

Univerzita Karlova v Praze
1. lékařská fakulta

DIPLOMOVÁ PRÁCE

Bc. Zdeněk Telička

Využití databázových aplikací v klinickém výzkumu

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

Vedoucí diplomové práce: MUDr. Jan Jiskra, PhD.

Praha, květen 2008

Čestné prohlášení

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

V Praze dne 18. května 2008

Zdeněk Telička

Abstrakt

Diplomová práce pojednává o možnostech využití databázových aplikací v klinickém výzkumu vědeckých pracovníků 3. interní kliniky 1. lékařské fakulty Univerzity Karlovy a Všeobecné fakultní nemocnice v Praze. Pro výzkumné účely v současné době lékaři používají k zaznamenávání sledovaných informací získaných vyšetřeními pacientů (klinickým, laboratorním a zobrazovacími metodami) tabulkové procesory. Důvodem je absence vhodného programového prostředku, který by byl součástí nemocničního informačního systému a který by umožnil ukládání předem definovaných parametrů, které lékaři požadují sledovat. Nedostatkem tabulkových procesorů je nemožnost sdílet data mezi více lékaři, protože s každou studií si lékaři vytvářejí své vlastní tabulky a dochází tak k redundanci dat. V této práci se dále zabývám vývojem relační databázové aplikace typu klient-server a jejího uživatelského prostředí, která umožní nejen ukládání strukturovaných dat, ale zjednoduší i jejich další analýzu ve výzkumných projektech. Databáze rovněž umožňuje sdílet data mezi různými pracovišti kliniky a tím zvýšit efektivitu práce výzkumných týmů. Zajímavými vlastnostmi aplikace jsou její spolupráce s nemocničním informačním systémem při výměně dat (výsledků laboratorních vyšetření a osobních dat pacientů) a schopnost generovat textové zprávy ze strukturovaných dat uložených v databázi.

Abstract

The Diploma Thesis „Utilization of database applications in clinical research“ deals with development of client-server database application for 3rd Medical Department, 1st Faculty of Medicine, Charles University in Prague. Application's aim is to store structured data of patients' examinations, treatments and biochemical results. 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 patient identification data 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	7
2. Současné možnosti sběru a extrakce dat v klinickém výzkumu	8
2.1 Data v nemocničním informačním systému	8
2.2 Extrakce dat z medicínských textů	9
2.3 Strukturovaná data v tabulkových procesorech	11
2.4 Strukturovaná data ve specializovaných databázích	13
2.5 Strukturovaná data v nemocničních informačních systémech	14
3. Aplikace a technologie použité při vývoji	16
3.1 Microsoft Access	16
3.2 Microsoft SQL Server	18
3.3 Visual Basic for Applications	20
3.4 Windows API	20
3.5 StaproMedea	22
4. Analýza databázové aplikace	23
4.1 Vývoj první verze aplikace	23
4.2 Požadavky na rozšíření aplikace na další pracoviště kliniky	24
4.3 Požadavky na bezpečnost databázové aplikace	25
4.4 Požadavky na komunikaci aplikace s NIS StaproMedea	26
5. Vývoj serverové části aplikace	27
5.1 Definice tabulek a vztahů	27
5.2 Migrace databáze Access na SQL server	31
5.3 Autentizace a autorizace	33
5.4 Zálohování databáze	34
6. Vývoj klientské aplikace	35
6.1 Návrh grafického uživatelského prostředí	35

6.2	Programování ve Visual Basic for Applications, generování zpráv	38
6.3	Programování komunikace s NIS pomocí Windows API	41
7.	Závěr	44
7.1	Přínos a využití aplikace	44
7.2	Budoucí rozvoj aplikace	45
8.	Literatura	47
9.	Přílohy	49

1. Úvod

Tato diplomová 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 v roce 2005 začal spolupracovat při návrhu databázové aplikace, která by umožňovala ukládat parametrizované informace z vyšetření pacientů v endokrinologických ambulancích, výsledky biochemických vyšetření a data ze sonografických vyšetření štítné žlázy. Důvodem k vývoji databáze byla nemožnost ukládat a vyhledávat detailní data, která by byla následně snadno zpracovatelná ve statistických analýzách, apod. Lékaři ukládali nálezy z vyšetření do nemocničního informačního systému ve formě zpráv, které již neobsahovaly původní strukturované informace. Výsledky biochemických vyšetření se nacházely v samostatném modulu nemocničního informačního systému (NIS), který nedovoluje zobrazovat přehledové tabulky s výsledky více pacientů najednou. Docházelo tedy k tomu, že si lékaři vytvářeli samostatné tabulky v tabulkových procesorech, které obsahovaly podrobná data. Vyhledávání dat z několika nesourodých zdrojů pak bylo složité a časově náročné. Vyvinul jsem proto databázovou aplikaci na platformě Microsoft Access, která do jediného místa sdružovala data z několika zdrojů najednou. Významným způsobem přispěla ke sběru a analýze dat z ultrazvukových vyšetření štítné žlázy, které je dnes základní zobrazování metodou v tyreoidologické diagnostice [1; 2; 3].

V této práci se zabývám dalším rozvojem databáze. V první části popisují současné možnosti a programy pro sběr výzkumných dat, jejich výhody a nevýhody. V druhé uvádím technologie použité k vývoji vlastního řešení. V praktické části práce se věnuji analýze a programování. V průběhu práce na projektu jsme společně s lékaři kladli na aplikaci stále vyšší nároky, například po stránce robustnosti, možnosti výměny dat s nemocničním informačním systémem (NIS StaproMedea) a zabezpečení dat. Naprogramoval jsem funkce pro import identifikačních údajů pacientů a výsledků z NIS do aplikace. Využil jsem k tomu Windows API, což je sada nízkoúrovňových funkcí pro komunikaci mezi procesy v operačním systému. Do nemocničního

informačního systému jsou naopak exportovány klinické zprávy vygenerované ze strukturovaných dat uložených v databázi. Tato vylepšení způsobila velký zájem o rozšíření aplikace na ostatní odborná pracoviště 3. interní kliniky, a proto bylo nutné migrovat databázi z jednoduché souborové databáze na SQL server. Pro tyto účely jsem začal spolupracovat s úsekem informatiky Všeobecné fakultní nemocnice, kde je databáze provozována na Microsoft SQL Serveru.

2. Současné možnosti sběru a extrakce dat v klinickém výzkumu

2.1 Data v nemocničním informačním systému

Na klinikách se dnes využívá k ukládání lékařských dat nemocničních informačních systémů. Tyto systémy fungují z pohledu lékařů jako elektronická zdravotní karta pacienta umožňující zaznamenávat a sledovat průběh léčby pacienta. Lékaři využívají nejčastěji klinické moduly pro ambulance, lůžková oddělení a modul s výsledky biochemických vyšetření. V poslední době se v nemocničních informačních systémech implementují moduly pro ukládání a práci s obrazovými daty, tzv. PACS (Picture Archival and Communications System). Součástí pacientovy elektronické dokumentace se tak mohou stát např. snímky pořízené během RTG vyšetření nebo i videa nahraná Dopplerovskými ultrazvukovými přístroji.

Největším problémem ve využití nemocničních informačních systému (NIS) v klinickém výzkumu je nemožnost ukládat data ve strukturovaném formátu. Současné systémy standardně umožňují pouze zaznamenávání informací ve formě textových lékařských zpráv. Pokud chtějí lékaři pro účely svého výzkumu získat informace z předešlých lékařských vyšetření, musejí vyhledávat požadované informace v pacientově dokumentaci. V této chvíli se projevují dvě největší nevýhody těchto systémů. První je časová náročnost, protože lékaři musejí ručně vyhledávat data v jednotlivých lékařských zprávách, což je časově velmi náročné. NIS často ani nedovolují zobrazit v jediné tabulce data více pacientů. Druhou nevýhodou je, že

průběžné lékařské zprávy obsahují informace rozhodující pro stanovení diagnózy a léčbu, ale již obvykle neobsahují detailní data o anamnéze a léčbě, která jsou přínosná právě pro klinické výzkumy. Srovnání výhod a nevýhod využití NIS pro klinické výzkumy uvádím v tabulce č. 1.

Nemocniční informační systémy často nabízejí lékařům jen základní možnosti ve vyhledávání a filtrování pacientů. Je tak možné vyhledávat pacienty dle data vyšetření, čísla diagnózy, nebo dispenzáře, což je prostředek, který dovoluje lékařům třídít pacienty do různých skupin podle toho, jaké studie se pacienti účastní, apod. Z toho vyplývá, že lze vyhledat data jen z některých základních parametrů, nelze tedy vyhledávat rozsáhlé informace např. v lékařských zprávách. Zpřístupnění takové funkce by patrně vyžadovalo zvýšené nároky na softwarové i hardwarové nároky serveru, na kterém systém běží. NIS Stapro Medea, který je využíván ve Všeobecné fakultní nemocnici v Praze obsahuje modul Medea Query Results [4], který je postaven na aplikaci Progress® Query/Results™ [5]. Modul by měl umožnit tvorbu ad-hoc dotazů pro vyhledávání dat v databázi nemocničního informačního systému, lékaři však o něm nemají žádné informace. Tento modul je navíc určen jen pro odborné IT uživatele po zaškolení. Tímto je znemožněno jeho snadné využití v každodenní praxi.

Data v nemocničním informačním systému	
Výhody	Nevýhody
Není nutné vyvíjet doplňkové aplikace.	Neobsahuje všechna data důležitá pro výzkum.
Nedochází k duplicitám dat ve více databázích.	Nedovoluje ukládání strukturovaných dat.
	Nemožnost komplexního vyhledávání v databázi NIS.

Tabulka č. 1: Srovnání výhod a nevýhod využití dat v nemocničním informačním systému

2.2 Extrakce dat z medicínských textů

Jednou z možností, jak zajistit správnost a úplnost dat v klinických zprávách z vyšetření pacienta, je psaní lékařských zpráv dle textů pro formalizaci lékařských

doporučení. „Lékařské doporučené postupy (lékařská doporučení, guidelines) jsou odborné medicínské texty zabývající se problematikou postupu diagnostiky, terapie či prevence jednotlivých onemocnění či jejich skupin. Mohou být využita ke standardizaci léčby daného onemocnění, zlepšení péče o pacienty zavedením nových postupů, porovnávání kvality péče na jednotlivých zdravotnických pracovištích, k výuce zdravotnických pracovníků, při vývoji aplikací pro automatizaci některých lékařských úkonů, tvorbu systémů pro podporu rozhodování, pro zlepšení mezioborové spolupráce v medicíně atd. [6]“ Lékařské zprávy psané podle těchto doporučení mohou být následně využity pro zpracování v aplikacích s technologií na extrakci informací z medicínských textů [7]. Tyto algoritmy jsou schopné vyhledat požadovaná data v semi-strukturovaném lékařském záznamu a extrahovat z něj strukturované informace, tzn. vyhledávají v textu tři typy informací – numerické hodnoty, lékařské výrazy a kategorické hodnoty. Výsledkem je tabulka obsahující parametry a jejich hodnoty. Taková data již lze využívat pro další statistické analýzy, apod. Srovnání přínosu extrakce dat z medicínských textů uvádím v tabulce č. 2.

Aplikace na extrakci medicínských textů nejsou běžnou součástí nemocničních informačních systémů a myslím, že by to byla zajímavá nadstavba, která by na klinikách našla rozmanité využití. Databáze klinických modulů NIS obsahují gigabyty informací, které jsou sice využívány k diagnostice a léčbě pacienta, ale po výzkumné stránce se s těmito daty již příliš nepracuje a kliničtí výzkumníci raději volí ukládání strukturovaných dat do separátních tabulek nebo databází vytvořených vlastní silou.

Extrakce dat z medicínských textů	
Výhody	Nevýhody
Využití již nasbíraných rozsáhlých dat.	Nejednotnost jazyka a formátu zpráv.
Nedochází k duplicitám v datech.	Zvýšené nároky na HW.
	Nutnost vývoje specializované aplikace.

Tabulka č. 2: Srovnání výhod a nevýhod extrakce dat z medicínských textů

2.3 Strukturovaná data v tabulkových procesorech

Nejběžnějším způsobem, jakým lékaři řeší nedostatky NIS v oblasti ukládání strukturovaných informací, je používání tabulkových procesorů. Důvodem k tvorbě jednoduchých tabulek jsou nízké pořizovací náklady na software, minimální čas strávený návrhem a jednoduchost používání. Tabulkové procesory výzkumníci hojně využívají pro sběr dat jak ve svých ambulancích již během vyšetření pacienta, tak i při retrospektivním vyhledávání informací v informačním systému. Nejčastěji se takto vyhledávají výsledky biochemických vyšetření, které jsou uloženy v laboratorním modulu informačního systému. Dalšími vyhledávanými informacemi jsou také zprávy a diagnózy z návazných vyšetření, například vyšetření zobrazovacími metodami. Smyslem vytváření vlastních tabulek je centralizace dat pro snadnou statistickou analýzu v klinických výzkumech. Nejjednodušším způsobem, jak zpracovat data v současných komerčních statistických programech je právě pomocí obyčejných tabulek.

Současné statistické programy již nabízejí přímou konektivitu k databázím pomocí univerzálních ovladačů pro přístup k databázím nejrůznějších výrobců [8]. Funkcí těchto ovladačů je poskytnout aplikacím či vývojářům jednotnou sadu knihoven pro přístup k databázovým zdrojům bez jeho konkrétní znalosti. Ovladač je vlastně prostředníkem, který převádí volání funkcí mezi aplikací a konkrétním databázovým strojem pomocí tzv. konektoru. Ten zabezpečuje převod příkazů aplikace na příkazy vyžadované databází konkrétního výrobce. Programátor tak má jednodušší práci při vývoji aplikace, která se stává méně závislou na konkrétní databázové platformě.

Starší, ale dodnes používanou technologií pro univerzální přístup k databázovým zdrojům je na platformě Windows ODBC (Open DataBase Connectivity). Přestože je tento systém vytvořen firmou Microsoft, využívá se i v operačních systémech Unix (Linux), OS/2, Mac OS a dalších. Velkou nevýhodou ODBC je jeho nízká rychlost i při zpracovávání menších objemů dat. Microsoft dnes ve svých operačních systémech dodává modernější technologii OLE DB [9]. Zajímavostí těchto řešení je, že jako databázový zdroj může být použita i tabulka vytvořená v aplikaci Microsoft Excel, nad kterou se lze dotazovat pomocí jazyka SQL (Structured Query Language). To je

v současné době běžně využívaný strukturovaný dotazovací jazyk pro řízení dat v relačních databázových systémech. Při instalaci sady Microsoft Office se automaticky instaluje i konektor, který zprostředkuje komunikaci s excelovskými soubory.

Pokud se ve statistickém programu využije konektivita k databázi, program pak k řízení dat dokáže využít SQL jazyk. Všichni výrobci se snaží zachovat kompatibilitu se standardem SQL-92 [15]. Tím je zajištěna kompatibilita nejběžněji využívaných příkazů jazyka. Výrobci pak svůj databázový systém doplňují o nadstavby SQL jazyka, které přidávají možnosti programování na straně SQL serveru. Např. Microsoft SQL server implementoval T-SQL (Transact SQL) a Oracle PL/SQL (Procedural Language/SQL). Tyto jazyky již ale mezi sebou kompatibilní nejsou, proto se robustní databázové aplikace vyvíjejí často s využitím konkrétní databázové platformy [10; 11].

Ve Všeobecné fakultní nemocnici jsem se doposud v rutinní praxi neseťkal s využitím těchto technologií při sběru nebo analýze dat mezi lékaři. Brání tomu zřejmě nejen složitost a rozsáhlost databází v informačních systémech (IS), ale i neochota správců zpřístupnit byť i jen čtecí přístup k některým tabulkám těchto systémů. Díky tomu lékaři tak zůstávají u kopírování dat z informačních systémů do svých tabulek a následným importem dat do statistického programu. Tento způsob sebou nese několik nesporných nevýhod. S tabulkou nelze pracovat na multiuživatelské úrovni, jelikož v jeden čas dovolí aplikace otevření souboru pro zápis jen jednomu uživateli. Dalším problémem je bezpečnost dat. Soubory tabulkových procesorů neumožňují žádné kvalitní zabezpečení přístupu k datům. Nelze se spoléhat ani na zakomponované funkce pro zaheslování souboru, apod. Případný útočník, který by soubor získal, může s dostatkem času data v souboru dešifrovat. Nebezpečím může být také nezabezpečený přístup k počítači, na kterém se soubory nacházejí. Zároveň, tabulky v těchto aplikacích jsou využitelné jen pro data malého rozsahu, a proto na vytvoření rozsáhlejšího souboru dat jsou vhodnější jednoduché databázové aplikace, které jsou například součástí kancelářských balíků. V tabulce č. 3 uvádím přehled výhod a nevýhod sběru dat v tabulkových procesorech.

Sběr dat v tabulkových procesorech	
Výhody	Nevýhody
Jednoduchost a rychlost návrhu tabulek	Nejednotnost jazyka a formátu v prohledávaných zprávách
Snadné používání aplikace.	Vhodné jen pro jednoduchou strukturu tabulek a nízké objemy dat.
Nízké finanční nároky.	Jednouživatelský přístup.
	Nedostatečné zabezpečení.

Tabulka č. 3: Srovnání výhod a nevýhod sběru dat v tabulkových kalkulátorech

2.4 Strukturovaná data ve specializovaných databázích

Při zvýšených požadavcích na složitost a provázanost tabulek pro sběr výzkumných dat je možné využít databázových vývojových nástrojů. V 80. a 90. letech byl velmi oblíbený nástroj FoxPro, který byl po prodeji firmě Microsoft přejmenován na Visual FoxPro. Nástroj integroval v jednom balíku vývojové prostředí, programovací jazyk, databázi a také uživatelské prostředí. Databáze byla ve formě tabulek uložených v separátních binárních souborech [12].

Pro tvorbu malých databázových aplikací je dnes v prostředí operačního systému Microsoft Windows využívána často aplikace Access společnosti Microsoft, která se stala součástí kancelářského balíku Microsoft Office Suite. Aplikace slouží nejen k vývoji databáze a uživatelského prostředí, ale i ke spuštění databázové aplikace. Přitom databáze i uživatelské prostředí jsou ukládány do jediného souboru [13]. K programování uživatelských funkcí je v aplikaci implementován programovací jazyk Visual Basic for Applications, který je derivátem velmi oblíbeného Visual Basicu [14]. Výhodou využití tohoto nástroje je jeho nenáročnost na zkušenosti uživatele. Vlastní databázovou aplikaci si může vytvořit i uživatel, který nemá o databázových systémech hluboké znalosti. Naopak, pokud má vývojář velké zkušenosti s programováním a návrhem databází, může v Accessu vytvořit i databáze typu klient-server. Microsoft Access podporuje připojení k externím databázovým zdrojům a je v něm implementována přímá podpora pro Microsoft SQL Server, čímž lze databázi částečně

ovládat přímo z prostředí Microsoft Access. Tato funkce je v Accessu pojmenována Access Project [15]. Přehled výhod a nevýhod využití databázových aplikací uvádím v tabulce č. 4. Cestou použití Microsoft Access v kombinaci s Microsoft SQL Server jsme se vydali i na 3. interní klinice. Vývojem a využitím databázové aplikace pojednávám od třetí kapitoly této diplomové práce.

Sběr dat ve specializovaných databázích	
Výhody	Nevýhody
Možnost propojení s NIS.	Vyšší časové a znalostní nároky na vývoj aplikace.
Snadné používání aplikace.	Duplicita některých dat s NIS.
Nízké finanční náklady na vývoj.	
Plná kontrola nad výzkumnými daty.	

Tabulka č. 4: Srovnání výhod a nevýhod sběru dat ve specializovaných databázích

2.5 Strukturovaná data v nemocničních informačních systémech

Kvůli výše zmíněným nevýhodám se vyvíjí různé způsoby, jak integrovat sběr strukturovaných dat přímo do nemocničních informačních systémů. Jednou z možností je vytvoření doplňkových sub-modulů, jež by byly součástí klinických modulů nemocničních informačních systémů. Submoduly pak mohou splňovat konkrétní požadavky klinického pracoviště na strukturu informací. Na kardiologickém oddělení 3. interní kliniky Všeobecné fakultní nemocnice v Praze se dříve rozvíjela spolupráce se společností Stapro, která implementuje a spravuje NIS StaproMedea ve VFN. Myšlenka byla taková, že součástí informačního systému budou také formuláře na ukládání výzkumných údajů o pacientech, které by byly téměř nezávislé na klinické části informačního systému, kterou lékaři využívají v ambulancích a na lůžkovém oddělení. Klinická a výzkumná část informačního systému by však byla propojena využitím společné tabulky s identifikačními údaji pacientů, čímž by nedocházelo k vedení duplicitních údajů o pacientech. Problémem výzkumných submodulů je to, že jsou

v rozpracovaném stavu, jejich používání je neergonomické a příliš časově náročné. V současné době práce na výzkumných modulech dále nepokračují.

Společnost Stapro se v současné době snaží o vývoj obecného řešení, které by pomohlo uživatelům s implementací specifických požadavků do již fungujícího informačního systému. Jedná se o flexibilní formulář. Prvním produktem vytvořeným tímto nástrojem je produkt s názvem „Řešení pro onkologii“. Jde o produkt, který obsahuje funkcionality, které umožňují zaznamenávat strukturovaně klinické údaje o onkologickém onemocnění a tyto údaje vyhodnocovat. Měl by pomoci onkologům v řízení kvality onkologické péče možnostmi zaznamenávat a následně vyhodnocovat data ve statistických výstupech [16]. Flexibilní formulář není přímou součástí nemocničního informačního systému StaproMedea, ale je jen jeho nadstavbou. To znamená, že aplikace vyžaduje vlastní SQL server, který je postaven na technologii Microsoft SQL server a uživatelské prostředí je vyvíjeno v Microsoft .NET frameworku 3.0. NIS Medea je vyvíjen na technologii společnosti Progress Software, tzn., že využívá databázový server Progress DB. Formuláře vytvořené pomocí nástroje „flexibilní formulář“ je možno provozovat pouze po připojení k nemocničnímu informačnímu systému – je úzce navázáno na NIS, neřeší správu uživatelů, ani přístupová práva. Flexibilní formuláře se volají z NIS, umožňují z něj načíst některé údaje (základní údaje o pacientovi, diagnózy, atp.) a zpět posílají zformátovaný RTF text závěru ze strukturovaných dat. Součástí aplikace je tisk a tvorba elektronických výstupů. Další vlastností je možnost konektivity k libovolnému NIS. Produkty, které se vytvoří ve flexibilním formuláři lze následně provozovat jak s NIS StaproMedea, tak i StaproAKORD. Možné je také připojit flexibilní formulář i na libovolný cizí NIS [16]. Nevýhodou toho komerčního řešení jsou další finanční náklady. Zákazník si musí zakoupit nemocniční informační systém a k tomu pak ještě i flexibilní formulář. Přehled výhod a nevýhod sběru strukturovaných dat v NIS uvádím v tabulce č. 5.

Strukturovaná data v nemocničních informačních systémech	
Výhody	Nevýhody
Nižší duplicita dat.	Vysoké časové nároky na vývoj aplikace.
Jedna aplikace pro více činností lékařů.	Výzkumníci nemají plnou kontrolu nad svými daty.
	Vysoké finanční nároky.
	Závislost na technologii dodavatelské firmy.

Tabulka č. 5: Srovnání výhod a nevýhod sběru strukturovaných dat v nemocničních informačních systémech

3. Aplikace a technologie použité při vývoji

3.1 Microsoft Access

K vývoji databázové aplikace byl zvolen Microsoft® Access 2007. Jako klientská aplikace je použit Access verze 2003, jelikož je součástí verze Microsoft® Office, která je dostupná na všech pracovištích kliniky, a také v celé fakultní nemocnici. Tímto byly eliminovány dodatečné finanční náklady na pořízení jiného vývojového nástroje, například CodeGear® 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. Aplikace vytvořené v Accessu vyžadují ke spuštění opět Access, nebo lze použít Microsoft Access Runtime 2007, který je distribuován zdarma. Tímto se stává Access lákavější platformou pro tvorbu aplikací v prostředí Windows, protože uživatelé nejsou nuceni kupovat sadu Office. Access má v sobě implementovány následující funkce:

- Návrh databázových tabulek a vztahů mezi nimi. Lze využít jak grafické prostředí, tak i knihovny v případě, že Access je zvolen pouze jako databázový stroj a uživatelské je vyvíjeno v odlišném nástroji.

- Návrh SQL dotazů pro různé uživatelské pohledy na uložená data nebo pro manipulaci s daty. Access je kompatibilní s SQL-92 standardem, který zajišťuje kompatibilitu mezi databázovými platformami různých výrobců.
- Tvorbu grafického uživatelského prostředí (GUI) pomocí formulářů umožňující snadnou uživatelskou práci v databázové aplikaci. Vývoj je podobný jako ve Visual Basicu nebo v Delphi.
- Programovací jazyk Visual Basic[®] for Applications dovolující vývojáři naprogramování automatizovaných funkcí nebo interakcí mezi uživatelem a databázovou aplikací, případně databázovou aplikací a operačním systémem. Tento jazyk je implementován ve všech aplikacích sady Office, čehož lze využít k naprogramování interakce mezi aplikacemi, například při přesunu dat z Accessu do Excelu, Outlooku, apod.
- Tvorba sestav pro tiskové výstupy. Sestavy se vytvářejí podobně jako formuláře a je možné je doplňovat kódem VBA.
- Podpůrné nástroje pro rozdělení databáze na dva soubory. První soubor je zdrojem dat a druhý má v sobě uloženo grafické uživatelské prostředí (formuláře) společně s makry a moduly. Tímto se vysoce zvyšuje produktivita práce, vývojář může pracovat na nové verzi uživatelského prostředí, zatímco uživatelé pracují s původní verzí aplikace. Při aktualizaci aplikace stačí jen nahradit soubor s uživatelským prostředím.
- Zálohování, případně replikování databáze pro možnost automatického udržování několika shodných kopií databáze.
- Podpora pro XML (eXtensible Markup Language), což je moderní formát pro dokumenty. Lze toho využít například pro import a export dat třetích stran.
- Migraci databáze na SQL server. Databázi lze vytvořit nebo převést na Microsoft[®] SQL Server nebo se lze připojit k existující databázi databázového stroje. V tomto případě bude soubor databáze sloužit pouze jako uživatelské rozhraní. Access pro

svou databází také zprostředkovává univerzální ovladače a tím je možné zvolit opačný postup: připojit se například z aplikace naprogramované v Delphi k databázi vytvořené v Accessu.

- Možnost vytvoření tzv. MDE, případně ADE souboru, který je určen pouze pro uživatelskou práci s databází. V tomto souboru není možné měnit strukturu databáze nebo formulářů a obsahuje pouze zkompilovaný kód Visual Basicu. Databázová aplikace ke svému spuštění bude nadále vyžadovat nainstalovaný Microsoft Access.

3.2 Microsoft SQL Server

Jako databázovou platformu jsem zvolil Microsoft SQL Server 2005. Vedly mě k tomu dva podstatné důvody. První je ten, že tento SQL server má přímou podporu v Microsoft Access pomocí funkce Access Project. V prostředí Accessu tak lze vzdálenou databázi na SQL serveru nejen spravovat, ale i vyvíjet. Samozřejmě, Access nepodporuje všechny funkce Microsoft SQL serveru, databázi lze proto spravovat i v Microsoft SQL Server Management Studiu, což je grafické uživatelské prostředí pro administrátory a vývojáře SQL databází. Druhým důvodem pro volbu MS SQL serveru byla spolupráce s úsekem informatiky Všeobecné fakultní nemocnice v Praze. Nemocnice používá pro běh svých databází technologie společnosti Microsoft. Databáze je tak umístěna na serveru, který spravuje úsek informatiky. Tímto má naše aplikace zajištěnou stejnou podporu, jako jiné aplikace provozované pro management fakultní nemocnice.

SQL Server 2005 představuje integrované řešení pro správu a analýzu dat. Je nabízen v několika variantách, které se liší nejen cenou, ale předpokládaným využitím. Vzhledem ke konkurenci, jaká v současné době panuje na poli databázových serverů, také Microsoft nabízí verzi Express, která je distribuována zdarma. Tuto verzi využívám na svém počítači k vývoji. Fakultní nemocnice má nasazenu profesionální verzi. Omezení verze zdarma jsou hlavně v maximální velikosti databáze (4 GB) a v maximálně využitelné paměti RAM (1 GB). V této verzi nelze vytvářet repliky, ale lze se k replikacím připojit. Replikace slouží k distribuování databáze na více strojů

zároveň, přičemž všechny databáze jsou navzájem synchronizovány. MS SQL serveru nabízí komplexní nástroje pro vývoj a správu databází:

- Vysoce dostupný relační databázový stroj se zvýšeným výkonem a podporou strukturovaných a nestruturovaných dat (XML).
- Služba Replication Services. Replikace dat pro aplikace zpracovávající distribuovaná či mobilní data, vysoká dostupnost systémů, škálovatelný souběh se sekundárními úložišti dat pro řešení vytváření podnikových sestav a integrace s heterogenními systémy, včetně stávajících databází Oracle.
- Služba Notification Services. Pokročilé funkce zasílání upozornění pro vývoj a nasazení škálovatelných aplikací, které mohou na nejrůznější připojená i mobilní zařízení zasílat individuálním požadavkům přizpůsobené aktuální informace.
- Služba Integration Services. Funkce extrakce, transformace a načítání dat (ETL) pro datové sklady a integraci dat v celém podniku.
- Služba Analysis Services. Funkce OLAP (Online Analytical Processing) pro rychlou a pokročilou analýzu velkých a složitých datových sad s využitím vícedimenzionálních úložišť.
- Služba Reporting Services. Komplexní řešení pro vytváření, správu a zasílání tradičních papírových i interaktivních webových sestav.
- Nástroje pro vývojáře. SQL Server nabízí integrované nástroje pro vývojáře určené pro databázový stroj, extrakci, transformaci a načítání dat (ETL), dolování dat, funkce OLAP a vytváření sestav, které jsou úzce integrovány se sadou Microsoft Visual Studio a poskytují komplexní funkce pro vývoj aplikací. Každý hlavní subsystém serveru SQL Server je dodáván s vlastním objektovým modelem a sadou aplikačních programovacích rozhraní (API), která rozšiřují datový systém libovolným směrem podle jedinečných požadavků zákazníků [22].

3.3 Visual Basic for Applications

Visual Basic for Applications (VBA) je událostmi řízený 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.

V této aplikaci jsem VBA používal ve formulářích k naprogramování funkcí, které v sobě Access nemá přímo implementovány nebo byly použity k výměně informací s NIS StaproMedea. 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. Součástí editoru jazyka VBA jsou prostředky k ladění aplikace, tzv. debugger.

VBA jsem také využíval k programovému přístupu k datům. Pomocí sady databázových knihoven lze direktně přistupovat k databázím a manipulovat s daty bez použití ovládacích prvků Accessu. Tyto funkce jsem využil například k automatizovanému generování klinických zpráv ze strukturovaných dat, anebo při přesunu dat mezi moduly aplikace.

3.4 Windows API

Windows API, Application Programming Interface, je sada nízkoúrovňových funkcí, které jsou součástí jádra operačního systému Windows. Slouží pro programování komunikace mezi objekty a procesy v systému. Windows API je naprogramováno v jazyku C++ a tvoří ucelený soubor DLL (Dynamic Link Library) knihoven. Ve Visual Basic for Applications je možné volat funkce v těchto knihovnách a tímto získat přístup k nízkoúrovňovým funkcím operačního systému [19]. Funkčnost Windows API lze rozdělit do 7 kategorií:

- Základní služby. Poskytuje přístup k nezbytným zdrojům poskytnutým systémem Windows. Zahrnuje souborový systém, periferie, procesy a vlákna, přístup do registrů Windows a ošetření chyb. Tyto funkce jsou uloženy v knihovnách kernel32.dll a advapi32.dll.
- Grafické uživatelské rozhraní (GUI). Poskytuje funkce pro výstup grafického obsahu na monitory, tiskárny a jiná výstupní zařízení. Knihovna je gdi32.dll.
- Uživatelské Rozhraní. Poskytuje funkce pro tvorbu a řízení počítačových oken a dalších základních prvků jako jsou tlačítka a posuvníky, zpracovává vstup z klávesnice a myši a jiných funkcí spojených s GUI. Tato funkční jednotka nachází od verze Windows XP v comctl32.dll, společně s běžnými prvky (Common Control Library).
- Knihovna běžných dialogových oken. Poskytuje aplikacím standardní dialogová okna pro otevření a ukládání souborů, volbu barvy a fontů, apod. Knihovna v souboru comdlg32.dll. Celkově také patří do kategorie Uživatelské Rozhraní.
- Knihovna běžných prvků (Common Control Library). Poskytuje aplikaci přístup k pokročilejším prvkům operačního systému. Zahrnuje stavový řádek, zobrazení průběhu výpočtu, toolbary a záložky. Knihovna je v souboru comctl32.dll. Celkově také patří do kategorie Uživatelské Rozhraní.
- Windows Shell. Umožňuje aplikacím přístup k funkcím poskytovaných shellem Windows. Komponenta je v souboru shlwapi.dll. Celkově také patří do kategorie Uživatelské Rozhraní.
- Síťové služby. Poskytuje přístup k různým počítačovým sítím. Zahrnuje také NetBIOS, Winsock, NetDDE, RPC a mnoho dalších funkcí.

V aplikaci jsem použil volání funkcí, které slouží zejména k načítání a vkládání dat mezi grafickými objekty, které se nacházejí v oknech aplikací. Díky Windows API jsem tak mohl naprogramovat funkce pro komunikaci aplikace s nemocničním informačním systémem.

3.5 StaproMedea

Na klinice je elektronická dokumentace pacienta vedena v NIS StaproMedea. Je to klinický informační systém pokrývající provoz klinických pracovišť řadou navzájem propojených produktů umožňujících vedení zdravotní dokumentace a podporujících provozní činnosti na jednotlivých klinických pracovištích. Systém zajišťuje zadání potřebných administrativních údajů, pořizování výkaznických a statistických dat a také podporuje činnost lékařů a sester při dokumentaci zdravotního stavu pacienta. Součástí řešení klinického informačního systému je patientská administrativa, která v sobě zahrnuje bloky funkcí vytvářející prostředí pro efektivní počítačové zpracování základních administrativních činností, jež jsou spojené s vedením patientské dokumentace. Tyto činnosti tvoří základní kostru informačního systému a jejich funkce jsou důležitým předpokladem pro navazující klinické procesy [21]. NIS z pohledu lékařů přináší následující možnosti:

- Přehlednost dokumentace. Všechny záznamy pacienta je možno vidět přehledně na jednom místě.
- Bezpečnost dat. U záznamů je evidováno, kým a kdy byly pořizeny, resp. editovány, současně lze nastavit i sledování kdo a kdy nahlížel do dokumentace pacienta.
- Omezení chybovosti. Systém vylučuje duplicitní zadávání údajů, všechny údaje se zadávají jen jednou a přenášejí se do všech míst, kde je třeba s nimi pracovat.
- Výstupy. K dispozici je mohutný nástroj pro tvorbu statistických výstupů z dat zadávaných do systému dle požadavků zákazníka.
- Centralizace dat. Centrální registr je základním nástrojem pro eliminaci chybovosti předávání identifikačních údajů pacientů, např. při požadavcích na komplementární péči, překladech pacienta mezi jednotlivými organizačními jednotkami nebo při vykazování péče plátcům

- Integrace celého NIS. Klinická část je integrována do modulu výkaznictví a činnosti spojené s pořizováním dat je možno optimalizovat nebo přímo zařadit do základního workflow vedení klinické dokumentace [21].

4. Analýza databázové aplikace

4.1 Vývoj první verze aplikace

První verze databázové aplikace jsem na 3. interní klinice začal vyvíjet již před třemi lety [17]. Během jejího vývoje byl kladen největší důraz na snadnost ovládní a rychlost vkládání dat do databáze. Práce s aplikací se nesměla stát další zdržující činností, neboť lékaři by ztratili motivaci k jejímu používání, což by bránilo v dalším rozšíření na ostatní pracoviště kliniky. Snažil jsem se při vývoji uživatelského rozhraní postupovat tak, aby zadávání jednotlivých údajů do databáze bylo rychlejší, než psaní textu do nemocničního informačního systému. Aplikaci jsem se společně s lékaři snažil navrhnout do takové podoby, která zaručuje intuitivní ovládní bez větších nároků na její studování. Aby neobtěžovala uživatele zbytečnými chybovými dialogy a nenutila je dělat činnosti, které vůbec nepožadují, mnoho údajů není povinné do databáze zadávat. Aplikace musela obsahovat jednotnou databázi pacientů společnou pro všechna vyšetření. S tím souvisí i další požadavek na možnost rychlého vyhledávání pacientů, kteří již nějaký záznam v databázi mají uložený.

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 určitou mez, která by zbytečně zvýšila nároky na jejich čas. Mohlo by pak dojít ke snížení přehlednosti aplikace a zvýšení náročnosti na ovládní. 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 lékařů, neboť vybírají jen jednu

z předdefinovaných hodnot a ručně vepisují údaje pouze v případech, pokud chtějí zprávu ukládanou do NIS 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 také její modulárnost. Aplikace je dělena na jednotlivé moduly, které odpovídají pracovištím kliniky nebo různým typům vyšetření pacienta. Aplikaci lze pak snadno doplňovat o další funkcionality nebo vkládat další moduly při rozšíření na nové pracoviště.

4.2 Požadavky na rozšíření aplikace na další pracoviště kliniky

Vývoj databázové aplikace byl původně soustředěn na odborné pracoviště zabývající se nemocemi štítné žlázy. Během jejího používání rostly požadavky na její modularitu, protože jsme zjišťovali, že databáze je významným pomocníkem pro ukládání rozličných údajů o léčbě pacienta, které by mohly být zajímavé v budoucích výzkumech. Během mého působení na klinice jsem postupně zjistil, že většina odborných pracovišť se také snaží o vývoj nějaké formy databázové aplikace. A pokud jsem se setkal s pracovníky oddělení, kde žádnou aplikaci pro sběr výzkumných dat nemají, zajímají se o možnosti rozšíření vyvíjené aplikace na jejich pracoviště.

Obezitologické oddělení využívalo jednoduchou databázovou aplikaci na internetu s využitím OpenSource technologií, což znamená, že programy jsou distribuovány zároveň se zdrojovými kódy, které lze za dodržení licenčních podmínek dále měnit a šířit. Programovacím jazykem byl PHP a databázová platforma server MySQL. Aplikace byla zabezpečena autentizací a autorizací. Přesto je umístění takovýchto databází na internetu nebezpečné, proto do databáze nebyly ukládány podrobné iniciály pacienta, rodné číslo, apod. Přestože aplikace byla velmi flexibilní s příjemným grafickým uživatelským prostředím, nevýhodou byla hlavně pomalost, pokud ji lékaři používali během vyšetření pacienta. Internetová aplikace má často

pomalejší reakční dobu, než klasické desktopové aplikace [18]. Druhým nedostatkem byla vyšší časová náročnost při jejím vývoji a v poslední řadě i nemožnost automatizované výměny dat s NIS. Rozhodli jsme se proto integrovat obezitologickou i endokrinologickou databázi do jednoho celku. Návrhem databáze pro tato pracoviště se zabývám v kapitole 5.1 Definice tabulek a vztahů.

Dalším požadavkem bylo rozšíření aplikace na diabetologické oddělení, které s obezitologií využívá často společná data. Diabetologické pracoviště v té době žádnou aplikaci pro výzkumné účely nepoužívalo. Zároveň mě kontaktovali pracovníci z kardiologického pracoviště s dotazy na databázi, kterou vyvíjíme. Zjistil jsem, že na kardiologii původně začali spolupracovat se společností Stapro na vývoji submodulů v NIS StaproMedea, které by sloužily k ukládání výzkumných informací. Vývoj byl ale z nezájmu společnosti Stapro přerušen. Proto si na tomto oddělení vyvinuli vlastní databázové aplikace pro konkrétní výzkumné činnosti, které se na oddělení provádí. Vzhledem k tomu, že na oddělení nemají k dispozici vývojáře, který má zkušenosti s programováním, databáze poskytuje jen základní možnosti ve sběru dat a její používání není úplně ideální, hlavně co se týče možnosti výměny dat s nemocničním informačním systémem. V době psaní této práce je rozšíření databáze na diabetologii a kardiologii ve fázi příprav.

4.3 Požadavky na bezpečnost databázové aplikace

Původní aplikace, kterou jsem na 3. interní klinice vyvíjel, neobsahovala skoro žádná bezpečnostní opatření. Aplikace byla zabezpečena jen tím, že se nacházela v adresáři na intranetu, který je přístupný až po autentizaci. Rozhodnul jsem se proto využít toho, že Microsoft Access má implementovanou podporu pro aplikace typu klient-server. Databázi je možné vytvořit nebo migrovat na databázový SQL server a soubor Accessu již pak slouží jen jako grafické uživatelské prostředí. Autentizace (ověření identity uživatele) a autorizace (ověření přístupových práv uživatele) je prováděna na straně databázového serveru.

Největším důvodem k migraci databáze byla právě bezpečnost dat. Chtěl jsem celý projekt dovést na profesionální úroveň, a proto jsem poprosil o spolupráci

zaměstnance úseku informatiky fakultní nemocnice. Původně jsem chtěl databázi provozovat na vlastním serveru dodaném partnerskou firmou a umístit jej na 3. interní klinice. Zaměstnancům úseku informatiky se však projekt zalíbil a databáze byla zprovozněna přímo na serverech fakultní nemocnice. To bylo pro mě výhodné také z toho hlediska, že jsem se mohl soustředit na vlastní programování a správu databáze a její bezpečnost jsem mohl přenechat na spolupracovnících.

4.4 Požadavky na komunikaci aplikace s nemocničním informačním systémem

Vzhledem k tomu, že lékaři nejsou nakloněni k používání dalších separátní aplikaci, hledal jsem způsob, jak automatizovat některé úkony při výměně dat s nemocničním informačním systémem. První funkce, kterou jsem do aplikace implementoval, bylo automatické generování zpráv ze zadaných strukturovaných údajů. Tato funkce urychlila práci s aplikací na téměř stejně krátký čas, jako kdyby lékaři používali ke své práci jen NIS, a přitom patří k nejkomplicovanějším částem aplikace. Snažili jsme vše vymyslet tak, aby u vygenerované zprávy nebylo poznat, že ji vytvořil počítač. Tímto se aplikace velmi odlišuje od jiných jednodušších systému v tom, že věty v textové zprávě jsou přesně přizpůsobeny tomu, co lékař zadal. Jiné systémy často jen vygenerují seznam parametrů s hodnotami.

Druhou možností programu je schopnost importu výsledků biochemických vyšetření z nemocničního informačního systému do databáze. Využili jsme mixéru, který je součástí NIS. Mixér umožňuje lékařům vygenerovat do textové podoby výsledky biochemických vyšetření. Ty jsou následně zpracovány algoritmy pro extrakci dat, které jsem naprogramoval ve VBA. Poslední zajímavou vlastností aplikace je import identifikačních dat pacienta. Pomocí knihoven Windows API [19] jsem naprogramoval funkce, které nám umožňují načíst do databáze identifikační údaje pacienta, který je právě zobrazen v NIS StaproMedea. Lékaři tak nikdy duplicitně nezadávají iniciály pacienta dvakrát, ale jen jednou v NIS. V aplikaci je pak po stisku tlačítka zobrazen pacient, který je zobrazen v NIS StaproMedea, anebo je automaticky

vytvořen nový záznam v databázi. V tabulce č. 6 uvádím přehled všech požadavků na vyvíjenou aplikaci.

Souhrn požadavků na aplikaci typu klient-server	
1.	Společná tabulka pacientů pro všechna odborná pracoviště.
2.	Nezávislost modulů pro jednotlivá pracoviště.
3.	Zabezpečený přístup k databázi.
4.	Spolupráce aplikace s nemocničním informačním systémem při výměně dat.
5.	Jednoduchost a přehlednost uživatelského prostředí.
6.	Nízké časové nároky na používání aplikace během vyšetření pacienta.
7.	Modulárnost a rozšiřitelnost aplikace na další odborná pracoviště kliniky.

Tabulka č. 6: Přehled požadavků na aplikaci typu klient-server

5. Vývoj serverové části aplikace

5.1 Definice tabulek a vztahů

Databáze se skládá z hlavní tabulky pacientů, která obsahuje iniciály pacienta a další základní údaje. Jako primární klíč je použit celočíselný datový typ s automatickou inkrementací. Jedinečný index je také použit pro rodné číslo pacienta, čímž je zabezpečeno, že nedojde k duplicitě záznamů. Rodné číslo je také hlavním identifikátorem při importu dat pacienta z NIS Medea. Pomocí cizích klíčů jsou k hlavní tabulce pacientů navázány tabulky, které odpovídají odborným pracovištím a jejich činnostem. Aby bylo možné sledovat změny v datech, kritické tabulky obsahují pole s identifikací uživatele a čas, kdy uživatel vytvořil nebo změnil záznam.

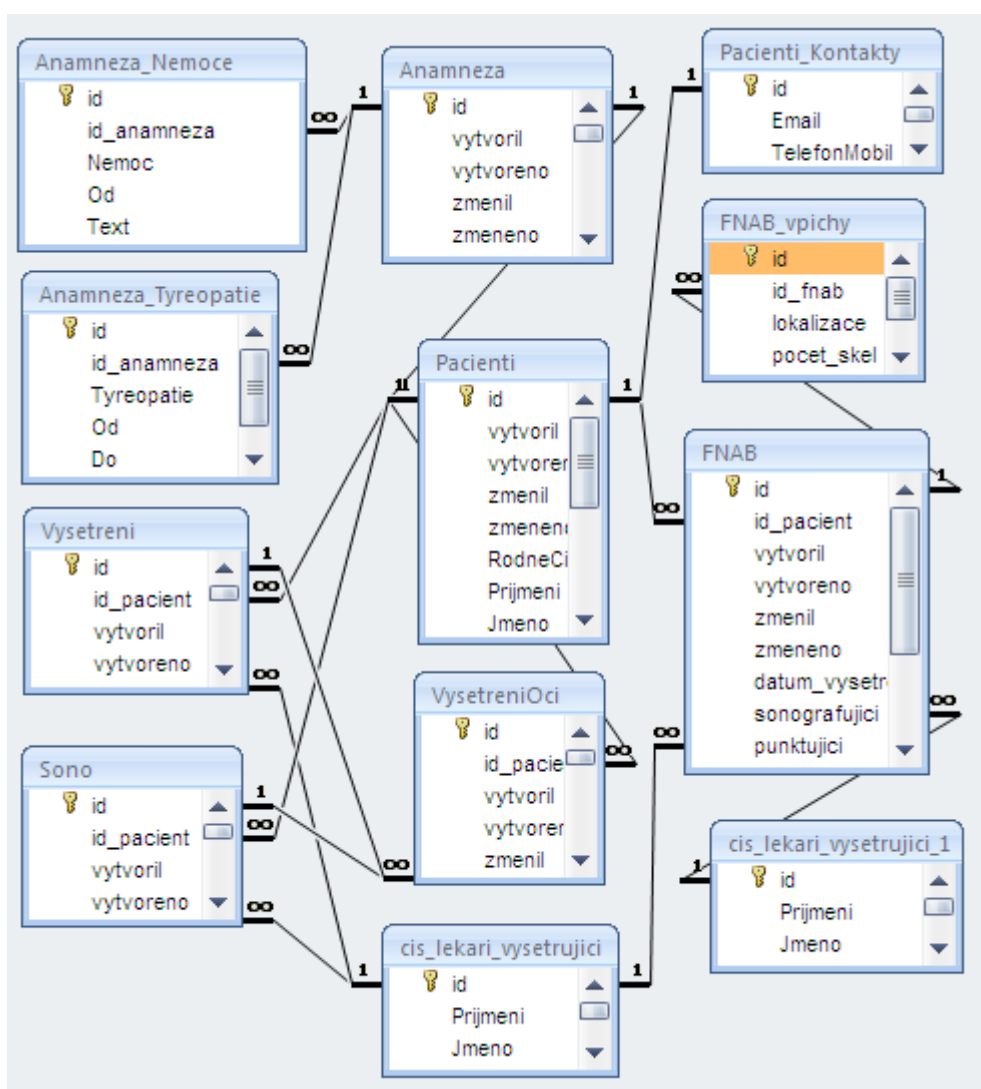
V současné době obsahuje databáze tabulky pro endokrinologické a obezitologické pracoviště. Obě pracoviště mohou nezávisle na sobě pracovat

s databází, mají své vlastní tabulky pro ukládání strukturovaných dat, přitom ale používají společnou tabulku pacientů. Díky tomu je možné v budoucnu vyhledávat výzkumná data napříč odbornými pracovišti 3. interní kliniky.

Databáze se v současné době skládala z 30 tabulek. Tabulky v databázi lze dělit ze dvou pohledů. První je podle odborných pracovišť kliniky, což je endokrinologie a obezitologie, druhý pohled je dělení tabulek podle druhu dat, jež obsahují:

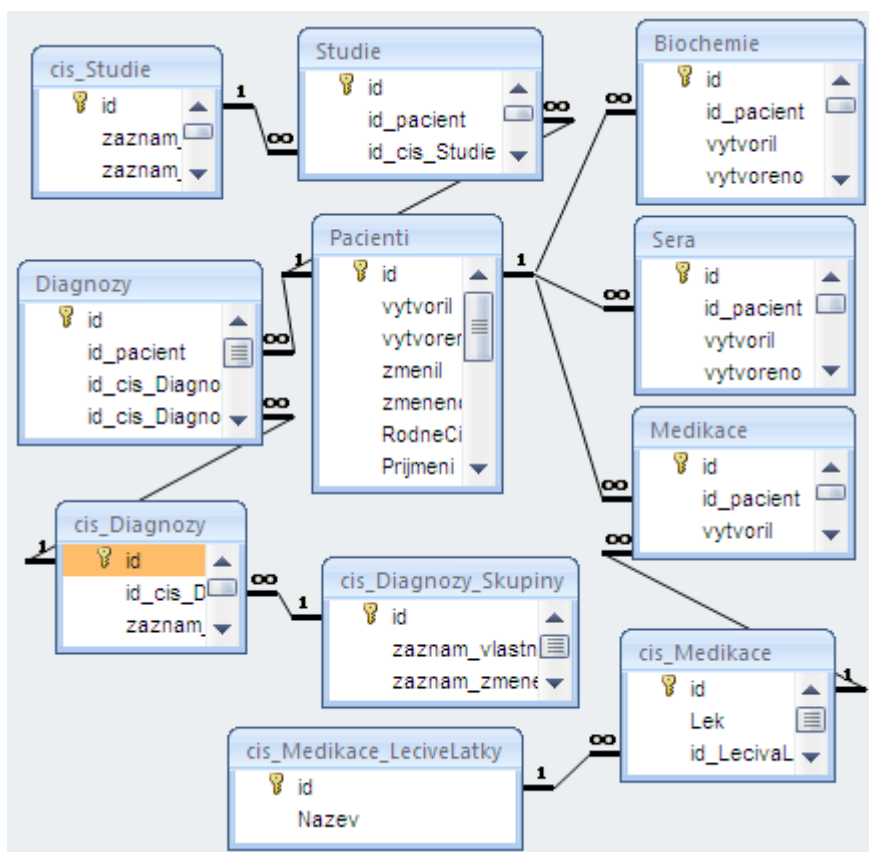
1. Administrativní data. Jde zejména o tabulky pacientů, kontaktů, lékařů a uživatelů v aplikaci.
2. Klinická data. Tabulky klinických vyšetření nebo vyšetření obrazovými metodami.
3. Výzkumná data. Databáze obsahuje tabulky se specializovanými daty, které slouží k výzkumným účelům. Struktura tabulek je flexibilně přizpůsobována aktuálním výzkumným záměrům pracovišť.
4. Číselníky. Tabulky obsahující předdefinované hodnoty pro zamezení duplicity dat a zrychlení zadávání dat do aplikace. Zároveň se tím zjednodušuje vyhledávání informací v databázi, protože je možné prohledávat předem definované hodnoty.

Vzhledem k rozsáhlosti relačního schématu databáze, rozdělil jsem graf schématu na tři části. Na obrázku č. 1 je zobrazeno relační schéma tabulek endokrinologického oddělení obsahující klinická data.



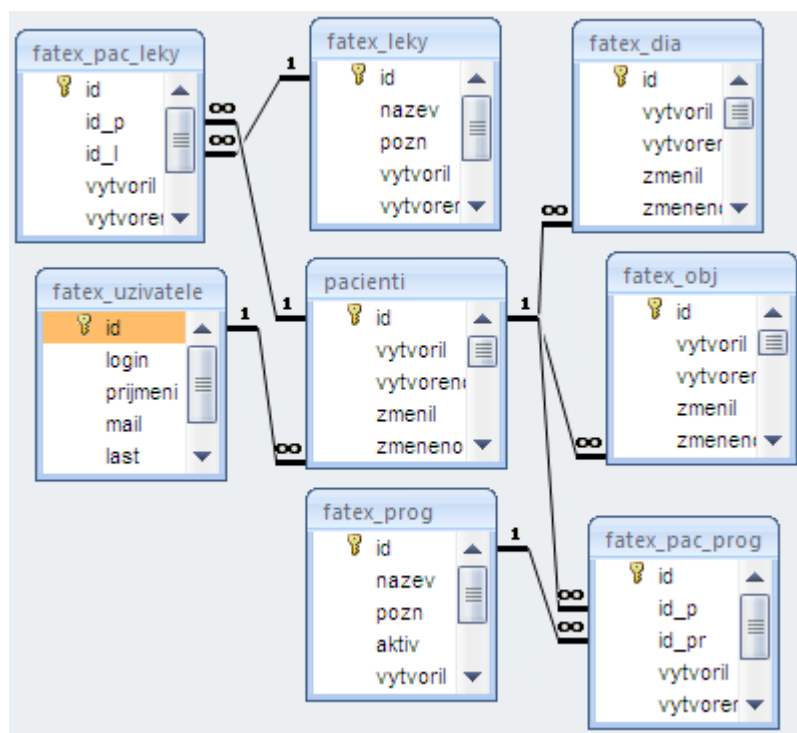
Obrázek č. 1: relační schéma endokrinologické části databáze – klinická data

Obrázek č. 2 obsahuje také tabulky endokrinologického oddělení, ale s výzkumnými daty.



Obrázek č. 2: relační schéma endokrinní části databáze – výzkumná data

Třetí součástí databáze jsou tabulky pro obezitologické oddělení. Na obrázku č. 3 zobrazují všechny tabulky využívané tímto oddělením.



Obrázek č. 3: relační schéma obezitologické části databáze

V dalším rozvoji aplikace bych chtěl pracovat na konsolidaci některých tabulek, hlavně pro oblast výzkumu. Cílem by měla být jednotná databáze umožňující univerzální použití pro všechna oddělení kliniky, protože plánujeme rozšíření na další pracoviště, například diabetologii.

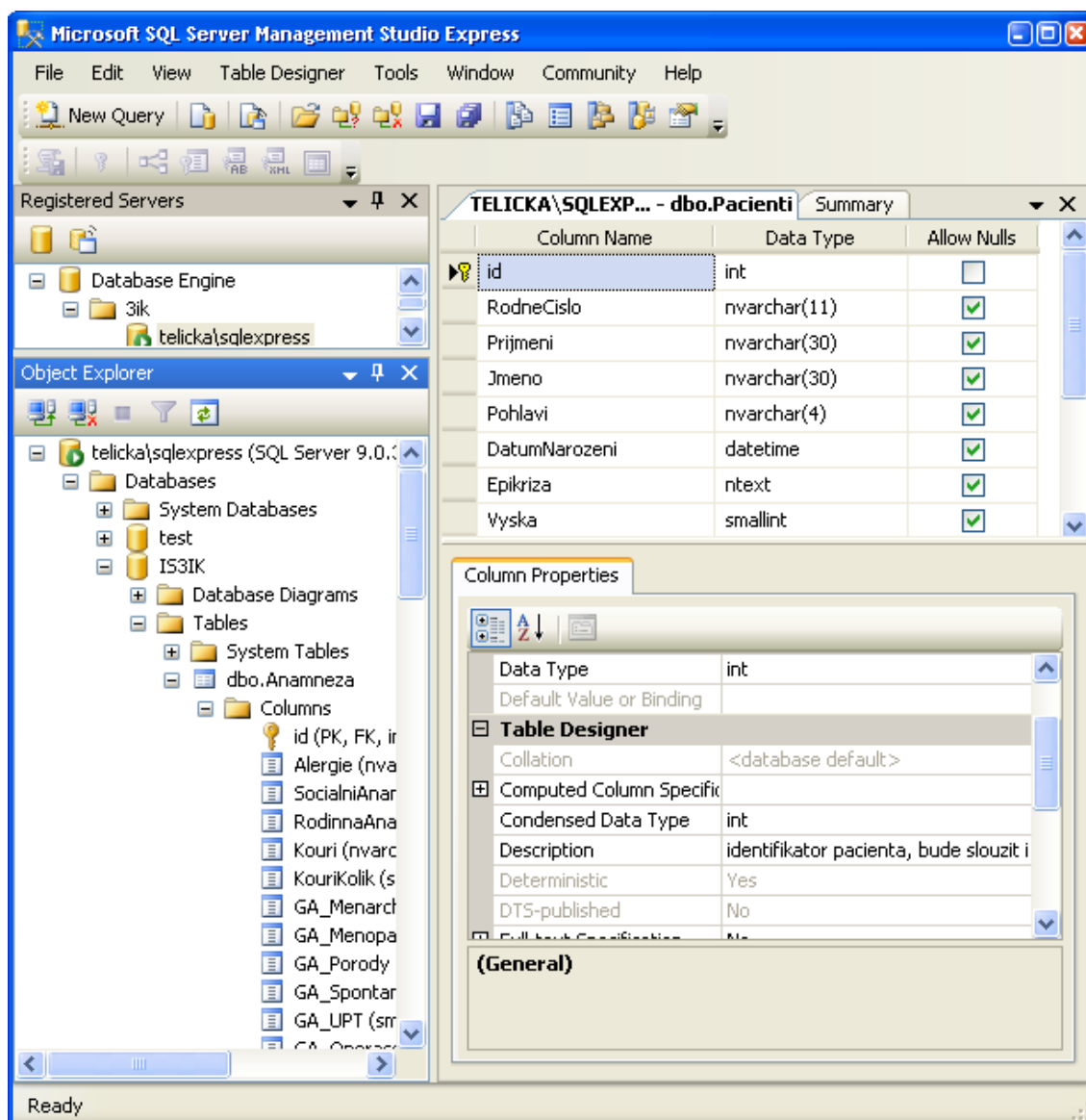
5.2 Migrace databáze Access na SQL server

Původní aplikace byla vyvíjena jen jako souborová databáze bez možnosti autentizace a autorizace. Migrací databáze na SQL server dojde ke zvýšení zabezpečení dat na stejnou úroveň, jaké je využíváno v běžných profesionálních aplikacích. Navíc, SQL server je výkonnější, než souborová databáze v souboru Accessu. Pro migraci byl využit nástroj migrace na SQL server, který je součástí Microsoft Access. Na SQL serveru, který je umístěn na úseku informatiky byla vytvořena prázdná databáze a k ní jsem získal administrátorský přístup. Při migraci jsem vytvořil nový MDA soubor Access Project, který obsahuje pouze formuláře a kód. Na SQL server byly přemístěny

tabulky, vztahy mezi nimi a dotazy, které bylo nutné změnit na pohledy. V MDA souboru jsou pak tyto tabulky nalinkovány k SQL serveru.

Migrace databáze na SQL server přinese uživatelům zjednodušení práce ve Windows. Pro přístup k původní databázi jsem musel zajistit, aby měli všichni uživatelé přístup k síťovému adresáři, ve kterém byla aplikace nainstalována. SQL server umístěný na úseku informatiky je dostupný v celé síti fakultní nemocnice a tím mají uživatelé zajištěnu konektivitu k databázi z kteréhokoliv pracoviště kliniky.

Migrace na databázový server je z hlediska vlastní databáze dobře zvládnutelným krokem. Zajímavostí je, že v současné době řeším problémy s převodem nikoliv dat, ale některých částí zdrojových kódů. Touto problematikou se zabývám v kapitole 6.2 Programování ve Visual Basic for Applications, generování zpráv. Na obrázku č. 4 je ukázka zobrazení databáze v nástroji SQL Server Management Studio Express. V levé části je vyobrazena stromová struktura objektů na databázovém stroji. Napravo je otevřena v editačním režimu jedna z tabulek vyvíjené databáze.



Obrázek č. 4: Vývoj a správa databáze v Microsoft Management Studio Express

5.2 Autentizace a autorizace

Pro přístup k databázi je využívána autentizace, což znamená ověření identity uživatele, který se chce do aplikace přihlásit. Ta probíhá na straně SQL serveru, který je umístěn v serverovně úseku informatiky. Všeobecná fakultní nemocnice se snaží prosazovat jednotný systém autentizace pomocí Microsoft Windows Domain [20]. Všichni zaměstnanci v nemocnici mají v doméně Windows vytvořen uživatelský účet, čehož se právě využívá i u databázových aplikací. Na SQL serveru jsou vytvořené

loginy, které jsou napojeny na doménu Windows. Při přihlášení dojde k ověření jména a hesla vůči doméně Windows. Tímto je odstraněna nutnost vytvářet duplicitní uživatelské účty na SQL serveru. Uživatelům stačí pouze jeden login, který používají pro přihlášení ke svému e-mailu, do sítě, anebo do databázových aplikací. Na druhé straně je to také zjednodušení pro úsek informatiky, protože zabezpečuje podporu pouze k jediné databázi s uživatelskými účty. Tento způsob autentizace má ještě jeden podstatný přínos. Pokud je do domény Windows připojená i klientská stanice, pak se uživatelé přihlašují pouze jednou při přihlášení na klientskou stanici. Při spouštění aplikací využívajících doménu Windows dojde k automatické autentizaci a uživatelé již nejsou dotazováni na zadání jména a hesla.

Na Microsoft SQL serveru lze snadným způsobem vytvářet uživatelské skupiny a do nich sdružovat jednotlivé uživatelské účty s rozdílnými přístupovými právy k různým objektům databáze. Tímto je zabezpečena autorizace uživatelů s rozdílnými přístupovými právy. Jedna skupina lékařů nemusí mít přístup k datům pracoviště, apod.

5.4 Zálohování databáze

K zabezpečení databáze patří také její zálohování pro případ pádu SQL serveru nebo zničení databázových souborů. Databáze je zálohována dvěma způsoby. Protože je uložena na serveru úseku informatiky, je tato databáze zahrnuta do záloh, které se na úseku informatiky pravidelně provádějí. Druhým způsobem je ruční zálohování, které umísťují na svůj počítač. Na svém počítači provozují několik kopií databáze. Některé využívám k vývoji a testování nových funkcí aplikace.

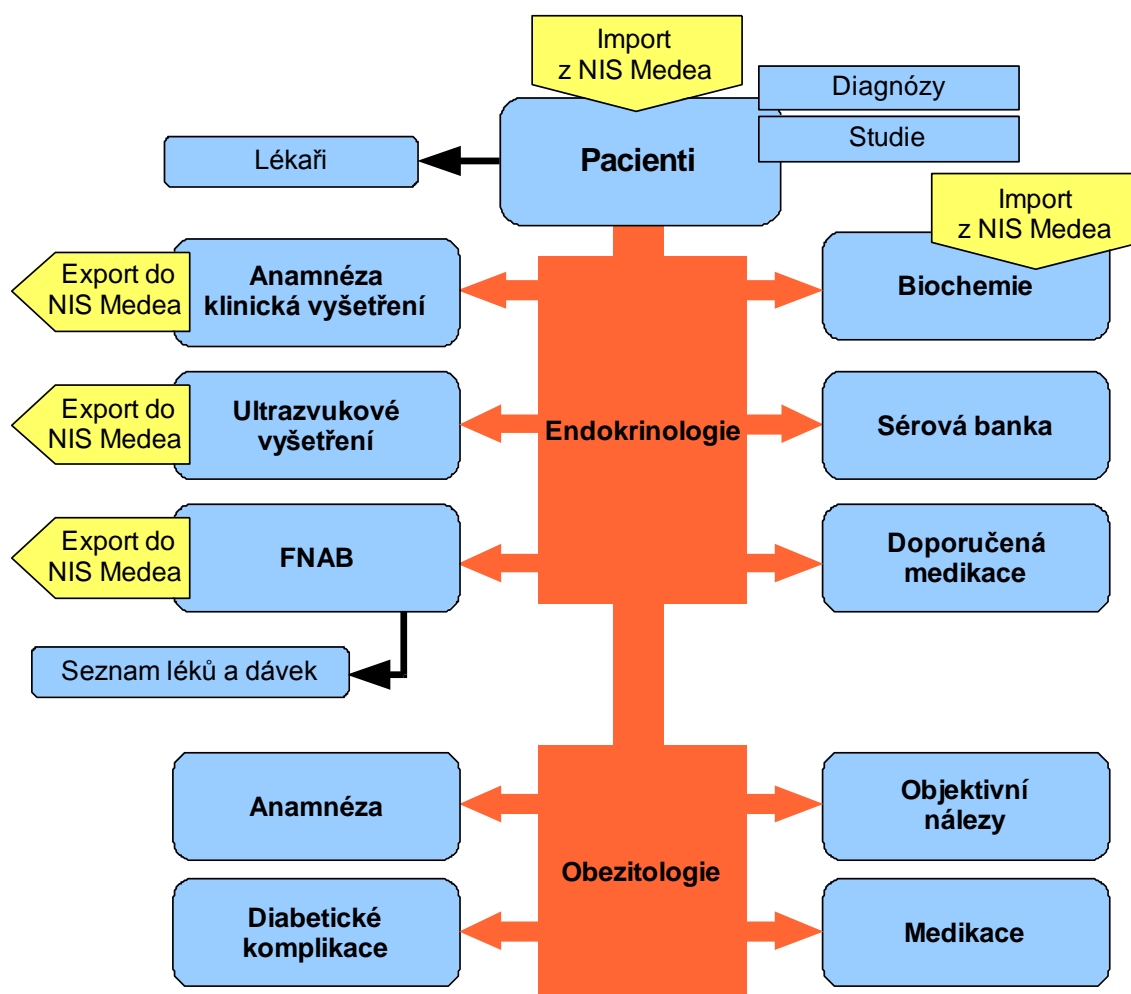
Zálohování je možné jak z SQL Management Studia, tak i z Microsoft Access Project. Osvědčila se mi práce se zálohami v Management Studiu, kde mám nad databází plnou kontrolu. Pro přístup k fyzicky uloženým souborům na serveru jsem také získal zabezpečený přístup, jelikož soubory lze zálohovat pouze na umístění viditelná z SQL serveru, nikoliv z klientské stanice.

6. Vývoj klientské aplikace

6.1 Návrh grafického uživatelského prostředí

Přestože vyvíjená aplikace je typu klient-server, pro vývoj klientské aplikace stále používáme Microsoft Access. Výhodou je to, že na všech pracovištích kliniky se používá Access jako součást sady Microsoft Office a není tedy nutné se zabývat vývojem nového programu v některém jiném vývojovém prostředí. Tím nevznikají další časové nároky na instalaci klientských aplikací, jejich správu, apod. Díky implementaci programovacího jazyka Visual Basic for Applications v sadě MS Office je možné vytvořit aplikace, které splňují rozličné nároky na automatizaci mnohá činnosti uživatelů, resp. lékařů. Ti mají k dispozici klientskou aplikaci, která poskytuje téměř stejný komfort ovládání a množství funkcí, jako by byla aplikace vyvíjená v profesionálním vývojovém prostředí, např. Microsoft Visual Studiu nebo Delphi.

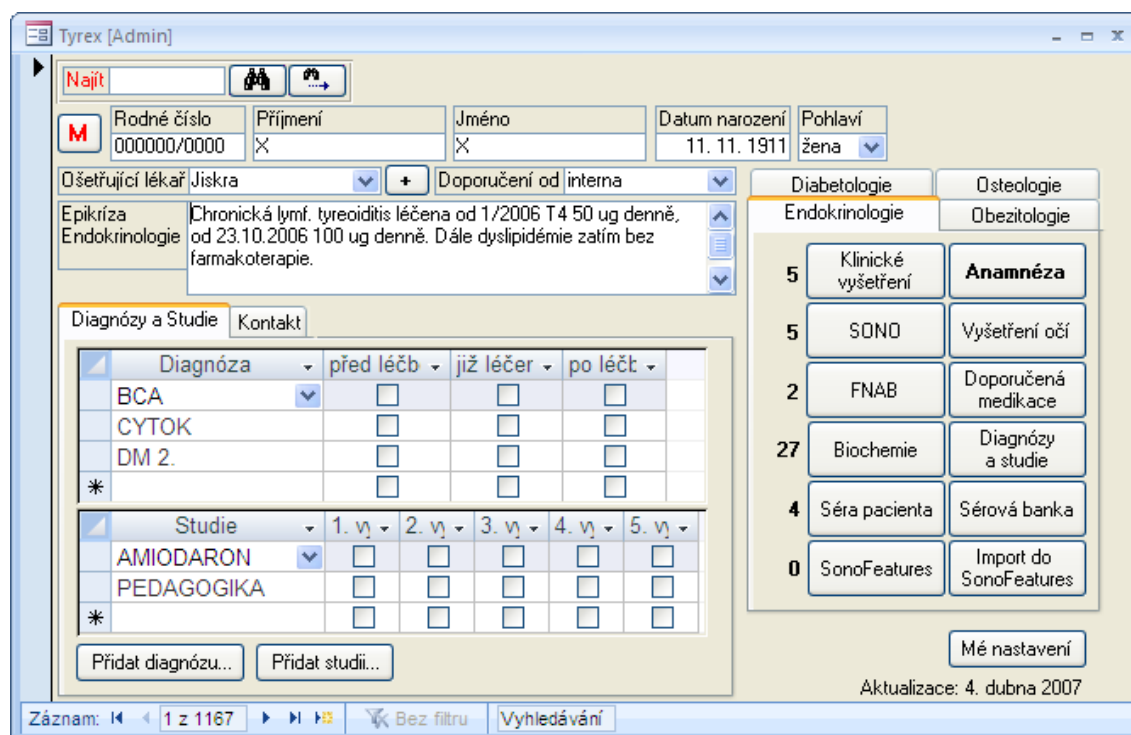
Klientská aplikace se skládá z modulů znázorněných na obrázku č. 5. Po přihlášení do databáze se lékařům zobrazuje hlavní modul s tabulkami pacientů a jejich zařazením do diagnóz a studií. Získávání identifikačních údajů pacientů z NIS je popsáno v kapitole 6.3 Programování komunikace s NIS pomocí Windows API. Z hlavního okna je přístupná endokrinologická a obezitologická část databáze. Lékaři zadávají data v modulech, které mohou být klinického nebo výzkumného charakteru. Nejrozvinutější jsou moduly pro endokrinologii. Kromě zadávání dat je součástí modulů také kód pro generování zpráv ze strukturovaných dat zadaných během vyšetření pacienta a pro komunikaci s NIS. Obezitologické moduly byly naprogramovány podle předlohy internetové aplikace, která byla používána před spojením s endokrinologickou databází. V současné době rozšiřují obezitologické moduly o komunikaci s NIS. Z obrázku je patrné, že se nesnažíme o sjednocení modulů pro různá odborná pracoviště, ale jsou programovány přesně podle zvyklostí lékařů daných pracovišť.



Obrázek č. 5: Schéma modulů aplikace

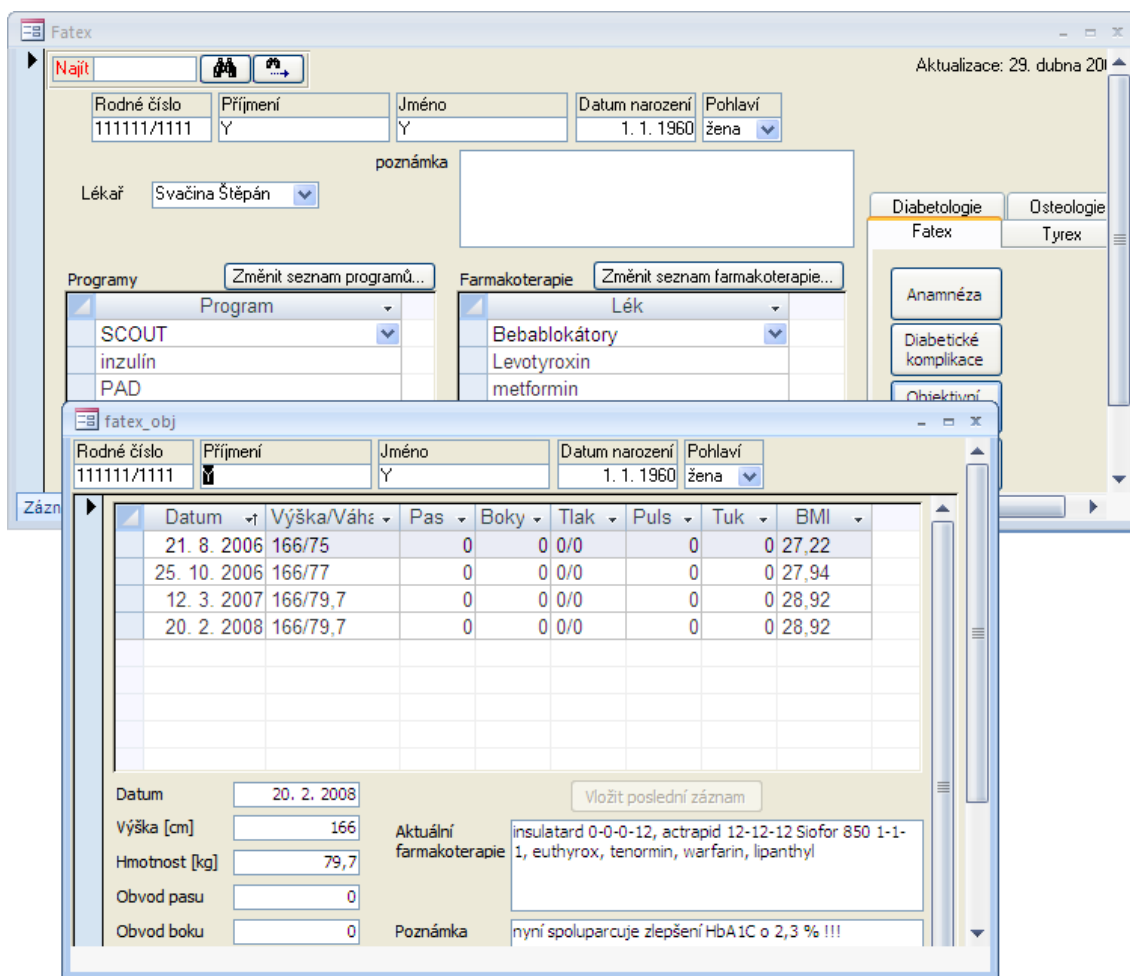
Na obrázku č. 6 je ukázka hlavního okna aplikace při práci v endokrinologické části aplikace. Pro zobrazení stejného pacienta jako v NIS stačí stisknout tlačítko „M“. Pokud pacient není nalezen, automaticky se vytvoří jeho záznam dle informací načtených z NIS. Lékaři mají ihned po zobrazení pacienta přehled o diagnózách a studiích. Tato funkce nahrazuje nevyhovující dispenzář v NIS StaproMedea, ve kterém nebylo možné členit pacienty do složitějších kategorií, ani zobrazit pacienty v tabulkách dle filtrů. Lékaři tak mají ihned k dispozici rozsáhlé tabulky s daty pacientů dle zadaných vyhledávacích kritérií. V pravé části okna jsou umístěna tlačítka pro vstup do klinických nebo výzkumných modulů aplikace. Zároveň je možné se snadno přepínat mezi odbornými pracovišti pomocí záložek. V přípravě je rozvoj databáze pro diabetologii a osteologii. Zároveň přemýšlíme o rozšíření na kardiologické pracoviště,

kde používají vlastní řešení sběru výzkumných dat. Tímto bychom obsáhli všechna pracoviště 3. interní kliniky a z aplikace se může stát robustní výzkumný informační systém, nad kterým bude mít klinika plnou kontrolu, což je podstatná výhoda oproti programování databáze profesionální firmou nebo integraci aplikace do nemocničního informačního systému.



Obrázek č. 6: Hlavní okno aplikace – endokrinologická část

Na obrázku č. 7 je zobrazeno hlavní okno pro obezitologické pracoviště a spolu s ním je v dolní části modul obsahující objektivní vyšetření pacienta. Z obrázku je vidět, že v současné době používají endokrinologické i obezitologické pracoviště odlišné hlavní okno. Je to dáno vysokou časovou náročností při vývoji těch částí programu, které jsou společné pro všechna pracoviště. V dolní části obrázku se nachází modul pro objektivní výsledky vyšetření, které lékař zadává během vyšetření pacienta v ambulanci. K dispozici je přehledný tabulkový seznam všech předešlých vyšetření.



Obrázek č. 7: Hlavní okno obezitologického pracoviště a objektivní vyšetření

6.2 Programování ve Visual Basic for Applications, generování zpráv

Prostředí Accessu poskytuje programátorovi dostatečný soubor možností, jak uživatelské prostředí aplikace automatizovaně propojit s databází. Není nutné psát složité kódy pro přístup k datům, aby se na formuláři zobrazila jen data dle uživatelských parametrů, apod. U rozsáhlejších projektů, jako je tento, jsem již musel využít implementace programovacího jazyka Visual Basic. V aplikaci jsem programoval doplňkové funkce, které přistupují k bázi dat přímo, například pro manipulaci s daty, apod. V současnosti pracuji na převodu kódu pro přístup k datům, který v počátečních verzích aplikace využíval odlišných knihoven, než které vyžaduje

aplikace nyní. U databázových aplikací typu klient-server je nutné psát kód s využitím knihoven OLE DB. Tento převod kódu znamená jediné úskalí při migraci databáze na SQL server. Získám tím ale širší možnosti ve využití funkcí zabudovaných v databázovém stroji Microsoft SQL Server. V kódu pro řízení dat je možné využívat transakční protokol T-SQL, který slouží pro programování na straně serveru. Z klientské aplikace je tak možné ovládat funkce nebo spouštět složitější operace na straně SQL serveru. Tím dochází ke snížení hardwarových nároků na klientských počítačích a ke snížení datových toků v síti. V následující ukázce zdrojového kódu je vidět propojení VBA, knihoven pro přístup k datům a T-SQL jazyka. Funkce slouží pro připojení k databázi a pomocí T-SQL příkazů („SELECT SYSTEM_USER“) vrátí login přihlášeného uživatele.

```
Function GetSqlUser() As String
Dim qrd As DAO.QueryDef
rst As DAO.Recordset

Set qrd = CurrentDb.CreateQueryDef("")
With qrd
.Connect = CurrentDb.TableDefs(0).Connect
.SQL = "SELECT SYSTEM_USER"
.ReturnsRecords = True
End With

Set rst = qrd.OpenRecordset()
GetSqlUser = rst(0).Value

rst.Close
Set rst = Nothing
Set qrd = Nothing
End Function
```

Programovací jazyk VBA využívám k automatické extrakci strukturovaných dat z výsledků v mixéru NIS. Mixér umožňuje výpis výsledků jen v textové podobě. Také jsem naprogramoval algoritmy ke generování textových zpráv ze strukturovaných dat uložených v databázi. Důvodem je usnadnění práce lékařů, aby nemuseli ručně psát zprávy z vyšetření do informačního systému. Na obrázku č. 8 je zobrazen modul pro ultrazvukové vyšetření štítné žlázy. Z obrázku je patrné, že se lékaři snaží uložit co nejvíce parametrů z vyšetření zobrazovacími metodami. Naproti tomu v nemocničním

informačním systému se ukládají pouze textové zprávy z vyšetření. Kdyby se nepoužívala tato aplikace, lékaři by data buď ztráceli, nebo by si museli vést vlastní tabulky v tabulkových kalkulátorech. A to je právě činnost, kterou má tato aplikace zefektivnit. V dolní části obrázku v poli „Zpráva“ je automaticky vygenerována zpráva ze zadaných strukturovaných dat. Text se automaticky přizpůsobuje vyplněným polím. Kód pro generování zprávy patří k nejsložitějším částem aplikace. Přínosem pro lékaře je ale kvalitně sestavená zpráva, která je v NIS téměř nerozeznatelná od zprávy psané ručně. Ještě jsem se v klinické praxi neseťkal s podobně komplexní funkcí pro generování zpráv a v ostatních aplikacích je často znát, že zpráva byla generována počítačem.

Obrázek č. 8: Modul ultrazvukového vyšetření na endokrinologii

6.3 Programování komunikace s NIS pomocí Windows API

V průběhu vývoje aplikace jsem hledal možnosti, jak alespoň částečně zautomatizovat výměnu dat mezi aplikací a nemocničním informačním systémem. Výměna dat mezi NIS StaproMedea a aplikací probíhá ve třech oblastech:

1. Import identifikačních údajů pacienta z NIS StaproMedea do aplikace. To jsou například rodné číslo, příjmení, datum narození, apod.
2. Import výsledků biochemických vyšetření a zpráv z cytologických vyšetření štítné žlázy uložených v laboratorním modulu NIS.
3. Export zpráv z aplikace do NIS vygenerovaných ze strukturovaných informací v databázi.

Lékaři začali aplikaci používat již od prvních zkušebních verzí aplikace. Zpočátku pracovali tak, že po zobrazení pacienta v NIS StaproMedea bylo nutné stejného pacienta vyhledat i ve vyvinuté aplikaci a pokud nebyl nalezen, ručně zadávali iniciály pacienta do databáze. Při importu biochemických vyšetření bylo zase nutné přes schránku Windows překopírovat data z NIS systému do aplikace, kde byla data převedena do databáze.

Chtěl jsem využít mé zkušenosti ze zaměstnání a postupně jsem začal programovat ve VBA kód, který volá externí funkce z DLL knihoven Windows API. Windows API je sada knihoven pro nízkoúrovňové programování v prostředí operačního systému Windows a popsal jsem jej v kapitole 3.4. V následující ukázce kódu je seznam deklarovaných funkcí a konstant pro práci s Windows API:

```
Declare Function GetForegroundWindow _  
    Lib "user32.dll" () As Long  
Declare Function GetWindow Lib "user32" _  
    (ByVal hWnd As Long, ByVal wCmd _  
    As Long) As Long  
Declare Function GetWindowText Lib "user32" _  
    Alias "GetWindowTextA" (ByVal hWnd As Long, _
```

```

    ByVal lpString As String, ByVal cch As Long) As Long
Declare Function GetClassName Lib "user32" Alias _
    "GetClassNameA" (ByVal hWnd As Long, _
    ByVal lpClassName As String, _
    ByVal nMaxCount As Long) As Long
Declare Function SendMessageS Lib "user32" Alias _
    "SendMessageA" (ByVal hWnd As Long, _
    ByVal wParam As Long, ByVal lParam As Long, _
    ByVal lParam As String) As Long
Declare Function EnumChildWindows Lib "user32" _
    (ByVal hWndParent As Long, _
    ByVal lpEnumFunc As Long, _
    ByVal lParam As Long) As Long

```

```

Private Const GW_HWNDNEXT = 2
Public Const WM_GETTEXT = &HD
Public Const GW_CHILD = 5

```

Součástí aplikace je rozsáhlý kód, který načítá data z objektů v oknech NIS StaproMedea. Vývoj kódu patří, vedle automatického generování zpráv, také ke složitějším funkcím systému. Kód bylo nutné přizpůsobit různým modulům nemocničního informačního systému, protože NIS StaproMedea ne vždy tvoří jeden homogenní celek. Příkladem je radiodiagnostický modul používaný na ultrazvukovém pracovišti, který je naprosto odlišný od zbytku NIS. Vzniklo to zřejmě tím, že radiodiagnostický modul byl původně samostatná aplikace, která byla napojena k NIS StaproMedea.

Naprogramovaná komunikace je jen pasivní, tzn. NIS neobsahuje žádné funkce, které by umožnily reakci systému na akce prováděné v aplikaci. Snažil jsem se jen využít současných možností, protože úpravy NIS dle požadavků kliniky nejsou možné. Výsledkem programátorské práce je sada efektivních funkcí šetřící čas lékařů, kteří pracují zároveň s aplikací a NIS StaproMedea. V aplikaci se při stisku tlačítka automaticky vyhledá stejný pacient, který je zobrazen v NIS, anebo je automaticky vytvořen nový záznam s nalezenými údaji. Při importu výsledků biochemických vyšetření stačí lékařům jedním stiskem tlačítka přenést veškerá data z laboratorního modulu nemocničního systému. V současné době pracuji na automatickém exportu vygenerované zprávy z aplikace do NIS.

V následující ukázce na obrázku č. 9 je zobrazen modul výsledků v NIS a zpráva vygenerovaná mixérem tohoto modulu. StaproMedea nepodporuje export laboratorních výsledků v podobě, která by dovolovala jednoduchý import do databázových tabulek. Využívám proto funkci mixéru, která uživatelům dovoluje získat výsledky ve formě, kterou by bylo možné programově zpracovávat. Vygenerovanou zprávu pak přes Windows API uložím do vyvinuté aplikace a jednotlivá data převádím na strukturované informace, které vkládám do tabulky biochemických vyšetření.

09/03/08	27/03/08	03/04/08	11/04/08	12/04/08	22/04/08	01/05/08	06/05/08	Třídy a metody
1CH-AMCH	2i-PRIAM	KDDL-AV	1CH-AMCH	1CH-AMCH	GER-AMB	1CH-AMCH	NEU-UZ	0
								Text
	Ergo							
			13,90	8,30	10,90	9,20	4,80	
				1				
				2				

Obrázek č. 9: Výsledky v modulu NIS StaproMedea

Na obrázku č. 10 je zobrazeno okno s biochemickými daty po importu výsledků z NIS do databáze aplikace. Data jsou při stisku tlačítka „Importuj do tabulky“ automaticky načtena z otevřeného okna mixéru, zpracována a vložena do tabulky. Z obrázku je patrné, že nad takto strukturovaně uloženými daty se již snadno vytvářejí vyhledávací dotazy v jazyku SQL pro zpracování ve statistických analýzách, apod. V databázi je tak možné vyhledávat korelace mezi biochemickými výsledky,

objektivními a subjektivními vyšetřeními v ordinaci a parametry z vyšetření zobrazovacími metodami.

Výsledky z Medey, které nebyly vloženy do tabulky:

Nastavení mixéru Medey...
Ulož označené do souboru ...
Importuj do tabulky

Kontakt na vývojáře:
e-mail: zdenek@telicka.cz
tel.: 732 305 039

Datum	Leukocyty	Erytrocyty	Hemoglobin	Hematokrit	Str_obj_eryt	Barivo_eryt	Str_barev_kon
10. 3. 2006 8:48:00				140			
2. 10. 2006 18:07:00							
13. 10. 2006 6:47:00							
3. 10. 2006 11:54:00	10,96	3,56	105	0,318		29,4	
1. 11. 2006 7:00:00							
6. 11. 2006 11:05:00							
6. 11. 2006 11:26:00	5,94	4,38	134	0,38	86,8	30,6	3
6. 11. 2006 12:38:00							
6. 11. 2006 12:39:00							
28. 11. 2006 9:37:00							
30. 9. 2004 8:24:00							
16. 8. 2005 9:48:00							
6. 2. 2007 11:34:00							
9. 1. 2008 8:48:00							

Záznam: 1 z 27 Bez filtru Vyhledávání

Obrázek č. 10: Okno s biochemickými výsledky po importu z NIS

7. Závěr

7.1 Přínos a využití aplikace

V této diplomové práci jsem se snažil shrnout výsledky mé spolupráce s lékaři na 3. interní klinice 1. lékařské fakulty Univerzity Karlovy a Všeobecné fakultní nemocnice v Praze. Popsal jsem postup vývoje od jednoduché souborové databáze, přes sloučení databází dvou odborných pracovišť, až po migraci databáze na SQL server. Ukázal jsem, že s využitím programových prostředků a technologií, dostupných ve Všeobecné fakultní nemocnici v Praze, lze vyvinout robustní aplikaci typu klient-server, která poskytne výzkumným týmům nástroj pro sběr a analýzu složitých výzkumných dat. Popisovaná databázová aplikace zvyšuje bezpečnost a snižuje duplicitu ukládaných

dat oproti konvenčnímu způsobu sběru výzkumných dat pomocí tabulkových kalkulátorů.

Aplikace přináší lékařům nové možnosti, které vzbuzují na klinice velký zájem o rozšíření na další odborná pracoviště. První je komunikace s nemocničním informačním systémem při výměně dat. Díky tomu není pro lékaře práce s aplikací složitá a časově náročná. Druhou je automatické generování textových lékařských zpráv ze strukturovaných dat uložených v databázi. Takto generované zprávy jsou v nemocničním informačním systému v podstatě k nerozeznání od zpráv psaných lékaři ručně.

7.2 Budoucí rozvoj aplikace

V současné době je aplikace v provozu na endokrinologickém a obezitologickém pracovišti kliniky. Dostupnost a spolehlivost databáze je zajištěna díky spolupráci s úsekem informatiky, kde je databáze provozována na profesionální verzi SQL serveru spolu s ostatními aplikacemi pro management fakultní nemocnice. Díky tomu bude možné využít jednotné doménové uživatelské účty pro autentizaci a autorizaci v databázi.

V blízké budoucnosti plánujeme její rozšíření na další pracoviště kliniky ve spolupráci s přednostou kliniky a ostatními lékaři. Neklademe si za cíl, aby se z aplikace stal masově nasazený systém, ale chtěli bychom, aby naplnila výzkumné potřeby vědeckých pracovníků, kteří ve svých ambulancích léčí stovky nových pacientů ročně. V těchto případech popisovaný program vyplňuje mezeru v nabídce trhu a stává se zajímavým i pro komerční využití. Oproti komerčnímu řešení má tento projekt výhodu v tom, že nad daty a rozvojem databáze má klinika plnou kontrolu a nemusí se spoléhat na finančně nákladné služby firem. Projekt je také zajímavou ukázkou spolupráce studentů, lékařů a zaměstnanců fakultní nemocnice.

Vyvíjená databáze se stává významným zdrojem informací pro analýzu nejen klinických, ale i ekonomických dat. V dalším studiu bych se proto chtěl věnovat

analýze nákladové efektivity léčby endokrinních a metabolických onemocnění a jejich trendů.

Na takto rozsáhlém projektu nepracuji sám a k jeho rozvoji přispěli:

Jan Blažíček	Actinet Informační systémy s.r.o.
Ing. Tomáš Brauner	úsek informatiky VFN
MUDr. Jan Jiskra, PhD.	3. interní klinika 1. LF UK a VFN
MUDr. Martin Matoulek, PhD.	3. interní klinika 1. LF UK a VFN
Doc. MUDr. Ing. Daniel Smutek, PhD.	3. interní klinika 1. LF UK a VFN
Ing. Milan Soldát	úsek informatiky VFN
Actinet Informační systémy s.r.o.	
business communication s.r.o.	
Pfizer spol. s r.o.	

8. 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. Tanzmann, D.:Integrace elektronické formy ZK do NIS VFN a její přínosy. Diplomová práce; 1. lékařská fakulta Univerzity Karlovy v Praze, 2004.
5. Progress® Query/Results™: White Papers. [online], [cit. 2008-05-14], http://www.progress.com/openedge/products/manage/query_results/index.ssp
6. Lékařská doporučení – guidelines: Textová lékařská doporučení. [online], [cit. 2008-04-12], <http://www.euromise.cz/health/recommendation.html>
7. Zhou, X. and Han, H. and Chankai, I. et. al.: Converting Semi-structured Clinical Medical Records into Information and Knowledge. In Proceedings of the 21st International Conference on Data Engin, 2005.
8. SAS/ACCESS 9.1 Suppelment for ODBC (SAS/ACCESS for Relational Databases); SAS Publishing; 2004.
9. Císař P.: InterBase/Firebird tvorba, administrace a programování databází. Computer Press, 2003.
10. Sack, J: Velká kniha T-SQL & SQL Server 2005. Zoner press, 2007.
11. Lacko, L.: Oracle správa, programování a použití databázového systému. Computer Press, 2007.
12. Granor, TE; Hennig, D; McNeish, K.: Přecházíme na Visual FoxPro 7.0, Computer Press, 2002.
13. Viescas, JL.: Mistrovství v Microsoft Office Access 2003. Computer Press, 2005.
14. Dobson, R.: Programování v Microsoft Access 2000. Computer Press, 2000.
15. Information Technology - Database Language SQL; ISO/IEC 9075:1992.
16. Bičíšřová, P.: Nové moderní technologie v produktovém portfoliu společnosti Stapro. [online], [cit. 2008-05-01], <http://www.stapro.cz/aktuality/nove-moderni-technologie-v-produktovem-portfoliu-spolecnosti-stapro.aspx>

17. Telička, Z.: Databázová aplikace pro výzkum nemocí štítné žlázy. Bakalářská práce; 1. Lékařská fakulta Univerzity Karlovy v Praze, 2006.
18. Castagnetto, J; Rawat, H.; Schumann, S.; Scollo, C.; Veliath, D.: PHP Programujeme profesionálně, Computer Press, 2002.
19. Petzold, C.: Programování ve Windows API, Computer Press, 1999.
20. Russel, C., Crawford, S., Gerend, J.: Microsoft Windows Server 2003 Velký průvodce administrátora, Computer Press, 2005.
21. Stapro: Klinický informační systém, [online], [cit. 2008-05-15], <http://www.stapro.cz/reseni-pro-zdravotnictvi/klinicky-informacni-system/pacientska-administrativa-a-klinicky-informacni-system.aspx>
22. Microsoft corp.: Přehled produktu SQL Server 2005 [online] [cit. 2008-05-02], <http://www.microsoft.com/cze/windowsserversystem/sql/prodinfo/overview/default.aspx>

9. Přílohy

Obsah CD-ROM:

- Elektronická forma diplomové práce