Obrázek č. 6 Návrh tabulek pro doporučenou medikaci.

4.5 Tabulka biochemických vyšetření

Vývoj tabulky pro biochemii si vyžádalo největší podíl času při vývoji databáze. Tabulka `Biochemie` slouží k uchovávání výsledků biochemických vyšetření, která mají lékaři dostupná z NIS Medea. Všeobecná fakultní nemocnice v Praze má centrální biochemickou laboratoř, ve které používají jejich vlastní laboratorní informační systém. K analýze odběrů od pacientů se v laboratoři využívají ve většině případů robustní automatické analyzátory. Tyto analyzátory umožňují dodatečné naprogramování komunikace s laboratorním informačním systémem, což umožní automatické ukládání výsledku do databáze LIS. Ten pak po potvrzení správnosti výsledků zodpovědným lékařem odesílá údaje o provedených biochemických vyšetřeních do NIS Medea.

Výsledky z biochemie jsou ukládány do databáze Medey společně s jinými výsledky vyšetření, které se ve VFN provádějí. Importu dat z výsledků z Medey do mé aplikace bylo docíleno tak, že jsem využil funkce mixéru ve výsledcích NIS. Výsledky jsou v něm uloženy s velkým množstvím doplňujících informací a parametrů. Mixér umožňuje lékařům tyto údaje volitelně zobrazovat a tvořit tak textové výstupy s výsledky dle vlastních požadavků. V mixéru jsem využil následující parametry:

- **Datum**: Výstup mixéru byl nastaven tak, aby rozdělil výsledky do skupin podle data provedení vyšetření daného vzorku. Jelikož mohou nastat situace, kdy v jeden den může být provedeno více identických vyšetření, je v datumu zobrazen i čas, který pomůže rozlišit jednotlivá vyšetření provedená v jeden den.

- **Třída**: Naprostá většina analytů je přiřazena do určité třídy, která charakterizuje danou skupinu analytů, např. vyšetření moče chemicky, apod. Pokud analyt nemá definovanou třídu, je v textu názvu třídy vypsána informace, že analyt do žádné třídy nebyl přiřazen. V exportu z Mixéru tvoří třída další vnořenou podskupinu v rámci data biochemického vyšetření.

- **Plný název**: Jde o delší verzi názvu analytu, kterou jsem upřednostnil před zkráceným názvem, který je pro lékaře nepřehledný a není z něj snadno rozlišitelné, o jaký analyt a případně metodu analýzy se jedná.

- **Výsledek bez jednotky**: Výsledky v exportu nemají zobrazenu jednotku, jelikož ji lékaři znají a hodnota se v exportu zobrazuje vždy v předem daném rozměru. Pro uchování výsledků do mé databáze nemá jednotka podstatný význam.

Databázi jsem pak navrhnul tak, aby se přizpůsobila možnostem exportních funkcí Medey a zároveň umožnila následné zpracování jednotlivých hodnot analytů dle požadavků lékařů. Níže uvádím seznam důležitých polí, které obsahuje každý záznam v mé tabulce analytů:

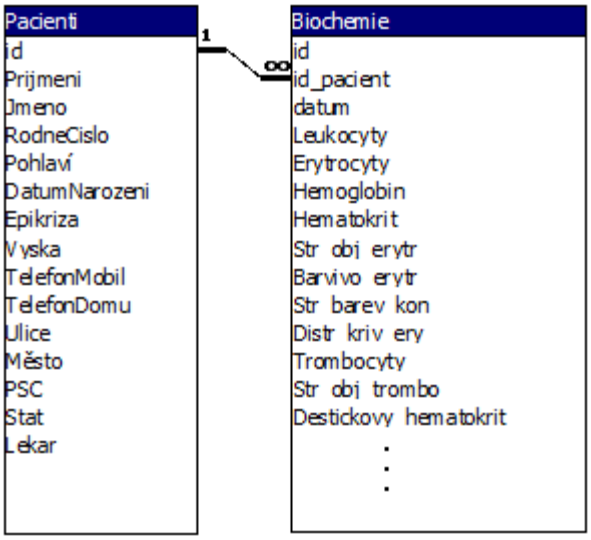
- **`ID`**: Primární klíč.

- **`ID_Pacient`**: Cizí klíč určující záznam pacienta v tabulce `Pacienti`.

- **`Datum:`** Podle data a času rozlišuji jednotlivé záznamy, do kterých patří analyty.

- **`Analyt`**: Název analytu v mé tabulce je jedinečný, v tabulce již nepoužívám rozlišení, ke které třídě analyt patří.

Na obrázku č. 7 je schéma tabulky `Biochemie`, pro přehlednost v ní nejsou obsažena všechna pole s analyty, kterých je několik desítek. Pří návrhu tabulky `Biochemie` jsem se musel vypořádat s několika nepříjemnostmi, které nemám možnost v Medee ovlivnit. Stává se, že dva různé analyty nebo stejný analyt, ale získaný jinou metodou, jsou v databázi Medey uloženy pod stejným plným názvem. Při importu do mé aplikace jsem se s tím vypořádal tak, že rozlišuji, ve které třídě se daný analyt nachází. Objevil jsem ale i situaci, kdy se v jedné třídě nacházejí analyty se stejným plným názvem, ale přitom jde o zpracování jinou metodou. Avšak zkrácené názvy těchto analytů již byly odlišné. Nedokáži určit, jestli šlo o něčí nepozornost při zadávání parametrů buď do NIS nebo LIS. Tuto situaci bych již při importu nemohl řešit. Zatím jsem takový analyt na žádost lékařů do mé databáze nepřidával, ale v případě, že by tento požadavek vznikl, musel bych požádat o spolupráci biochemickou laboratoř, aby dané názvy analytu pozměnila nebo nalézt jinou selekční metodu.

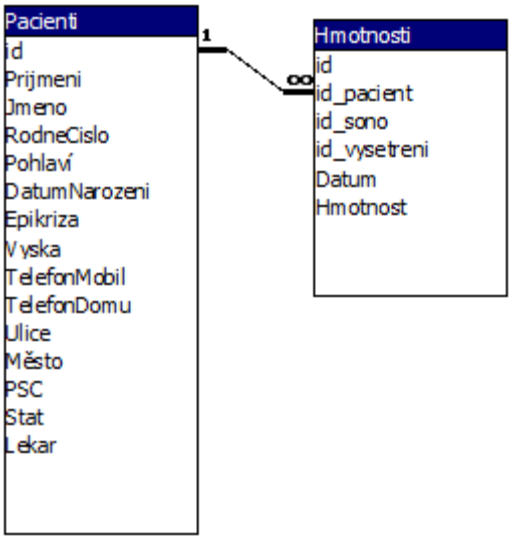


Obrázek č. 7 Základní pole tabulky biochemie s naznačením polí s analyty.

Z výše uvedeného textu tedy vyplývá, že neimportuji všechny biochemické výsledky, ale jen ty, které si lékaři předem zvolili. Pokud lékaři chtějí dodatečně přidat do tabulky další analyt, mají v aplikaci možnost kontroly, které analyty nebyly importovány. Tabulku musím doplnit já, jelikož Access neumožňuje snadné prostředí pro manipulaci s návrhem tabulky. Této situaci a vlastnímu importu výsledků do aplikace, řešeném pomocí Visual Basicu, se věnuji v kapitole 5.5.

4.6 Tabulka hmotností

Funkcí tabulky `Hmotnosti` je shromažďovat hmotnosti pacientů z různých formulářů, které lékaři využívají při vyšetřeních. Ukládání dat do této tabulky je implementováno pomocí kódu Visual Basicu ve formulářích pro klinické a sonografické vyšetření. Takové řešení bylo zvoleno z toho důvodu, že lékaři potřebují mít jednotný přehled o vývoji hmotnosti pacienta, ať už kvůli správnému stanovení léčby, nebo kvůli využití těchto údajů ve studiích. Záznam v této tabulce obsahuje, kromě samotné hmotnosti, i nevlastní klíč záznamu klinického nebo sonografického vyšetření. Důvodem je identifikace vyšetření, ve kterém byl záznam vytvořen. Seznam polí v tabulce hmotností je v obrázku č. 8.



Obrázek č. 8 Seznam polí tabulky hmotností a relace s tabulkou pacientů.

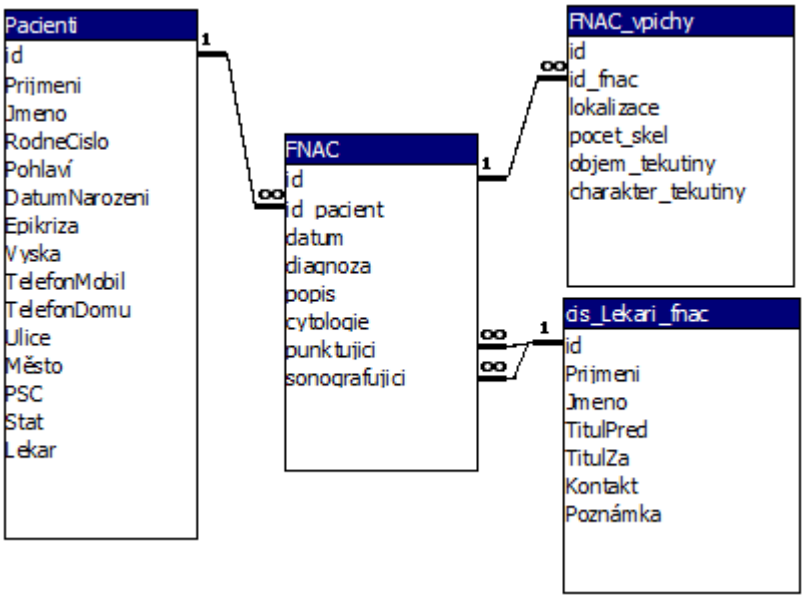
4.7 Tabulky pro aspirační cytologii tenkou jehlou

FNAC (fine needle aspiration cytology), nebo-li aspirační cytologie tenkou jehlou, slouží k odběru vzorků ze štítné žlázy pomocí punkce. Tato tabulka vznikla z potřeby přehledu o provedených punkcích a hlavně výsledků cytologických vyšetření. Navíc, formulář, jehož zdrojem je tato tabulka, lékařům urychluje práci při psaní žádanek na vyšetření vzorků v cytologické laboratoři.

Základní tabulkou je `FNAC`, která obsahuje záznamy s informací o datu provedení odběru v poli `datum`, diagnóze (pole `diagnoza`) a výsledku cytologického vyšetření (pole `cytologie`). Tabulka obsahuje ještě dvě pole `punktujici` a `sonografujici` s nevlastním klíčem z tabulky `cis_Lekari_fnac`, která obsahuje seznam lékařů provádějí punkci štítné žlázy. Důvod pro vytvoření samostatné tabulky lékařů pro punkci místo využití tabulky ošetřujících lékařů `cis_Lekari` byl ten, že tyto dvě skupiny lékařů spolu nemají nic společného a uživatelé pro zjednodušení práce s aplikací chtěli mít seznam lékařů pro FNAC v samostatné tabulce. Struktura tabulky `cis_lekari_fnac` je identická s tabulkou `cis_Lekari`.

Druhou důležitou tabulkou pro FNAC je tabulka jednotlivých punkcí, která má v záznamu pole s nevlastním klíčem z tabulky `FNAC`. Do této tabulky lékaři ukládají

lokalizaci provedeného vpichu (pole `lokalizace`), počet sklíček (pole `pocet_skel`) se vzorkem nebo objem (`objem_tekutiny`) a charakter (`charakter_tekutiny`) odebrané tekutiny. Schéma s relacemi a poli v popisovaných tabulkách je vyobrazeno na obrázku č. 9.

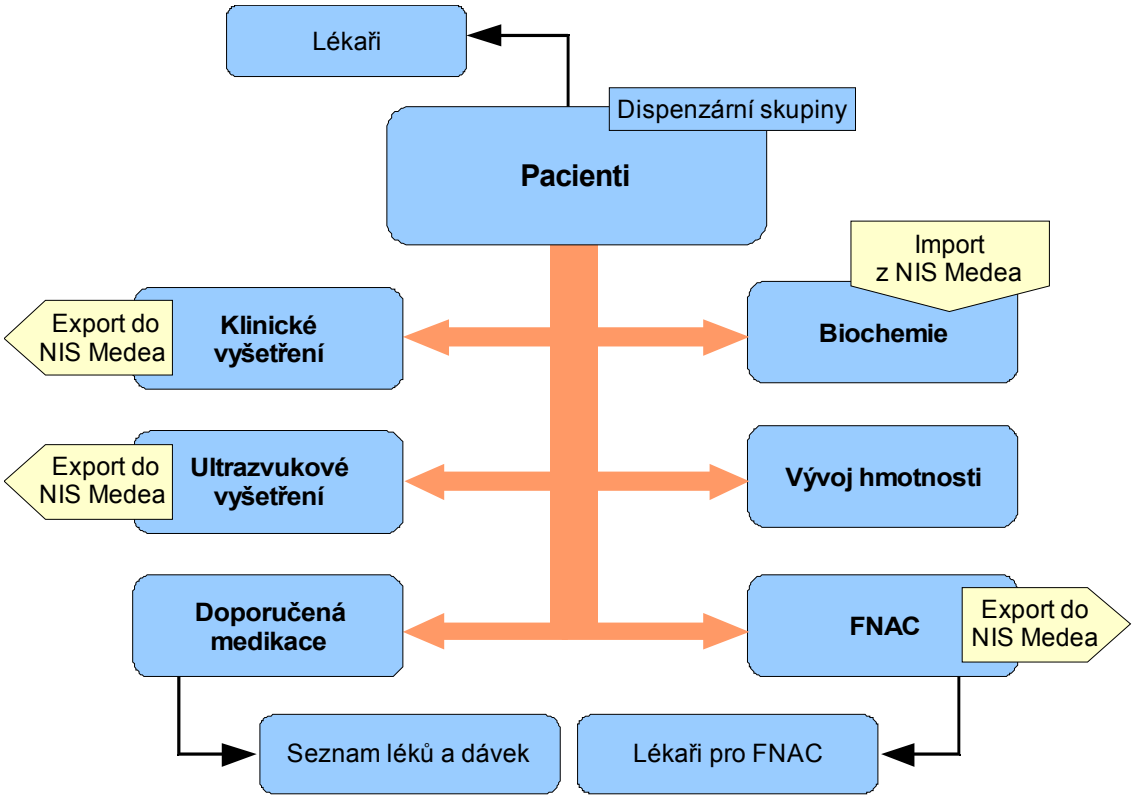


Obrázek č. 9 Schéma tabulek a relací pro aspirační cytologii.

5. Návrh grafického uživatelského rozhraní a kódu ve Visual Basic for Applications

Databázová aplikace je z uživatelského hlediska rozdělena na sedm modulů, které obsahují různý počet formulářů. Databázový soubor `Tyrex-BC.mdb` má nastaveno automatické zobrazení formuláře `frm_Pacienti` po spuštění databázové aplikace. Ostatní moduly byly řešeny vždy tak, že jejich hlavní formulář obsahuje jako zdroj dat tabulku `Pacienti`, ze které zobrazuje některá pole a je v něm vnořený formulář, který již slouží ke specifické funkci modulu, např. pro klinické vyšetření, atd. V každém modulu se kromě základních identifikačních údajů pacienta zobrazuje i jeho epikríza a je možné zobrazit formulář s dispenzárními skupinami. Zatímco hlavní formulář s pacienty se při spuštění aplikace zobrazí v režimu pro zadávání nového pacienta, tak moduly pro konkrétní vyšetření zobrazí vždy poslední vytvořený záznam,

pokud existuje. Schéma modulárního uživatelského prostředí je zobrazeno v obrázku č. 10.



Obrázek č. 10 Schéma modulů databázové aplikace

5.1 Modul pacienti

Modul pacienti obsahuje seznam pacientů a jejich identifikační údaje. Datovým zdrojem formuláře je tabulka `Pacienti`. Klepnutím na tlačítko „+“ lze začít s přidáváním záznamu o novém pacientovi. Vstupní maska v poli pro rodné číslo je nastavena tak, aby se lékařům zobrazoval jednotný formát rodných číslem s lomítkem. Do databáze ale ukládám pouze číslo bez lomítka. Předcházím tak případům, kdy je každý lékař zvyklý psát rodné číslo v odlišném formátu, čímž by se ztížilo vyhledávání pacienta dle rodného čísla. Pro zadání dalšího pole stiskne lékař klávesu `TAB`, která přenese zaměření do prvku pro zadání příjmení pacienta. Tato událost je obsloužena makrem, které vyhledává, zda-li rodné číslo v tabulce již neexistuje. Pokud jej nalezne,

nabídne lékaři zobrazení příslušného záznamu. Lékaři tak nemusí zbytečně zvlášť vyhledávat již uložené pacienty.

V záložce `Dispenzární skupiny` je podformulář, jehož zdrojem dat je tabulka se seznamem dispenzárních skupin. V záložce `Kontakt` mohou lékaři zadávat případně kontaktní údaje pacienta. Každý pacient má svého lékaře, kterým byl doporučen na vyšetření na 3. interní klinice. Změnu seznamu těchto lékařů lze provést v dialogu, který se otevírá tlačítkem `Přidat lékaře...`

5.2 Modul klinických vyšetření

Z formuláře s pacienty lze tlačítkem `Klinické vyšetření` otevřít formulář s provedenými klinickými vyšetřeními, které se ukládají do tabulky `Vysetreni`. V tomto formuláři ukládají uživatelé informace o subjektivních pocitech pacienta a objektivní nálezy. Ukládání pole `Hmotnost` je v současné době řešeno pomocí Visual Basicu, protože tato hodnota se ukládá do zvláštní tabulky `Hmotnosti`. Pokud lékař vytváří nebo mění ostatní hodnoty klinického vyšetření, dojde k automatickému uložení hmotnosti, ale změní-li lékař v celém formuláři jen hmotnost, je nutné klepnout na tlačítko `Uložit`. Tuto situaci způsobuje v Accessu absence obsluhy události pro přechod mezi více záznamy formuláře. Pokud lékař v tomto případě hmotnost ručně neuloží, může tak dojít ke ztrátě informace o hmotnosti pacienta, protože Access při přechodech mezi záznamy tabulky ve formuláři ukládá pouze hodnoty v prvcích, která mají nastaven datový zdroj.

Nejzajímavější částí formuláře je `Endokrinní orbitopatie`. Pokud lékař tuto volbu označí, má možnost zadat další objektivní nálezy, které se u Endokrinní orbitopatie zjišťují. Při vstupu uživatele do jednoho z polí endokrinní orbitopatie se zobrazuje okno s nápovědou, formulář `Nápověda`. Tento formulář je nastaven tak, aby se zobrazoval nad ostatními okny aplikace a jeho funkcí je náhrada informací, které měli lékaři vytisknuté na papíře. Zkušenost byla taková, že lékaři endokrinní orbitopatii příliš nesledovali, proto nám toto řešení pomůže s pravidelným získáváním požadovaných údajů. Záložka `CAS` ve formuláři s nápovědou, která se zobrazí při

vstupu uživatele do pole pro zadávání `Clinical Activity Score`, je navíc interaktivní. CAS se stanovuje na základě součtů bodů za každý příznak, který lékař u pacienta stanoví. Tlačítkem `Zkopírovat do vyšetření` se přenese hodnota do příslušného pole ve formuláři klinického vyšetření. Formulář s nápovědou se automatický uzavírá, pokud lékař opustí rámec pro endokrinní orbitopatii.

Z formuláře klinického vyšetření je ještě po stisku příslušného tlačítka dostupný modul pro zadávání doporučené medikace. Po stisku tlačítka `Zpráva` se vygeneruje zpráva pro NIS Medea a automaticky se zkopíruje do schránky Windows. Lékař pak jen v Medee vloží vygenerovanou zprávu a dopíše jen případně předepsané léky. Vzhledem k tomu, že do databázové aplikace ukládáme pouze generické názvy léčiv, které se do Medey nepíší, není součástí generované zprávy seznam léků, které má pacient aktuálně předepsány v modulu `Doporučená medikace` (kapitola 5.4). Navíc, lékaři na 3. interní klinice v současné době používají speciální program pro tisk receptů, který umožňuje v něm napsaný seznam léků zkopírovat do schránky. Pokud byl při generování zprávy nalezen nějaký záznam v tabulce sonografických vyšetření u daného pacienta, provede se vložení zprávy z posledního ultrazvukového vyšetření do zprávy o klinickém vyšetření.

5.3 Modul sonografických vyšetření

Modul pro sonografická vyšetření se z formuláře pacientů spouští tlačítkem `SONO`. Je rozdělen na tři části. První obsahuje pole s informacemi o vzhledu štítné žlázy a diagnóze. Ve druhé části lékař zadává rozměry laloků štítné žlázy, které mu zobrazuje ultrazvukový přístroj. Objemy obou laloků jsou automaticky vypočítány ze zadaných údajů a zobrazují se v rámečku `Objemy`. Součástí výpočtu je i algoritmus pro stanovení, je-li štítná žláza normální, zvětšená, či zmenšená. Algoritmus jsem vytvořil přepsáním speciální tabulky do Visual Basicu. Tato tabulka obsahovala hodnoty objemů štítné žlázy v závislosti na věku a pohlaví. Funkce pro stanovení zvětšení štítné žlázy významně ulehčuje práci lékařům, kteří by hodnotu museli složitě vyhledávat v tabulce. Tímto způsobem také dosáhnu standardizace pro stanovení, je-li štítná žláza zvětšená.

Tlačítkem `Zpráva` lze vygenerovat výslednou zprávu, která se vkládá do NIS Medea. Obsah pole se zprávou se do databáze neukládá.

5.4 Modul doporučené medikace

Účelem tohoto formuláře je sledování léků a jejich dávkování, které byly pacientům v rámci léčby předepsány. Lze zde sledovat nejen léky, které má pacient aktuálně předepsány, ale je možné sledovat léky, které byly pacientovi předepsány k užívání již dříve a jejich užívání bylo ukončeno. Tento modul je řešen tak, že lékař pro nově předepsaný lék zadá datum začátku jeho užívání. Pokud je užívání určitého léku ukončeno nebo chce lékař změnit dávkování, zapíše lékař datum ukončení užívání tohoto léku a do nového řádku tabulky zapíše lék s aktuálním datem užívání.

Ke zjednodušení práce lékařů a standardizaci názvů sledovaných léků byla vytvořena samostatná tabulka generických názvů léčiv a jejich dávek v tabletách. Tato tabulka je datovým zdrojem rozbalovacího seznamu pole `Lék` ve formuláři doporučené medikace. Ke změně záznamů tabulky s léky a jejich dávkováním slouží formulář `Seznam léků a dávek`. Lékař v něm zadá generický název léku a do hranatých závorek jednotku hmotnosti účinné dávky v tabletě. Jednotka slouží k větší přehlednosti, aby bylo zaručeno, že všichni lékaři budou do tabulky zadávat údaje se stejnou jednotkou a nedocházelo tak ke komplikacím při zpracování dat. Ke každému léku je možné zadat seznam dávek v tabletě. Lékaři si pak mohou ve formuláři doporučené medikace vybrat jednu z nabízených dávek a po zadání počtu tablet se automaticky napíše vypočtená hodnota celkové předepsané dávky. Ke zpracování je však důležitá pouze celková předepsaná dávka.

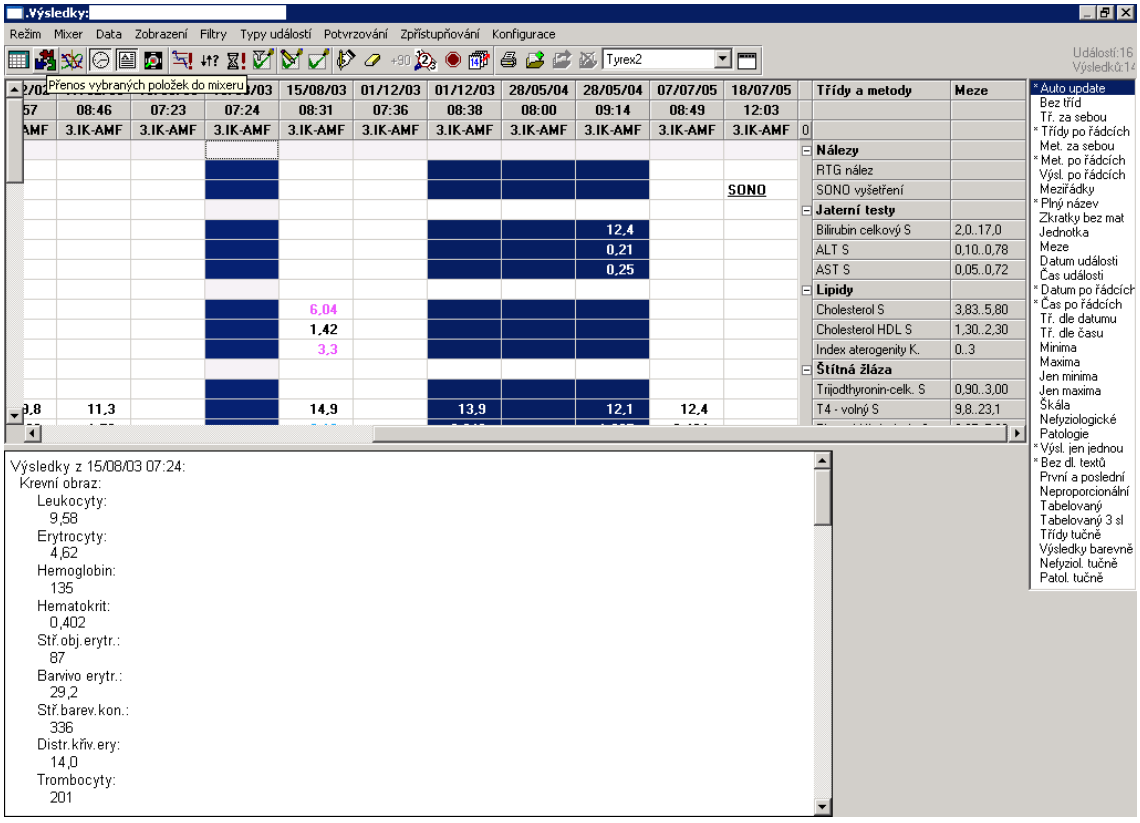
5.5 Modul biochemie

Vývoj tohoto modulu patřilo k nejsložitější práci na této aplikaci jak z hlediska analýzy, tak i z hlediska programování, ale patří k nejzajímavějším komponentám této aplikace. Modul biochemie obsahuje tabulku se seznamem výsledků biochemických vyšetření. Výsledky jsou dostupné v NIS Medea, odkud je do tohoto modulu lékaři importují. Důvod k importu výsledků je nemožnost sledování a následné zpracování

výsledků, které jsou uloženy v NIS Medea. Jejich vyhledávání je v tomto systému zdlouhavé a pokud tyto výsledky chtějí lékaři zpracovávat hromadně, stejně musí jednotlivé hodnoty nejdříve opisovat do svých tabulek.

Jednotlivá vyšetření jsou ve výsledcích NIS Medea zobrazeny ve sloupcích. Je možné je exportovat do textových výstupů dle nejrůznějších voleb v nastavení mixéru. Vybral jsem nejvhodnější nastavení mixéru, které jsem v Medee uložil do seznamu stereotypů pro opakované použití, aby si jej lékaři nemuseli pamatovat. Toto nastavení je pro mne důležité v tom, že jeho formát využívám v algoritmu, který tento textový výstup opět dekóduje a ukládá jednotlivé hodnoty analytů do tabulky. Medea je ale v práci se stereotypy mixéru nedořešená, protože stereotyp lze definovat pouze v daném uzlu Medey. Uzly slouží k rozlišení jednotlivých oddělení kliniky. Pokud se lékař v Medee nachází v jiném uzlu, může se setkat s tím, že v nabídce bude požadovaný stereotyp postrádat. Pro tyto případy je ve formuláři biochemie tlačítko `Nastavení exportu Medey...`, které lékařům zobrazí, jaké má provést nastavení mixéru. Toto nastavení se provádí aktivací voleb v pravé části okna Medey poklepáním na jednotlivé volby. Lékař pak pro další použití může dané nastavení uložit opět do stereotypu a zpřístupnit jej ostatním lékařům. V Medee lze totiž zvolit, je-li vytvořený stereotyp viditelný pro všechny uživatele nebo jen pro jeho tvůrce.

Import je v mé aplikaci řešen ve Visual Basicu a postup uživatele při importu výsledků probíhá následovně: Uživatel nejdříve v Medee označí sloupce s výsledky, které chce do databáze importovat a stiskne tlačítko `Přenos vybraných položek do mixéru`. Není potřeba volit jednotlivé řádky s výsledky, protože algoritmus již sám zjistí z předem definovaných hodnot, které analyty má importovat. Nyní je nutné zvolit formát údajů v mixéru tak, aby jej byl schopen algoritmus v mé aplikaci dekódovat. K tomu je v Medee k dispozici přednastavený stereotyp dostupný z rozbalovacího boxu v tlačítkové liště Medey. Nakonec uživatel zkopíruje obsah mixéru do příslušného okna mé aplikace a po stisku tlačítka `Vlož do tabulky` dojde ke spuštění funkce, která z textového výstupu Medey vybere analyty, jež chci do tabulky uložit a rozdělí je do záznamů podle data vyšetření. Obrázek č 11 znázorňuje okno Medey s výsledky a jejich exportem do mixéru.



Obrázek č. 11 Ukázka okna NIS Medea s výsledky a jejich exportem do mixéru.

Analyty, které nebyly do tabulky importovány, se zobrazí v seznamu neimportovaných analytů. Tento seznam má tři důležité funkce. Za prvé: pomáhá lékařům kontrolovat, zda aplikace provedla bezchybný import dat z Medey. Za druhé: názvy tříd nebo analytů mohl být na oddělení klinické biochemie změněn a algoritmus pak správně nerozezná, že měl analyt do tabulky uložit. Pro tento případ mohou uživatelé využít funkce pro uložení seznamu neimportovaných analytů do textového souboru, který mi pak mohou elektronickou poštou odeslat. Tento soubor obsahuje všechny informace k tomu, abych mohl databázi upravit a přizpůsobit ji nové situaci. A nakonec, pokud lékaři v tomto seznamu naleznou analyt, jež by chtěli nově přidat do tabulky, odesláním příslušného textového seznamu mi usnadní práci při programování databáze.

Na CD-ROM přiloženém v této práci se nacházejí dva textové soubory, které obsahují export z mixéru NIS Medea. Obsah těchto souborů lze po vložení do pole pro výsledky importovat do databázové aplikace (soubor `Tyrex-BC.mdb`). Aplikace

provede import všech výsledků ze souboru `Export1.txt` a při importu analytů ze souboru `Export2.txt` budou na ukázku vynechány analyty, které v databázi nejsou sledovány.

Tento modul bude mít využití v nejrůznějších klinických studiích, ve kterých je potřeba zpracovávat biochemické výsledky do statistických analýz. Příkladem může být studie sérových koncentrací protilátek proti tyreoidální peroxidáze korelujících ke kvantitativním ukazatelům sonogramu štítné žlázy u pacientek s karcinomem prsu. V této studii byly zpracovány kvantitativní ukazatele digitálních ultrazvukových obrázků štítné žlázy pomocí aplikace Matlab. Jako kvantitativní ukazatele byly použity stupně šedi jednotlivých pixelů v obrázku a vlastnosti textury. Následně byly porovnány výsledky těchto ukazatelů s výsledky laboratorních parametrů štítné žlázy. U pacientek s rakovinou prsu sérové koncentrace proti tyreoidální peroxidáze štítné žlázy negativně korelovaly se stupni šedi a pozitivně korelovaly s určitými vlastnostmi textury. Tato korelace je pravděpodobně způsobena strukturálními změnami ve tkání štítné žlázy. Rovněž byla zjištěna negativní korelace mezi stupni šedi a sérovými hodnotami nádorového markeru CA 15-3 [15].

5.6 Modul vývoje hmotnosti

Formulář vývoje hmotnosti obsahuje jako datový zdroj tabulku `hmotnosti`, která zobrazuje hmotnosti pacientů, jež zadali lékaři v klinickém vyšetření nebo v sonografickém vyšetření. Hmotnost lze samozřejmě ukládat zvlášť i v tomto formuláři. V tomto modulu lze na jednom místě zobrazit vývoj hmotnosti pacienta, i když se hmotnost zapisuje v různých modulech aplikace. Lékařům to pomůže ve sledování vývoje hmotnosti pacienta v závislosti na stanovené léčbě.

5.7 Modul aspirační cytologie tenkou jehlou

Modul aspirační cytologie tenkou jehlou se otevírá klepnutím na tlačítko `FNAC` v hlavním formuláři s pacienty. Tento modul slouží lékařům k usnadnění psaní žádanky pro vyšetření vzorků ze štítné žlázy v cytologické laboratoři. Jeho další funkcí je také ukládání detailů jednotlivých vpichů pro možnost sledování ve studiích.

Pole Diagnóza slouží ke zvolení diagnózy z nabízených možností. Zdrojem seznamu je tabulka cis_Diagnozy, jejichž záznamy mohou lékaři měnit ve formuláři se seznamem diagnóz. Tento formulář se otevře stiskem tlačítka Přidat diagnózu... Součástí žádanky jsou jména lékařů, kteří provedli punkci nebo při provádění punkce asistovali (pole punktující a sonografující). Oba rozbalovací seznamy na výběr lékařů mají jako zdroj dat tabulku cis_Lekari_FNAC. Seznam punktujících lékařů je ve zvláštní tabulce z toho důvodu, že tito lékaři nemají nic společného s lékaři, kteří jsou v tabulce cis_Lekari. Navíc, punktujících lékařů bude i do budoucna jen několik a lékaře by zbytečně zdržovalo vybírat z obsáhlého seznamu všech lékařů. V poli Popis je možné zadat doplňující údaje k sonografickému vyšetření. Do pole Cytologie lékaři vloží výslednou zprávu z cytologické laboratoře, kterou mohou nalézt v NIS Medea. Jednotlivé vpichy se zadávají do zvláštní tabulky. Je možné zadat lokalizaci vpichu a pak počet sklíček se vzorkem nebo objem odebrané tekutiny. Doplňkově může lékař zadat ještě charakter tekutiny. Po stisku tlačítka Zpráva se vygeneruje žádanka do pole Závěr, který lze ze schránky Windows vložit do Medey. Obsah pole Závěr do databáze neukládám.

6. Zkušenosti s používáním databázové aplikace a její další vývoj

Popisovaná databázová aplikace je vyvíjena ve spolupráci s MUDr. Janem Jiskrou, PhD. a MUDr. Ing. Danielem Smutkem, PhD., s kterými konzultuji výběr dat ukládaných do tabulek. Strukturu databázových tabulek a návrh grafického prostředí analyzuji sám a snažím se vybrat vhodný kompromis mezi komfortem v ovládání aplikace a širokými možnostmi při ukládání vyšetřovaných parametrů. Práce na této aplikaci se nezastavila s vypracováním této bakalářské práce. V aplikaci jsou jednotlivé moduly vylepšovány a doplňovány o nové funkce na základě zkušeností uživatelů. Zároveň pokračuje analýza a návrh dalších modulů aplikace. Cílem je poskytnout řešení pro ukládání dat a komplexně obsáhnout problematiku nemocí štítné žlázy. Jelikož jsou

ukládaná data součásti souboru databázové aplikace, využívají toho její uživatelé k práci s daty i mimo pracoviště kliniky, například při práci v zahraničí.

Během psaní této práce pokračuje vývoj nové verze modulu klinických vyšetření, který bude doplněn o možnost ukládání volitelného seznamu subjektivních obtíží pacienta i s parametry, které pomohou lékařům sledovat, zda-li se tyto obtíže zlepšují nebo zhoršují v závislosti na léčbě pacienta. Pro plné obsažení všech možností této aplikace v endokrinologii štítné žlázy bude v nejbližší době vyvíjen nový modul pro anamnézu pacienta. Funkcí tohoto modulu bude ukládání podrobných informací o celkové rodinné anamnéze, která může souviset s jeho léčbou nebo tyto informace budou zajímavé pro sledování ve výzkumných projektech.

Po dokončení vývoje endokrinologické části bude tato aplikace nabídnuta k použití pro ostatní lékaře interní kliniky, kteří budou mít zájem využít rozsáhle společné databáze pacientů a mít tak možnost velmi jednoduchým způsobem zpracovávat podklady pro své studie. Dle nových požadavků lékařů lze v aplikaci naprogramovat i další moduly, které by pokryly požadavky kardiologických pracovišť 3. interní kliniky s využitím společné obsáhlé databáze pacientů.

V databázové aplikaci byl odzkoušen export celé databáze na Microsoft SQL Server™ 2005. Aplikace je tak připravená pro nasazení v robustnějších řešeních, která by mohla vyžadovat zvýšené nároky na správu dat a uživatelů databáze. Nemusí přitom dojít ke zvýšení finančních nákladu na toto řešení, jelikož verze Express zmíněného databázového stroje je k dispozici zdarma pro nekomerční použití.

7. Závěr

Tato bakalářská práce ukázala, že s využitím běžných programových prostředků dostupných ve Všeobecné fakultní nemocnici v Praze lze vyvinout aplikaci, která poskytne lékařům široké možnosti v ukládání a následném zpracování informací o pacientech, které by jinak museli složitě vyhledávat v NIS Medea nebo si je ukládat do pomocných tabulek. Přínosem aplikace je také rychlost, s jakou lze v ní sestavit lékařskou zprávu, která se následně vkládá do nemocničního informačního systému.

Aplikace je tak určená i těm lékařům, kteří nepotřebují zpracovávat žádná data, ale chtějí svou práci s NIS Medea jen urychlit. Pomohou tím k rozšiřování velikosti databáze pacientů.

Velmi zajímavou vlastností popisované databázové aplikace je její univerzálnost. S využitím podobných postupů, aplikovaných při vývoji na 3. interní klinice, lze naprogramovat databázovou aplikaci i pro jinou kliniku, která může používat NIS Medea. Přínosem pro její uživatelé bude nejen možnost snadného přístupu k informacím o pacientech, ale i zvýšení rychlosti při zpracování lékařských zpráv z vyšetření pacientů.

Vývoj této databázové aplikace je podporován grantem Ministerstva zdravotnictví České republiky (IGA MZ NR/8130-3) a grantem Akademie věd České republiky (IET101050403).

8. Použité zkratky

NIS	nemocniční informační systém
LIS	laboratorní informační systém
SQL	Structured Query Language
GUI	Graphic User Interface
DB	Databáze
ODBC	Open DataBase Connectivity
VBA	Visual Basic for Applications
XML	Extensible Markup Language

9. Literatura

1. Gooding GA: Sonography of the thyroid and parathyroid. Radiologic clinics of North America. Saunders, 1993; 31(5): 967-989.

2. Hopkins CR, Reading CC. Thyroid and parathyroid imaging. Seminars in ultrasound, CT, and MR. W B Saunders, 1995; 16(4): 279-295

3. Loevner LA. Imaging of the thyroid gland. Seminars in ultrasound, CT, and MR. W B Saunders, 1996; 17(6): 539-562

4. Wartfsky L, Ingbar SH. Disease of the thyroid. In: Harrison's principles of internal medicice. 12th ed. New York: McGraw-Hill, 1991: 1712

5. Solbiati L, Osti V, Cova L, Tonolini M. Ultrasound of thyroid, parathyroid glands and neck lymph nodes. European Radiolog,y 2001; 11(12): 2411-2424

6. Smutek D., Šára R., Sucharda P., Tjahjadi T., Švec M.: Image Texture Analysis of Sonograms in Chronic Inflamations of Thyroid Gland. Ultrasound in medicine & biology. Pergamon Press, 2003; 29: 1531-1543

7. Jerke N: Microsoft Office Access 2003 Professional Results. McGraw-Hill Professional, 2003; 4-44.

8. Kimmel P.: Building Delphi 6 Applications. Osborne/McGraw-Hill, 2001; 421-425.

9. Freytag JC, Maier D, Bossem G: Query Processing for Advanced Database Systeme. Morgan Kaufmann, 1993; 441.

10. Wikipedia.org: Relational model. [cit. 2006-6-1].
URL: <http://en.wikipedia.org/wiki/Relational_model>

11. Wikipedia.org: Edgar F. Codd. [cit. 2006-6-1].
URL: < http://en.wikipedia.org/wiki/Edgar_F._Codd>

12. ISO/IEC 9075-3:1995: International Organisation for Standardization (ISO): Information technology - Database languages - SQL - Part 3: Call-Level Interface (SQL/CLI). ISO, 1995.

13. Císař P.: InterBase/Firebird tvorba, administrace a programování databází. Computer Press, 2003; 61

14. Dobson R.: Programování v Microsoft Access 2000. Computer Press, 2000; 2-41.

15. Jiskra J., Smutek D., Barkmanová J., Antošová M., Límanová Z., Potluková E., Sucharda P.: Serum Levels of Antibodies to Thyroid Peroxidase Correlate with Quantitative Descriptors of Thyroid Ultrasound Images in Patiens with Breast Cancer. Prague Medical Report. Universitas Carolina Pragensia, 2005; 106: 399-408

10. Přílohy

Elektronické přílohy jsou na CD-ROM přiloženém v této bakalářské práci a uvádím je v tabulce č. 2.

Seznam elektronických příloh	
Soubor	**Popis**
`BC-prace.pdf`	Elektronická podoba této bakalářské práce.
`Tyrex-BC.mdb`	Databázová aplikace v Microsoft Access 2003.
`Export1.txt`	Textový soubor s exportem mixéru NIS Medea.
`Export2.txt`	Textový soubor s exportem mixéru NIS Medea.

Tabulka č. 2 Seznam elektronických příloh