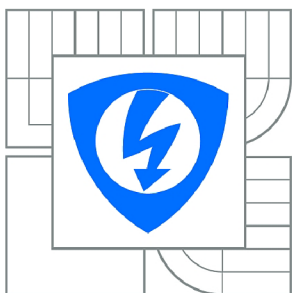




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

DATOVÝ STANDARD ZDRAVOTNICKÝCH INFORMAČNÍCH SYSTÉMŮ

DATA STANDARD OF THE HEALTH INFORMATION SYSTEMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN HOLUB

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR FEDRA

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav biomedicínského inženýrství

Diplomová práce

magisterský navazující studijní obor
Biomedicínské a ekologické inženýrství

Student: Bc. Martin Holub

ID: 83743

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Datový standard zdravotnických informačních systémů

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte datový standard Ministerstva zdravotnictví České republiky, národní zdravotní registry a legislativu o povinných hlášeních do těchto registrů. Dále prostudujte technologii nemocničního informačního systému Clinicom a možnosti vzdáleného přístupu k serveru pomocí CSP (Caché Server Pages).

Práce musí obsahovat:

- návrh a realizaci bezpečného vzdáleného přístupu oprávněného lékaře;
- návrh a realizaci uživatelského webového rozhraní v CSP včetně generování zprávy v DS MZ o pacientech k povinnému hlášení do registrů NZIS v zadaném časovém intervalu;
- návrh a realizaci tiskové sestavy o pacientech v předešlém bodě k archivaci;
- analýzu zabezpečení přístupu k serveru NIS a současně datové elektronické komunikace mezi různými zdravotnickými organizacemi v síti Internet;
- zhodnocení funkce celého systému.

DOPORUČENÁ LITERATURA:

[1] KURSTEN, Wolfgang, et al. Caché Databáze postrelačního typu a tvorba aplikací. Brno: CP Books, 2005. ISBN 80-251-0491-5.

[2] MLÝNKOVÁ, Irena, et al. XML technologie. Principy a aplikace v praxi. Praha: Grada Publishing, 2008. ISBN 978-80-247-2725-7.

Termín zadání: 12.10.2009

Termín odevzdání: 21.5.2010

Vedoucí práce: Ing. Petr Fedra

prof. Ing. Jiří Jan, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Martin Holub
Bytem: Modletická 1388, Praha 4, 149 00
Narozen/a (datum a místo): 11. srpna 1984 v Chomutově

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Jiří Jan, CSc, předseda rady oboru Biomedicínské a ekologické
inženýrství
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
 - diplomová práce
 - bakalářská práce
 - jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP: Datový standard zdravotnických informačních systémů

Vedoucí/ školitel VŠKP: Ing. Petr Fedra

Ústav: Ústav biomedicínského inženýrství

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- v tištěné formě – počet exemplářů: 2
- v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 20. května 2010

.....
Nabyvatel

.....
Autor

ABSTRAKT

Práce „Datový standard zdravotnických informačních systémů“ se zabývá strukturou datového standardu Ministerstva zdravotnictví České republiky, včetně jazyka XML. Dále se práce zaměřuje na nemocniční informační systém CLINICOM včetně přístupu k záznamům databáze Caché firmy InterSystems. Součástí práce je rovněž popis Národních zdravotních registrů. Hlavní cíl této práce je návrh webové aplikace, která umožňuje generovat zprávy v DS MZ k povinnému hlášení do registrů NZIS v zadaném časovém intervalu z databáze Caché. Nalezené záznamy z databáze lze vytisknout k archivaci. Práce se také zabývá zabezpečením přístupu k serveru NIS a datové komunikaci mezi zdravotnickými organizacemi.

KLÍČOVÁ SLOVA

Databáze, Caché, Datový standard, NIS, CLINICOM, CSP, SQL, XML, ODBC, Národní zdravotní registry, ÚZIS, Ústav zdravotnických informací a statistiky

ABSTRACT

The diploma thesis "Data Standard of the Health Information Systems" deals with the structure of data standard including XML language. Further the thesis is focusing on hospital information system CLINICOM with the inclusion of the access to data records from its database Caché company InterSystems. Description of the National registers of health is also included. The concept of web application is also one part of the thesis which enables to generate reports in the data standard of the Ministry of Health of the Czech Republic for mandatory reporting by NHIS registers in the time interval from the Caché database. Found records from the database can be printed for archiving. The work also deals with security access to the HIS server and data communication between health organizations.

KEYWORDS

Database, Caché, Data standard, HIS, CLINICOM, CSP, SQL, XML, ODBC, The National registers of health, IHIS, Institute of Health Information and Statistics

BIBLIOGRAFICKÝ ZÁZNAM

HOLUB, M. *Datový standard zdravotnických informačních systémů.*

Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií,
2010. 68 s. Vedoucí diplomové práce Ing. Petr Fedra.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Datový standard zdravotnických informačních systémů“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 20. května 2010

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Petru Fedrovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne 20. května 2010

.....
podpis autora

Obsah

SEZNAM OBRÁZKŮ	10
ÚVOD.....	11
1. DATOVÝ STANDARD (DS)	12
1.1 VERZE DS	12
1.2 PŘEDPOKLÁDANÁ FINÁLNÍ PODOBA DS.....	13
2. STRUKTURA BLOKŮ A SOUBORŮ DATOVÉHO STANDARDU	14
2.1 STRUKTURA BLOKU DS	14
2.2 POPIS BLOKU DS	14
2.3 ZNAKOVÉ SADY V SOBORECH DS.....	16
2.4 STRUKTURA SOUBORU DS.....	16
2.5 NÁZEV SOUBORU DS	18
2.6 UKÁZKA OBSAHU SOUBORU DS	18
3. ROZŠÍŘITELNÝ ZNAČKOVACÍ JAZYK (XML)	21
3.1 HISTORIE ZNAČKOVACÍCH JAZYKŮ (MARKUP LANGUAGE)	21
3.2 KONTROLA JAZYKA XML	22
3.3 SHRnutí JAZYKA XML	22
4. NÁRODNÍ ZDRAVOTNÍ REGISTRY	23
4.1 ÚDAJE A LEGISLATIVA O POVINNÝCH HLÁŠENÍ DO NZR.....	23
4.2 ÚČEL A LEGISLATIVA JEDNOTLIVÝCH ZDRAVOTNÍCH REGISTRŮ	23
5. ZABEZPEČENÍ SERVERU NIS A DATOVÉ KOMUNIKACE	32
5.1 ZABEZPEČENÍ SERVERU NIS	32
5.2 DATOVÁ KOMUNIKACE MEZI ZDRAVOTNICKÝMI ZAŘÍZENÍMI	33
5.3 ZABEZPEČENÍ DATOVÉ KOMUNIKACE	33
5.3.1 VPN (<i>Virtuální Privátní Síť</i>).....	34
5.3.2 SSL (<i>Secure Socket Layer Protocol</i>)	34
5.3.3 S/MIME (<i>Secure/Multipurpose Internet Mail Extension Protocol</i>)	35
5.3.4 PGP (<i>Pretty Good Privacy</i>)	35
6. NEMOCNIČNÍ INFORMAČNÍ SYSTÉM CLINICOM	36
6.1 UŽIVATELSKÉ ROZHRAŇÍ CARECENTER SYSTÉMU CLINICOM	37
6.2 UŽIVATELSKÉ ROZHRAŇÍ NETACCESS SYSTÉMU CLINICOM	38
6.3 UŽIVATELSKÉ ROZHRAŇÍ TELNET/SSH SYSTÉMU CLINICOM.....	39
7. DATABÁZOVÁ PLATFORMA CACHÉ	40
7.1 VZDÁLENÝ PŘÍSTUP K ULOŽENÝM DATŮM V DATABÁZI CACHÉ	40
7.1.1 Ovladač ODBC	40
7.1.2 Architektura ODBC.....	41
7.1.3 Instalace ovladače ODBC.....	41
7.1.4 Nastavení ovladače ODBC.....	41
7.2 REALIZACE PŘÍSTUPU DO DATABÁZE CACHÉ.....	42
7.3 UTILITY CACHÉ (TZV. KOSTKA).....	44
7.3.1 Caché studio.....	45
7.3.2 Caché SQL Manager	45
7.4 CACHÉ UDA (UNIFIED DATA ARCHITECTURE).....	46
7.5 NORMÁLNÍ FORMY	46
7.5.1 Vazby mezi tabulkami.....	47
7.6 SQL (STRUCTURED QUERY LANGUAGE).....	48

8. CACHE SERVER PAGES (CSP)	49
8.1 SUBSTITUCE HODNOT	49
8.2 PROVÁDĚNÍ SKRIPTŮ	50
8.3 CYKLY	50
8.4 VĚTVENÍ KÓDU	50
8.5 DATABÁZOVÉ DOTAZY	50
9. REALIZACE WEBOVÉ APLIKACE.....	51
9.1 UKÁZKA WEBOVÉ APLIKACE	53
9.1.1 Menu „Hlášení“	55
9.1.2 <i>Doplňkové funkce webové aplikace</i>	58
ZÁVĚR.....	63
POUŽITÁ LITERATURA.....	64
ABECEDNÍ SEZNAM POUŽITÝCH ZKRATEK.....	66
PŘÍLOHA Č.1 - ZDROJOVÝ KÓD WEBOVÉ APLIKACE.....	68

Seznam obrázků

Obr. 5.1: Metody zabezpečení dle způsobu přenosu [9]	34
Obr. 5.2: Způsoby zabezpečení dle vlastností [9]	34
Obr. 6.1: Schéma Nemocničního Informačního Systému [13]	36
Obr. 6.2: Ukázka grafického rozhraní CareCenter	38
Obr. 6.3: Ukázka rozhraní NetAccess [13]	39
Obr. 6.4: Ukázka přístupu přes SSH pomocí programu KoalaTerm	39
Obr. 7.1: Ukázka nastavení ovladače ODBC	42
Obr. 7.2: Ukázka jednoho z možných způsobů zapojení první zmiňované varianty	43
Obr. 7.3: Položky v menu Caché kostky	44
Obr. 7.4: Caché studio	45
Obr. 7.5: Ukázka aplikace Caché SQL Manager spolu se zpracovaným SQL dotazem	46
Obr. 8.1: Ukázka zpracování jednoduchého skriptu na straně serveru	49
Obr. 9.1: E-R diagram vybraných tabulek NIS CLINICOM	51
Obr. 9.2: Návrh programového diagramu webové aplikace	52
Obr. 9.3: Přihlášení do webové aplikace	53
Obr. 9.4: Administrace uživatelů	53
Obr. 9.5: Výzva k zadání správného jména a hesla	54
Obr. 9.6: Úvodní stránka webové aplikace	54
Obr. 9.7: Položka „Hlášení“ – výběr požadovaného registru a intervalu	55
Obr. 9.8: Vyhledaná data dle zvolených parametrů	56
Obr. 9.9: Výzva ke změně-opravě zvolených parametrů	56
Obr. 9.10: Tlačítka: „Vytisknout nalezená data“ a „Zobrazit nalezené výsledky v DS 3“	56
Obr. 9.11: Autorizace a zobrazení vzorového kódu s datovým standardem verze 3.	57
Obr. 9.12: Diagnózy vybraného pacienta	57
Obr. 9.13: Ukázka výběru požadované diagnózy	58
Obr. 9.14: Ukázka výskytu zvolené diagnózy	59
Obr. 9.15: Ukázka výsledku hledání dle zadané diagnózy a vygenerování vzorového DS	59
Obr. 9.16: Grafy výskytu zvolené skupiny diagnóz	60
Obr. 9.17: Vyhledání pacienta z NIS CLINICOM dle příjmení	61
Obr. 9.18: Výběr konkrétního pacienta ze seznamu	61
Obr. 9.19: Diagnózy vybraného pacienta	61
Obr. 9.20: Hledání diagnóz dle intervalu	62
Obr. 9.21: Odhlášení uživatele z webové aplikace	62

Úvod

Vývoj v informačních technologiích velmi ovlivňuje naši každodenní činnost ve všech sférách společenského života a to prakticky ve všech směrech a ohledech. Stejně tak je tomu i ve zdravotnictví.

Mnoho firem vyrábí přístroje, které nám ve výsledku zachraňují život, a to jak přímo, tak nepřímo. Aby bylo možné zefektivnit práci laboratoří, nemocnic či samotných lékařů, je zapotřebí poskytnout všem těmto subjektům stejné normy a standardy.

Pokud by například výrobci nepoužívali stejné normy a standardy, nastal by zřejmě velký zmatek a samotná práce, včetně jejích výsledků, by nebyla efektivní.

Jedním z cílů této diplomové práce je alespoň částečně objasnit poměrně složitou strukturu datového standardu Ministerstva zdravotnictví České republiky spolu s rozšiřitelným značkovacím jazykem XML. V práci si také objasníme problematiku povinných hlášení do registrů Národního zdravotnického informačního systému (NZIS).

Dále se zaměříme na možnosti přístupu do nemocničního informačního systému (NIS) CLINICOM, včetně přístupu k datům obsažených v databázi Caché, kterou NIS CLINICOM využívá pro ukládání veškerých svých záznamů. Rovněž se budeme zabývat základními možnostmi zabezpečení serveru s NIS a datové komunikace mezi různými zdravotnickými organizacemi.

V závěru této práce si popíšeme jednotlivé aplikace z balíčku Caché firmy InterSystems, včetně návrhu a realizace webové aplikace pomocí Caché Server Pages.

Tato webová aplikace bude dle zadání diplomové práce vyhledávat a formátovat vybrané záznamy pacientů z databáze NIS CLINICOM, které poté zobrazí a umožní jejich převod do datového standardu Ministerstva zdravotnictví České republiky.

1. Datový standard (DS)

Datový standard (dále DS) vznikl za účelem přenosu dat o pacientech mezi informačními systémy zdravotnických zařízení [1].

1.1 Verze DS

Datový standard prochází stejně jako informační systémy lékařů, nemocnic a laboratoří neustálou modernizací a vývojem.

První verze DS 1.00 byla publikována v listopadu roku 1994 ve Věstníku MZ částce 8-9 pod názvem „Metodický návod MZ k datové struktuře pro přenos dat mezi informačními systémy zdravotnických zařízení, verze 1.00“ [1].

Vydání verze 1.00 byl první důležitý krok k sjednocení komunikace mezi informačními systémy zdravotnických zařízení.

Po verzi 1.00 následovala verze 1.01, v níž byla inovována datová struktura, pro potřebu dalších oborů.

Verze DS byly postupně doplňovány, ovšem základní struktura DS byla stále stejná. Do verze 1.20 pracoval DS pouze s textovými strukturami, které neumožňovaly syntaktické kontroly. Verze DS 1.20 byla podporována do 31.12.2002 [2].

Současně se však od 1.6.2002 začal používat zcela nový DS 2.01.01, který nahradil předchozí verzi 1.20.

Verze DS 2.01.01 zaznamenala oproti předchozí verzi velké změny, které se týkaly především vnitřní struktury DS. Celá struktura byla předělána do platformy celosvětově rozšířeného jazyka XML (eXtensible Markup Language) a to se všemi jeho výhodami. Současně s těmito změnami byla inovována a doplněna řada datových bloků.

Zcela zásadní změnou byla inovace v obousměrné komunikaci s laboratorním komplementem, dále pak v blocích pro práci s textovou dokumentací a nově přibyly bloky pro mikrobiologii, předávání obrazových informací atd. [3].

Obousměrná komunikace s laboratorními informačními systémy využívá národní číselník laboratorních položek (dále NČLP), který bylo nutno pro možnost použití nové verze DS rozšířit.

Nová verze NČLP 2.01.01 byla rozšířena o řadu dalších speciálních položek a současně byla u každé z položek rozšířena paleta údajů popisujících tuto položku [3].

DS verze 03.00.01 byl společně s verzí 02.05.01 NČLP vydán v červnu 2003. Finální verze DS 3.01.01, společně s verzí 02.06.01 NČLP, byla k 1.11.2003 vydána ve Věstníku MZ, částka 9, rok 2003, s termínem platnosti předmětných standardů od 1.1.2004. V této verzi DS byla zachována vnitřní struktura z předchozí verze DS. Změny se týkaly především popisu struktury bloků DS. Ty byly částečně upraveny, případně doplněny o nové položky. Oficiální vydání aktuální verze DS 03.13.01 bylo dne 1.4.2010. NČLP je neustále modernizován a doplňován o nové laboratorní položky. Aktuální verzi NČLP je 4.04.01. Vývoj DS verze 3 byl ukončen 1.1.2007, v současnosti jsou však datové bloky a číselníky stále udržovány.

DS verze 04.01.01 spolu s NČLP 2.17.01 je platný od 1.1.2007. Verze 4 vznikala v souběhu s předchozí verzí datového standardu. Hlavní změnou je nahrazení formálního

popisu pomocí DTD za popis, ve kterém je využito více XML schémat. Rovněž došlo k zavedení klinických událostí, jejich formalizaci a jednoznačnou identifikaci pomocí bloku „ku“. Oficiální vydání aktuální verze DS 04.04.01 bylo dne 1.4.2010. Verze NČLP je shodná s DS3, tedy NČLP 4.04.01.

V rámci této práce se budeme převážně věnovat stále platnému datovému standardu verze 3, z jehož struktury jednotlivých bloků vychází i verze 4 a tudíž jsou zde pouze nepatrné rozdíly.

1.2 Předpokládaná finální podoba DS

Předpokládaná finální podoba DS vzniká na základě [4]:

- požadavků Odboru informatiky Ministerstva zdravotnictví ČR
- platného datového standardu DS 1.20 a DS 2.01.01
- námětů debatovaných na konferenci DASTA
- připomínek tvůrců NČLP spolu se zástupci odborných společností laboratorních oborů s klinickou složkou
- práce autorského kolektivu DS a řady konzultantů
- praktických průběžných testů jednotlivých připravovaných modulů
- připomínek slovenských kolegů a požadavky uživatelů verze 3.01.01

Předpokládaná finální podoba vzniká po rozsáhlých debatách nad velkou řadou názorů uživatelů a různých fakultních pracovníků. Předpokládaný tvar respektuje požadavky laboratorních informačních systémů, nemocniční informační systémy i informační systémy praktických lékařů, dále pak respektuje i požadavky dalších informačních systémů jako jsou například výukové programy a jiné [4].

2. Struktura bloků a souborů datového standardu

Definice každého bloku DS je dána jeho textovým popisem ve formě tabulek a poznámek, dále pak jeho přepisem do tvaru DTD (Definice Typu Dokumentu) a příkladem použití tohoto bloku.

Jelikož textovým popisem nelze přesně vyjádřit strukturu DTD a současně ve struktuře DTD nelze přesně popsat vše potřebné pro definici bloku, je tedy využíván textový popis spolu s DTD.

Datový standard verze 4 již využívá, místo formálního textového popisu, rozdělení do více jednotlivých XML schémat.

2.1 Struktura bloku DS

Vzor struktury bloku v DS [4]:

blok - označení bloku

Základní informace o bloku.

{stav}

kód	T	D	V	plný název	hodnota	podmínky, pokyny, poznámky	změny
aaa	a	000	?	název určující obsah	„výčet hodnot“	volný text	
zzz	e		*		TAB A		

2.2 Popis bloku DS

V bloku se nejprve uvádí označení bloku, které pojmenovává daný element XML. Pro pojmenování bloku se používají malá písmena bez diakritiky.

Označení „blok“ se používá pro přehlednost s ohledem na předchozí verze DS, pro XML není toto označení podstatné. V textových dokumentech se pro snazší vyhledávání dává před označení „blok“ ještě znak „*“, který tak usnadní rozlišení bloku od shodných pojmenování.

Po označení bloku následují základní informace o bloku, které stručně informují o obsahu daného bloku. Tyto informace mohou být i na několik řádků a mohou obsahovat fakultativní informace o původním označení bloku v DS 1.20 [4].

Dále následuje stavová hodnota bloku, ta udává, zdali je blok rozpracován, neaktuální atd. V distribuční síti budou bloky označené jako {distribuováno od verze} a také bloky s poznámkou {OBSOLETNÍ!}. Tento stav nás informuje o tom, že tento blok již nemáme používat, protože byl navržen na zrušení a bude tedy po určité době neaktuální.

Ve vývoji se používá také označení stavu {rozpracováno} a mnoho dalších označení [4].

Jeden z nejvýznamnějších prvků bloku je tabulka, která má pevně definované sloupce, šířka sloupců je logicky variabilní v závislosti na obsahu buňky.

Celá tabulka je navržena tak, aby byla přehledná a logická.

První buňka v tabulce je označena jako kód. Zde vkládáme identifikátor pro potřeby jazyka XML. Identifikátor má tvar malých písmen bez diakritiky. Odkazy na jiné bloky jsou v hypertextové formě skutečně aktivní odkazy na příslušné bloky.

Další buňkou je typ pro XML, v tabulce označený jako „T“.

Typ může obsahovat hodnoty: a, e, d.

Hodnota „a“ (atribut) nám říká, že datový obsah je obsahem atributu bloku, představující popisovaný blok. Hodnota „e“ (element) nám říká, že datový obsah je obsahem jednoduchého bloku nebo jde o vnořenou strukturu dalších bloků.

Poslední hodnotou typu je „d“ (data) v DTD symbol „#PCDATA“. Tato hodnota nám říká, že údaj je přímo obsahem bloku, představující popisovaný datový blok. Tento blok pak neobsahuje vnořené struktury, ale může obsahovat atributy [4].

Třetí buňkou v tabulce je délka položky „D“, která je pro potřebu databází informačních systémů. Délka položky může obsahovat přímo číslo, které udává pevnou délku položky nebo má před číslem znak „-“. V takovém případě nesmí být délka položky větší než toto číslo. Pokud není uveden číselný údaj o délce, jedná se pak o element nebo o atribut s libovolnou délkou [4].

Další, čtvrtá, buňka obsahuje výskyt „V“, tato položka je určena pro potřeby XML. Její výchozí hodnota je 1 (vyskytuje se jen 1x). Další možnou hodnotou je znak „+“ (vyskytuje se alespoň 1x), znaky „*“ (může se vyskytnout maximálně 1x) a „?“ (může se vyskytovat opakovaně), znaky „*“ a „?“ jsou nepovinné.

Pátá buňka obsahuje plný název, tím se myslí plný název položky, případně i její stručné charakteristiky.

V buňce „hodnota“ je mnoho variant obsahu. Buňka nemusí být vyplněna, nebo může obsahovat odvolání na tabulku hodnot uvedenou níže pod hlavní tabulkou. Taková tabulka se pak váže jen k tomuto bloku. Dále může buňka „hodnota“ obsahovat odvolání na číselník interní nebo externí a jiné údaje.

Další buňka obsahuje podmínky, pokyny, poznámky. Tato buňka může obsahovat další údaje psané volným textem nebo hypertextovými odkazy. Vždy musí být uvedeno, jestli se jedná o podmínku, pokyn, výklad nebo poznámku.

Poslední položka v tabulce je vyhrazena pro změny, které se uvádí rovnou nebo dojde-li k více změnám, uvádí se odkazem.

2.3 Znakové sady v souborech DS

Velmi důležité je dodržování znakových sad v DS.

Podporované znakové sady jsou [4]:

- utf-8 (unicode transformation 8-bit)
- IBM852 (PC Latin 2) = *cp852*
- ISO-8859-2 (ISO Latin 2)
- Windows-1250 (MS Windows)

Systémy, které pracují s DS by měly podporovat tyto znakové sady také.

Je však na subjektech, které vyrábí tyto systémy, zda tyto znakové sady budou podporovat nebo budou podporovat jen některé z nich.

Důležitý je správný zápis desetinné části čísel, pro ty se v DS používá desetinná čárka „ , “.

2.4 Struktura souboru DS

Záhlaví datového souboru, pro DS 3.01.01 a výše, vyplývá ze specifikace XML.

Musíme v něm tedy dodržovat malá a velká písmena.

První řádek XML souboru obsahuje deklaraci XML s příslušnou verzí a kódováním například: `<?xml version='1.0' encoding='iso-8859-2' ?>`

Druhý řádek XML souboru se odvolává na DTD řetězcem „`<!DOCTYPE`“, pak následuje mezer a základová značka následujícího dokumentu. V případě DS je tato značka „`dasta`“. Dále pak pro odlišení veřejné definice od systémové se volí slovo „`SYSTEM`“ a jako poslední je uvedena část DS například „`ds020101.dtd`“. Z tohoto zápisu je pak patrné, s kterou verzí DS pracoval odesílatel souboru.

Další řádky v XML souboru jsou dány zněním DS verze 2.01.01 a výše [4].

Následující řádek XML souboru obsahuje hlavní blok „`dasta`“, jehož obsahem jsou povinné a volitelné kódy. Mezi povinné patří například kódy [5]:

`id_soubor` – Tento kód slouží k jednoznačné vnitřní identifikaci souboru v rámci firmy a jejího programu nebo informačního systému. Každá firma je povinna zajistit jednoznačnost tohoto řetězce pro všechny svoje aplikace, které soubory vytvářejí.

To znamená, že není možné, aby dva různé nebo stejné programy u různých uživatelů tvořily soubor se shodnou identifikací. Identifikační řetězec musí být bez mezer a musí začínat osmiznakovým kódem firmy, která soubor generuje.

Za tímto řetězcem se zpravidla uvádí kód programu, verze, licenční číslo, určení souboru, typ odesílajícího místa, označení souboru, datum a čas vzniku ve formátu „`YYYY-MM-DDTHH:MM:SS`“.

Noví firemní uživatelé DS se musí registrovat u správce DS.

Provizorně se může, pouze pro testování a ladění, použít neadresní kód, který má na první pozici znak „`_`“, za kterým následuje sedm písmen.

`verze_ds` , `verze_nclp` – Kód verze DS a NČLP nás informuje, že je soubor vytvořený na základě dané verze, například pro DS „`2.01.01`“.

ur – Tímto důležitým kódem určujeme typ přenášených dat. V případě patientských dat určujeme též urgentnost sdělení nebo zpracování souboru.

Hodnota : „R“ - patientská data pro rutinní zpracování
„S“ - rovněž patientská data, která mají být zpracována
urgentně čili dříve než data s označením „R“
„U“ - výkazy a zprávy pro ÚZIS (Národní zdravotnický IS) ČR
„V“ - soubor vykazovaných výkonů
„B“ - laboratorní bloky dat
„C“ - číselníky
„H“ - hlášení zpráv z oblasti hygieny a epidemiologie
„T“ - technická testovací data

potvrzeni – Pokud má být potvrzen příjem souboru, nabývá tento kód hodnoty
„P“ – potvrdit, že soubor byl doručen a přijat, „N“ – nepotvrzovat.
Pokud kód potvrzení nevyplníme, bude automaticky dosazen hodnotou „N“.

Za kódem „potvrzeni“ je hlavní blok „dasta“ ukončen a následují bloky [5]:

zdroj_is – Tento blok obsahuje kódy: kod_firmy, kod_prog, verze_prog.
V těchto kódech je uložena informace o firmě a jejím programu nebo
informačním systému, kterým byl soubor vytvořen.

pm – V bloku „pm“ nalezneme pouze jeden kód, který určuje příjmové místo.
Tím myslíme místo zasílaného souboru. Zde můžeme volit jeden kód
z následujících kódů: ico, icz, icp, icl, oddel, pcz.
Dále jsou v tomto bloku doplňující bloky:

„as“ – adresní spojení
„a“ – adresa příjemce, jejíž typ je „P“

is – Tento blok je jeden z nejpodstatnějších v rámci našeho souboru.
Obsahuje základní informace o odesílateli zasílaného souboru a data pacienta/tů.
Kódy v tomto bloku jsou: ico, icz, icp, icl, pcz, oddel, oavl.
Doplňující bloky bloku „is“ jsou:

as – adresa spojení, doplňující blok adres „a“ (kódy: typ, obsah, atd.)
a – adresy vázané k odesílateli, příjemci, pacientovi i pro další účely
ip – pacient (pacientů může být i více, data pro jednoho pacienta
mohou být zasílána i ve více blocích „ip“)
idu – data pro ÚZIS (Národní zdravotnický informační systém) ČR
ivv – vykázané výkony a podklady pro MIS a jiné expertní programy
ilb – laboratorní bloky dat
icl - číselníky pro Informační Systém
ihe - hlášení a sestavy pro hygienu a epidemiologii
text – technická (testovací) data (obsahuje například kód „autor“)

Kódy doplňujícího bloku „ip“ z bloku „is“ jsou:

id_pac – identifikace pacienta v IS odesílatele
rodcis – rodné číslo, znakově v délce 9 nebo 10 znaků
jmeno, prijmeni, titul_pred, titul_zo – jméno, příjmení, titul před-za
sex – pohlaví, nabývá hodnoty „M“ muži, „F“ ženy, „X“ blíže neurčeno
rod_prijm, jine_idu – rodné příjmení a jiné identifikační údaje

Dále obsahuje blok „ip“ doplňující bloky (pár příkladů – neúplný výčet):

dat_dn – datum a čas narození formalizované (kódy: dat_xx, dat_du)
dat_de – datum a čas úmrtí formalizované (kódy: dat_xx, dat_du)
ipi_o, ipi_v – identifikační údaje odeslané/vracené
a – adresy vázané k pacientovi
h – výška a hmotnost standardní (kódy: vyska, hmotnost)
pv – platební vztah pacienta („kdo hraď“) pojišťovna, přímý plátce
p – údaje o zdravotní pojišťovně (pojišťovnách)
u – urgentní informace o pacientovi (kódy: ua, uks-krevní skupina)
oc – očkování (kódy: garant_dat, ocz, dat_ak, dat_ab)
dg – diagnózy trvalé a přechodné (kódy: dgz-diagnózy, dat_ab)
le – podávané léky (kódy: lez, dat_ab – datum a čas aktualizace bloku)
pn – pracovní neschopnosti (kódy: pnz, dat_ab)
lo – informace o vzorcích, objednávka vyšetření a doplňující údaje

2.5 Název souboru DS

Konstrukce názvu výsledného DS vychází z prostředí DOS, jelikož UNIX a ostatní systémy podporují více možností, které by DOS nepodporoval. Konstrukce názvu souboru DS je podle pravidel, které byly zavedeny již ve verzi DS 1.20 [4].

Při tvorbě jména souboru DS je počítáno s volným šířením, a tudíž je potřeba, aby byl soubor s DS patřičně odlišný. Soubor DS má v nezkomprimovaném tvaru koncovku xml, pokud je soubor zkomprimován, může mít koncovku dle autora komprimačního programu. Nejznámější koncovky komprimačních programů jsou zip, rar, arj, arc, a jiné.

2.6 Ukázka obsahu souboru DS

Ukázka obsahu souboru DS verze 3 se zprávou o rodiče:

```
<?xml version='1.0' encoding='Windows-1250' ?>
<!DOCTYPE dasta SYSTEM "ds030701.dtd" >
<dasta id_soubor="XYZ*****_03_00002_2004-04-28T12:55:09"
verze_ds="03.13.01" verze_nclp="02.28.01" bin_priloha="T" ur="T"
typ_odesm="NN" ozn_soub="00002" dat_vb="2008-06-13T11:51:09">
<zdroj_is kod_firmy="STAPRO__" kod_prog="NISED" verze_prog="0.1"
liccis_prog="3"/>
<pm ico="00023833">
<as typ="I">
<vnitrni>NZIS</vnitrni>
</as>
<a typ="P">
<jmeno>ÚSTAV ZDRAVOTNICKÝCH INFORMACÍ A STATISTIKY ČR</jmeno>
<adr>NZIS</adr>
<mesto>Praha</mesto>
</a>
</pm>
<is ico="22345678">
<as typ="I">
```

```

<vnitrni>805</vnitrni>
</as>
<a typ="0">
<jmeno>Nemocnice YY</jmeno>
<adr>Oddělení informatiky</adr>
<dop1>Budova 5/45</dop1>
<mesto>Město 2</mesto>
</a>
<idu>
<nr>
<nrr rico="64124142" rpcz="000" rodd="18567">
<nrrrod rcispor="5270" rrcm="6352301384" rbydlm="623" robecm="545058"
rorp="7210" rstaobc="1" rzp="213" rprij="2002-07-16T11:00" rstav="2"
rvzdel="3" rcelpor="02" rpredpor="0" rscpor="0" rmrtve="0" rcnu="0"
rpnu="0" rsampot="0" rupt="0" rmimo="0" rprenat="07" rkontr="17"
rhosp="0" rhostyd="00" rprirus="19" rkour="0" ralkoh="0" rdroy="0"
rultr1="13" rultr2="36" rdiab1="0" rdiab2="0" rdiab3="0" rdiab4="0"
rdiab5="0" rdiab6="0" rteh1="0" rteh2="0" rteh3="0" rteh4="0"
rteh5="0" rteh6="0" rteh7="0" rteh8="0" rteh9="0" rteh10="0"
rteh11="0" rteh12="0" rpred="2002-07-22" rodhad="1" rdatpor="2002-
07-18T14:00" rodtok="2002-07-18T10:00" rcetteh="1" rgesta="39"
rdgind1="048 " rriziko1="0" rriziko2="0" rriziko3="0" rriziko4="0"
rctg1="1" rctg2="1" ranes="2" rporod1="0" rporod2="0" rporod3="0"
rporod4="0" rporod5="0" rporod6="0" rporod7="0" rporod8="0"
rleky1="0" rleky2="0" rleky3="0" rleky4="0" rleky5="0" rleky6="0"
rleky7="0" rvedl="2" rzhodn="1" rdatuk="2002-07-23T12:00"
rduvuk="1">
<nrrrodp rpord="0" rplod="1" rvagin="1"/>
<nrrrodn rpord="0" rpohl="1" rvit="1" rhmot="2950" rapgar1="09"
rapgar5="10" rapgar10="10" rph="735" rstavd="1"/>
</nrrrod>
</nrr>
</nr>
</idu>
</is>

```

Ukázka obsahu souboru DS verze 4 se zprávou o rodiče:

```

<?xml version='1.0' encoding='Windows-1250' ?>
<ds:dasta xsi:schemaLocation="urn:cz-mzcr:ns:dasta:ds4:ds_dasta
http://ciselniky.dasta.mzcr.cz/xmlschema/ds_dasta-4.01.02.xsd"
id_soubor="XYZ*****_03_00002_2004-04-28T12:55:09"
verze_ds="04.04.01" verze_nclp="02.28.01" bin_priloha="T" ur="T"
typ_odesm="NN" ozn_soub="00002" dat_vb="2008-06-13T11:51:09">
<ds:zdroj_is kod_firmy="STAPRO__" kod_prog="NISED" verze_prog="0.1"
liccis_prog="3"/>
<ds:pm ico="00023833">
<ds:as typ="I">
<ds:vnitrni>NZIS</ds:vnitrni>
</ds:as>
<ds:a typ="P">
<ds:jmeno>ÚSTAV ZDRAVOTNICKÝCH INFORMACÍ A STATISTIKY ČR</ds:jmeno>
<ds:adr>NZIS</ds:adr>
<ds:mesto>Praha</ds:mesto>

```

```

</ds:a>
</ds:pm>
<ds:is ico="22345678">
<ds:as typ="I">
<ds:vnitrni>805</ds:vnitrni>
</ds:as>
<ds:a typ="0">
<ds:jmeno>Nemocnice YY</ds:jmeno>
<ds:adr>Oddělení informatiky</ds:adr>
<ds:dop1>Budova 5/45</ds:dop1>
<ds:mesto>Město 2</ds:mesto>
</ds:a>
<dsidu:idu xsi:schemaLocation="urn:cz-mzcr:ns:dasta:ds4:ds_idu
http://ciselniky.dasta.mzcr.cz/xmlschema/ds_idu-4.04.01.xsd">
<dsidu:nr>
<dsidu:nrr rico="64124142" rpcz="000" rodd="18567">
<dsidu:nrrod rcispor="5270" rrcm="6352301384" rbydlm="623"
robecm="545058" rorp="7210" rstaobc="1" rzp="213" rprij="2002-07-
16T11:00" rstav="2" rvzdel="3" rcelpor="02" rpredpor="0" rscpor="0"
rmrtve="0" rcnu="0" rpnu="0" rsampot="0" rupt="0" rmimo="0"
rprenat="7" rkontr="17" rhosp="0" rhostyd="00" rprirus="19"
rkour="0" ralkoh="0" rdrogy="0" rultr1="13" rultr2="36" rdiab1="0"
rdiab2="0" rdiab3="0" rdiab4="0" rdiab5="0" rdiab6="0" rteh1="0"
rteh2="0" rteh3="0" rteh4="0" rteh5="0" rteh6="0" rteh7="0"
rteh8="0" rteh9="0" rteh10="0" rteh11="0" rteh12="0" rpred="2002-07-
22" rodhad="1" rdatpor="2002-07-18T14:00" rodtok="2002-07-18T10:00"
rcetteh="1" rgesta="39" rdgind1="048 " rriziko1="0" rriziko2="0"
rriziko3="0" rriziko4="0" rctg1="1" rctg2="1" ranes="2" rporod1="0"
rporod2="0" rporod3="0" rporod4="0" rporod5="0" rporod6="0"
rporod7="0" rporod8="0" rleky1="0" rleky2="0" rleky3="0" rleky4="0"
rleky5="0" rleky6="0" rleky7="0" rvedl="2" rzhodn="1" rdatuk="2002-
07-23T13:33" rduvuk="1">
<dsidu:nrrod rpord="0" rplod="1" rvagin="1"/>
<dsidu:nrrodn rpord="0" rpohl="1" rvit="1" rhmot="2950" rapgar1="09"
rapgar5="10" rapgar10="10" rph="735" rstavd="1"/>
</dsidu:nrrod>
</dsidu:nrr>
</dsidu:nr>
</dsidu:idu>
</ds:is>
</ds:dasta>

```

Z předchozích příkladů je zřejmé, že obsah XML souborů má poměrně složitou a rozvětvenou strukturu. Rovněž je znatelná podobnost DS verze 3 a 4.

3. Rozšířitelný značkovací jazyk (XML)

Jazyk XML (z anglického eXtensible Markup Language) byl vybrán pro účely DS MZČR jako nejvhodnější spolu s DTD (Document Type Definition). Spojením XML spolu s DTD vznikne takzvané „XML schéma“.

V tomto spojení platí syntaxe XML spolu s elementy a atributy z DTD.

3.1 Historie značkovacích jazyků (markup language)

Již od svého počátku, pomineme-li vědecké a armádní výpočty, byla výpočetní technika využívána pro přípravu a publikování textu. Situace však byla dosti odlišná od té dnešní.

Na počítačích se připravovaly dokumenty pro profesionální tisk knih, časopisů atd. Výsledek se pomocí osvitové jednotky přenesl na film, z kterého pak tiskárny vyráběly knihy, časopisy atd. Jelikož konkurence byla velká, snažila se každá firma vyvinout svůj vlastní „lepší“ jazyk pro ovládání jednotky. Dokumenty pro sazbu se tedy připravovaly tak, že se do textu vepisovaly speciální řídicí sekvence pro ovládání dané osvitové jednotky. To mělo za následek, že jednou vytvořený dokument byl svázan s výstupním zařízením konkrétního výrobce. Jeho převod pro použití na jiné „konkurenční“ osvitové jednotce byl téměř nemožný.

Tento stav nebyl ideální, a proto vzniklo hned několik systémů, které problém nekompatibility různých výstupních zařízení řešily. Princip spočíval v tom, že se v dokumentu používaly obecné příkazy, které se pak pomocí speciálních konvertorů převedly do jazyka srozumitelného pro konkrétní zařízení. Pokud, jsme tedy chtěli dokument vytisknout na jiném zařízení, než pro které byl vytvořen, stačilo sehnat příslušný konvertor. Samotný dokument tedy zůstal stejný a nemusel se měnit [6].

Mezi nejrozšířenější programy (konvertory) patřily troff a TeX. Důležité je, že oba dva jazyky byly čistě prezentační – dalo se pomocí nich určit, jak se mají jednotlivé části textu formátovat [6].

Tyto programy se však hodí pouze pro zpracování dokumentů, které se mají ve výsledku tisknout. Hlavně kvůli tomu, že nabízejí příkazy, umožňující měnit druh použitého písma, způsob zarovnání a velké množství dalších parametrů, ale neumožňují přímo označit logickou strukturu dokumentu. S neustálým vývojem informačních technologií totiž vznikla potřeba jedny a tytéž informace prezentovat více způsoby, a to jak v elektronické podobě, tak v podobě knižní. Pro tyto účely je však potřebné znát logickou strukturu dokumentu, ve kterém musíme rozpoznat nadpis od popisu obrázků atd.

Potřebujeme tedy jazyk, jenž umožní označit význam jednotlivých částí textu, a ne jejich vzhled. Takovýmto jazykům, které umožňují vyznačovat části textu, se říká značkovací jazyky (markup language).

Jeden z prvních značkovacích jazyků byl GML (Generalized Markup Language), který vytvořili Goldfarb, Mosher a Lorie při práci na systému pro uchovávání a následné využití právních textů pro IBM. Museli se tehdy vypořádat s nekompatibilitou jednotlivých systémů a programů a nejsnazší cesta vedla právě přes vytvoření obecného značkovacího jazyka [6].

Na základě GML začala vyvíjet standardizační organizace ANSI jazyk, který umožňoval definici vlastních značkovacích jazyků. Výsledkem byl jazyk SGML (Standard Generalized Markup Language). Jazyk SGML je hodně obecný a také hodně složitý – umožňoval definici vlastních značkovacích jazyků (sad značek a jejich vzájemných vztahů) pomocí definic typu dokumentu (DTD) [6].

Asi nejznámější aplikací SGML je jazyk HTML (Hypertext Markup Language), který se používá pro tvorbu webových stránek. To, jaké značky můžeme na stránkách používat, určuje příslušné DTD, které je pro každou verzi HTML mírně odlišné. Jazyk HTML si získal velkou oblibu díky své jednoduchosti. Ukázalo se však, že pevně daná skupina značek, které HTML používá, už nestačí. Pro účely vyhledávání a vůbec efektivnější výměny dat by bylo lepší mít možnost používat vlastních značek, které by přesně vymezily význam textu [6].

Tyto požadavky by tedy splňoval opět standard SGML, ale jelikož je hodně složitý a byla by z něj využita opět jenom část jeho možností, byl proto „navržen“ nový jazyk.

Jazyk dostal jméno XML (eXtensible Markup Language). Jedná se o podmnožinu SGML, která si zachovává možnost definování vlastních DTD, tedy značek, pro jednotlivé skupiny dokumentů [7].

3.2 Kontrola jazyka XML

XML nám umožňuje definovat si pomocí DTD vlastní sadu značek, které chceme v dokumentu používat. DTD se tedy hodí pro popis formátů, reprezentujících především textové dokumenty. Neobsahuje však přímo nástroje na kontrolu různých typů dat jako čísla, údaje o datu a čase. To je přitom velice důležité především pro aplikace, které si pomocí XML posílají data databázového charakteru. Tato data musíme tedy kontrolovat dodatečně.

3.3 Shrnutí jazyka XML

I když je jazyk XML jednodušší nežli SGML, není příliš jednoduché se v něm plně orientovat, hlavně díky DTD. Proto nám jsou k dispozici programy, umožňující snazší prohlížení a kontrolu správnosti kódu (validační programy). Tyto programy by měly být volně k dispozici na CD-ROMu spolu s DS.

Jsou volně šiřitelné, proto by neměl být problém v jejich rozšíření mezi zákazníky laboratoří a ostatní subjekty, které DS využívají [3].

Na CD-ROMu s DS 3 je validační program v adresáři „ds_validator“ ve verzi 1.12.

Práce s ním je rozsáhle popsána v příložené dokumentaci.

Jelikož na CD-ROMu s DS 4 nejsou žádné nástroje pro validaci zařazeny, doporučují se následující programy: XSV a Topologi Validator, které lze bezplatně stáhnout na adrese: <http://www.ltg.ed.ac.uk/~ht/xsv-status.html> a <http://www.w3.org/XML/Schema>.

4. Národní zdravotní registry

Národní zdravotní registry (NZR) jsou vedeny Národním zdravotnickým informačním systémem (NZIS). Ten je kromě vedení NZR určen ke sběru a zpracování zdravotnických údajů a informací, k poskytování informací v rozsahu určeném právními předpisy, respektování podmínek ochrany dat a využití informací v rámci zdravotnického výzkumu [8].

4.1 Údaje a legislativa o povinných hlášení do NZR

Národní zdravotní registry jsou naplňovány i údaji o výskytu vybraných společensky závažných onemocnění a stavů. Údaje o těchto onemocněních jsou součástí zdravotnické dokumentace.

Tyto informace jsou v registrech zpracovány, i bez souhlasu jednotlivých subjektů, ve smyslu novely zákona o péči o zdraví lidu č.156/2004 Sb. Jedná se zde především o následující registry: Národní onkologický registr, Národní registr hospitalizovaných, Národní registr rodiček, Národní registr novorozenců, Národní registr vrozených vad, Registr lékařů, zubních lékařů a farmaceutů, Národní registr potratů, Národní registr cévní chirurgie, Národní kardiochirurgický registr, Národní registr kloubních náhrad, Národní registr nemocí z povolání, Národní registr kardiiovaskulárních intervencí a Národní registr uživatelů lékařsky indikovaných substitučních látek.

Dále vyhláška 552/2004 Sb., o předávání osobních a dalších údajů do Národního zdravotnického informačního systému pro potřeby vedení národních zdravotních registrů, vymezuje zdravotnická zařízení, jež mají povinnost předávat požadované údaje do NZIS. Z těchto údajů jsou pak vedeny jednotlivé NZR. Vyhláška rovněž stanovuje jednotlivé lhůty, v kterých musí zdravotnické zařízení předat požadované informace o subjektech. Ty se předávají v tištěné nebo elektronické podobě. V současnosti je před tištěnou podobou upřednostněna podoba elektronická ve formátu DS MZ ČR, ta umožňuje účinně zefektivnit zadávání informací do NZIS. Hlavním urychlením, je absence nutnosti přepisování jednotlivých dat.

Do NZIS jsou rovněž předávány údaje i z informačních systémů orgánů ochrany veřejného zdraví [19] (Registr tuberkulózy, Registr pohlavních nemocí a informační systém Infekční nemoci), které vychází ze zákona č. 258/2000 Sb., o ochraně veřejného zdraví a o změně některých souvisejících zákonů, ve znění pozdějších předpisů a prováděcí vyhlášky č. 195/2005 Sb., kterou se upravují podmínky předcházení vzniku a šíření infekčních onemocnění a hygienické požadavky na provoz zdravotnických zařízení a ústavů sociální péče [8].

4.2 Účel a legislativa jednotlivých zdravotních registrů

Samotná legislativa Národních zdravotních registrů vychází z legislativy Národního zdravotnického informačního systému (vyhláška 552/2004 Sb.), jehož je NZR součástí. Dále je pak rozšířena-upravena zákonem, případně vyhláškou, dle jednotlivých registrů.

Národní zdravotní registry dělíme na [8]:

Národní registr hospitalizovaných (NRHOSP):

Účelem je zjišťování požadovaných údajů, získání zdroje informací o zdravotním stavu populace, které jsou důležitým nástrojem pro řízení zdravotnictví a stanovení koncepce a realizace zdravotní politiky státu, potřebné k definování optimální sítě lůžkových zdravotnických zařízení.

Správce a současně zpracovatelem je Ústav zdravotnických informací a statistiky ČR.

Do NRHOSP patří péče, trvající déle než 24 hodin. Nepatří sem tedy ambulantní péče, i v případech, kdy se jednalo o operační výkon, po kterém byl pacient propuštěn do domácí péče – tzv. jednodenní chirurgie.

Hlášení podává každé zdravotnické zařízení, jenž poskytuje ústavní zdravotní péči, vyjma lázeňských léčebeň.

Samotné hlášení do NZIS je v současné době možné předat elektronickou formou v podobě txt souboru, výjimečně lze předat i pomocí vyplněného formuláře.

Termín předání je v listinné podobě nejpozději do desátého kalendářního dne následujícího měsíce. V elektronické podobě lze hlášení odevzdat nejpozději do konce následujícího měsíce. Hlášení v listinné podobě se odevzdává dříve, z důvodu zpracování-převedení formulářů do elektronické podoby.

Legislativa platná pro NRHOSP:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů, ve znění pozdějších předpisů.

Národní registr rodiček (NRROD) :

Účelem zjišťování požadovaných údajů je zajištění základních údajů o reprodukční anamnéze ženy, o průběhu jejího těhotenství, porodu a o novorozenci. Sledování rodiček slouží k hodnocení zdravotního stavu rodičky z pohledu kvality péče. Získané informace jsou cenným zdrojem informací pro gynekologicko-porodnickou péči a jsou důležitým nástrojem pro zlepšování péče o těhotné a rodičky.

Hlášení podléhají všechny rodičky, které porodily v České republice, a podává je gynekologicko-porodnické oddělení, kde byla žena do posledního dne šestinedělí, nebo po porodu, ošetřena. Pokud dojde k porodu mimo zdravotnické zařízení (doma, v dopravním prostředku atd.) má oznamovací povinnost zdravotnický pracovník, jenž byl u porodu, případně provedl první poporodní ošetření rodičky a první poporodní ošetření novorozence.

Hlášení - zpráva o rodičce, je podáváno v elektronické podobě, buď v datovém rozhraní ve formátu xml, nebo txt. Výjimečně lze hlášení předat v listinné podobě na formuláři „Zpráva o rodičce“. Ke každému hlášení musí být rovněž podáno hlášení o novorozenci v počtu, který odpovídá četnosti těhotenství. Jedinou výjimkou je, vícečetné těhotenství s mrtvě narozeným plodem s hmotností nižší než 1kg. V tomto případě se vyplňuje zpráva o novorozenci pouze za narozené s hmotností nad 1kg.

Termín předání je v listinné podobě nejpozději do desátého kalendářního dne následujícího měsíce. V elektronické podobě lze hlášení odevzdat nejpozději do konce

následujícího měsíce. Hlášení v listinné podobě se odevzdává dříve z důvodu zpracování-převedení formulářů do elektronické podoby.

Legislativa platná pro NRROD:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů, ve znění pozdějších předpisů.

Národní registr novorozenců (NRNAR):

Účelem zjišťování požadovaných údajů je zajištění nezbytných informací z oblasti perinatální péče jak pro potřeby odborných zdravotnických pracovníků, Ministerstva zdravotnictví, tak pro mezinárodní vykazování údajů. Získané informace jsou důležitým zdrojem hodnocení zdravotního stavu novorozenců a využívají se pro řízení, hodnocení a zlepšování péče o novorozence. NRNAR obsahuje základní údaje o okamžitém stavu novorozence po porodu, jeho další zdravotní stav, komplikace, léčbu.

Hlášení o novorozenci se hlásí za všechny živé novorozence bez ohledu na délku gestace a porodní hmotnost. U mrtvě narozených s porodní hmotností ≥ 1 kg nebo gestačním věkem ≥ 28 týdnů. U vícečetných těhotenství se posuzuje každý novorozenec zvlášť. Každý novorozenec se tedy hlásí jako samostatná věta datového rozhraní.

Hlášení podléhají novorozenecké oddělení, na kterých bylo dítě hospitalizováno do 3 měsíců svého života. Pokud je dítě porozeno mimo zdravotnické zařízení, musí oznamovací povinnost vykonat zdravotnický pracovník, který byl při porodu nebo provedl první poporodní ošetření rodičky a první poporodní ošetření novorozence.

Hlášení – zpráva o novorozenci, je podáváno v elektronické podobě, buď v datovém rozhraní ve formátu xml, nebo txt. Výjimečně lze hlášení předat v listinné podobě a to na formuláři „Zpráva o novorozenci“. Ke každému hlášení musí být rovněž podáno hlášení o rodičce. U mrtvě narozeného dítěte vyplňuje zprávu o novorozenci lékař, který provedl porod.

Termín předání je v listinné podobě nejpozději do desátého kalendářního dne následujícího měsíce. V elektronické podobě lze hlášení odevzdat nejpozději do konce následujícího měsíce. Hlášení v listinné podobě se odevzdává dříve z důvodu zpracování-převedení formulářů do elektronické podoby.

Legislativa platná pro NRNAR:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů, ve znění pozdějších předpisů.

Národní registr vrozených vad (NRVV):

Účelem zjišťování požadovaných údajů je registrace prenatálně a postnatálně diagnostikovaných vrozených vad v populaci, která je v současné době jedním

ze základních faktorů potřebných pro hodnocení zdravotního stavu populace a je nedílnou součástí hodnocení prenatalní, perinatální a postnatální péče. Sledování výskytu vrozených vad slouží k vyhodnocování včasného záchytu vrozených vad. Získané informace se využívají k hodnocení zdravotního stavu a kvality nové populace.

Hlášení podává každý odborný lékař, jenž vrozenou vadu u plodu nebo u dítěte do 15 ti let diagnostikuje. Sledují se tedy vrozené vady u dětí do 15. roku života, u mrtvě narozených dětí a u samovolných potratů nad 0.5 kg.

Hlášení – zpráva o vrozené vadě plodu nebo dítěte, je podáváno v elektronické podobě, buď v datovém rozhraní ve formátu xml, nebo txt. Výjimečně lze hlášení předat v listinné podobě na formuláři „Vrozená vada plodu nebo dítěte“.

Termín předání je v listinné podobě nejpozději do 10.tého kalendářního dne následujícího měsíce. V elektronické podobě lze hlášení odevzdat nejpozději do konce následujícího měsíce. Hlášení v listinné podobě se odevzdává dříve z důvodu zpracování-převedení formulářů do elektronické podoby.

Legislativa platná pro NRVV:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů, ve znění pozdějších předpisů.

Národní registr potratů (NRPOT):

Účelem zjišťování požadovaných údajů je zajištění údajů pro posouzení kvality péče o reprodukční zdraví. Sběr údajů týkající se potratů v České republice se stal dlouholetou tradicí a nezbytnou součástí demografických a perinatologických informací o české populaci. Anonymizované údaje jsou v měsíční periodicitě předávány Českému statistickému úřadu pro potřeby demografické statistiky.

Analýza dat poskytuje řadu mezinárodně uznávaných kritérií kvality péče a kvality zdraví a poskytuje nezbytný komplement k ostatním perinatologickým údajům, bez kterého není možné komplexně posoudit kvalitu péče o reprodukční zdraví.

Hlášení je podáváno při všech druzích potratů, přičemž potratem se rozumí ukončení těhotenství ženy pokud, plod neprojevuje žádnou známku života a jeho hmotnost je nižší než 1kg, plod projevuje alespoň jednu ze známek života a má porodní hmotnost nižší než 0.5kg, ale nepřežije 24 hodin po porodu, nebo pokud z dělohy ženy bylo vyňato plodové vejce bez plodu, anebo těhotenská sliznice. Rovněž se za potrat považuje ukončení mimoděložního těhotenství, nebo přerušení těhotenství, které je provedeno dle zvláštních předpisů.

Umělé přerušení těhotenství lze provést pouze, pokud doba těhotenství nepřesahuje 12 týdnů, v případě genetických důvodů je možné potrat uskutečnit do 24 týdnů těhotenství.

Do sledované skupiny patří ženy žijící na území ČR – cizinky i Češky.

Ohlašovací povinnost má každé gynekologické oddělení, pokud však k potratu došlo mimo zdravotnické zařízení, pak podává hlášení lékař, jenž byl potratu přítomen nebo ženu dodatečně ošetřil.

Hlášení – zpráva o umělé přerušení těhotenství, hlášení potratu a mimoděložního

těhotenství, je podáváno v elektronické podobě, buď v datovém rozhraní ve formátu xml, nebo txt. Výjimečně lze hlášení předat v listinné podobě na formuláři „Žádost o umělé přerušení těhotenství (UPT), hlášení potratu a mimoděložního těhotenství“.

Termín předání je v listinné podobě nejpozději do desátého kalendářního dne následujícího měsíce. V elektronické podobě lze hlášení odevzdat nejpozději do konce následujícího měsíce. Hlášení v listinné podobě se odevzdává dříve z důvodu zpracování-převedení formulářů do elektronické podoby.

Legislativa platná pro NRPO:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Zákon ČNR č. 66/1986 Sb., o umělém přerušení těhotenství.

Vyhláška MZ ČSR č. 75/1986 Sb., kterou se provádí zákon ČNR č.66/1986 Sb., o umělém přerušení těhotenství.

Vyhláška MZ ČSR č. 11/1988 Sb., o povinném hlášení ukončení těhotenství, úmrtí dítěte a úmrtí matky.

Zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů, ve znění pozdějších předpisů.

Registr lékařů, zubních lékařů a farmaceutů (RLZF):

Účelem zjišťování požadovaných údajů je zajištění údajů o věkové struktuře, pohlaví, oboru činnosti, kvalifikační skladbě lékařů, zubních lékařů a farmaceutů, druhu vlastnictví a druhu zařízení, a to bez ohledu na skutečnost, zda lékař, zubní lékař nebo farmaceut vykonává své povolání v resortu zdravotnictví, obrany, vnitra nebo spravedlnosti.

Aktualizace RLZF se provádí jedenkrát ročně, vždy k 31.12. daného roku. Předmětem aktualizace jsou změny, které v průběhu roku u lékaře, zubního lékaře či farmaceuta nastanou, např. vydání průkazu odbornosti, složení specializované způsobilosti, ukončení pracovního poměru, nástup do pracovního poměru, změna evidenčního stavu nebo změna úvazku.

Údaje z RLZF jsou předávány Českému statistickému úřadu pro Statistickou ročenku České republiky, Ministerstvu zdravotnictví, Institutu postgraduálního vzdělávání ve zdravotnictví, jednotlivým odborným společnostem České lékařské společnosti Jana Evangelisty Purkyně.

Agregované údaje jsou dále zasílány mezinárodními organizacím (počty lékařů podle věku, pohlaví, specializací či NUTS II) jako např. Světové zdravotnické organizaci (WHO), Organizaci pro ekonomickou spolupráci a rozvoj (do databáze OECD Health Data), EU (do databáze Eurostatu).

Legislativa platná pro RLZF:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Zákon č. 95/2004 Sb., o podmínkách získávání odborné a specializované způsobilosti k výkonu zdravotnického povolání lékaře, zubního lékaře a farmaceuta a další prováděcí předpisy.

Národní registr uživatelů lékařsky indikovaných substitučních látek (NRULISL):

Prioritním účelem sběru údajů v Národním registru uživatelů lékařsky indikovaných substitučních látek je možnost zdravotnických zařízení poskytujících substituční léčbu ověřit si před zahájením léčby, zda pacientovi není poskytována substituční terapie v jiném zdravotnickém zařízení a zabránění vícečetné preskripci a úniku substituční látky na nelegální trh. Dále shromažďování dat o pacientech při vstupu a výstupu ze substitučního programu, jejich kontrola, ukládání a zpracování, možnost zpracovávat statistické údaje za pacienty a zdravotnická zařízení poskytující substituční péči a vyhodnocení substituční léčby.

Změnu v substituční léčbě přinesla novela zákona č. 379/2005 Sb., o opatřeních k ochraně před škodami způsobenými tabákovými výrobky, alkoholem a jinými návykovými látkami a o změně souvisejících zákonů, která nabyla účinnost dnem 1. ledna 2006. Podle § 20 odst. 2 písm. j) tohoto zákona jsou zdravotnická zařízení ambulantní péče a zdravotnická zařízení podávající nebo předepisující látky nahrazující původní návykovou látku, tedy zařízení a lékaři poskytující substituční léčbu, povinna hlásit pacienty podstupující tuto léčbu do Národního registru uživatelů lékařsky indikovaných substitučních látek. Tím by se mělo zabránit vícenásobné preskripci, což je zároveň primárním účelem tohoto registru. Tato povinnost je stanovena bez ohledu na odbornou specializaci lékaře.

Národní registr uživatelů lékařsky indikovaných substitučních látek tedy poskytuje souhrnné údaje pro statistické přehledy, epidemiologické studie a zdravotnický výzkum. Souhrnná data jsou podkladem pro tvorbu, realizaci a vyhodnocování preventivních zdravotnických programů a pro odhady potřebných finančních nákladů na zajištění substituční léčby.

Údaje jsou poskytovány Národnímu monitorovacímu středisku pro drogy a drogové závislosti, Sekretariátu rady vlády pro koordinaci protidrogové politiky, Ministerstvu zdravotnictví České republiky a dalším státním orgánům.

Ústav zdravotnických informací a statistiky je správcem a zpracovatelem Národního registru uživatelů lékařsky indikovaných substitučních látek ve smyslu zákona č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů, ve znění pozdějších předpisů.

Legislativa platná pro NRULISL:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění pozdějších předpisů.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Usnesení vlády č. 111/1998 ke Koncepci a programu protidrogové politiky.

Zákon 379/2005 Sb., o opatřeních k ochraně před škodami způsobenými tabákovými výrobky, alkoholem a jinými návykovými látkami, ve znění pozdějších předpisů.

Standard substituční léčby (částka 3/2008 Věstník MZ).

Národní onkologický registr (NOR):

Účelem NOR je registrace onkologických onemocnění a periodické sledování jejich dalšího vývoje, tj. shromažďování dat, jejich verifikace, ukládání, ochrana a zpracování. NOR poskytuje souhrnné údaje pro statistické přehledy jak na národní, tak i mezinárodní úrovni, dále pro epidemiologické studie a zdravotnický výzkum.

Údaje NOR slouží také k podpoře včasné diagnostiky a léčby novotvarů a přednádorových stavů, ke sledování trendů jejich výskytu, příčinných faktorů a společenských důsledků.

Souhrnná data jsou podkladem pro tvorbu, realizaci a vyhodnocování preventivních zdravotnických programů a pro odhady potřebných finančních nákladů na zabezpečení komplexní onkologické péče.

Legislativa platná pro NOR:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Výnos MZSV ČSR č. 3/1989 Věstníku MZSV ČSR (reg. ve Sbírce zákonů částka 19/1988) - Dispenzární péče o nemocné s přednádorovými stavy a novotvary a povinné hlášení novotvarů.

Národní registr cévní chirurgie (NRCCH):

Účelem zjišťování požadovaných údajů (sledování typů operací tj. tepenných rekonstrukcí a specializovaných výkonů na žilách, včetně rizikových faktorů, četnosti komplikací) je možnost hodnocení kvality léčby, sledování dostupnosti odborné péče o pacienty s cévními chorobami v jednotlivých regionech.

Získané informace jsou podkladem pro zlepšení péče o nemocné s chorobami cév v ČR s přímým ekonomickým dopadem. Informace dále slouží Ministerstvu zdravotnictví a odborné společnosti pro potřebná organizační opatření.

Legislativa platná pro NRCCH:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Národní kardiologický registr (NKCHR):

Účelem zjišťování požadovaných údajů je možnost sledování vývoje, příčin a důsledku závažných kardiovaskulárních onemocnění a stavů; stanovení, sledování a vyhodnocení národních ukazatelů kvality, výsledků lékařské a sesterské kardiologické péče; hodnocení potřeb a stavu kardiologických intervencí z hlediska kvality, efektivity, výsledků a výdajů.

Legislativa platná pro NKCHR:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Národní registr kloubních náhrad (NRKN):

Účelem zjišťování požadovaných údajů je registrace údajů o nemocných léčených operativně s užitím endoprotézy a specifických informací blíže upřesňujících tuto léčbu.

Souhrnná data jsou podkladem pro tvorbu, realizaci a vyhodnocování preventivních zdravotnických programů a pro odhady potřebných finančních nákladů na zabezpečení komplexní ortopedické péče.

Legislativa platná pro NRKN:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Národní registr kardiovaskulárních intervencí (NRKI):

Účelem zjišťování požadovaných údajů je vytvořit národní centrálně vedenou zdravotnickou dokumentaci osob se společensky závažným onemocněním ischemickou chorobou srdeční, u kterých byla provedena kardiovaskulární intervence (katetrizace, angioplastika). Registr umožní hodnocení výsledků provedených výkonů v celostátním i mezinárodním srovnání i hodnocení vývoje kvality zdravotnické péče o pacienty, u kterých byla provedena kardiovaskulární intervence. Rychlá dostupnost údajů lékařům s oprávněním vstupu do Národního kardiochirurgického registru a do Národního registru kardiovaskulárních intervencí zásadně ovlivňuje možnost záchrany života při poskytování léčebné péče v této oblasti. Souhrnná data jsou podkladem pro tvorbu zdravotnických programů a odhady finančních nákladů na zdravotnickou péči v daném oboru.

Legislativa platná pro NRKI:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Národní registr nemocí z povolání (NRNP):

Účelem zjišťování požadovaných údajů je získání informací pro analýzu problémů v oblasti ochrany zdraví při práci, kdy nemoci z povolání a ohrožení nemocemi z povolání (dále jen „nemoci z povolání“) jsou jedním ze základních ukazatelů účinnosti prevence, pro rozhodování kompetentních orgánů o přijetí potřebných organizačních a dalších opatření, pro vědecký výzkum, pro vzdělávání v oboru a pro mezinárodní srovnávání. Výskyt nemocí z povolání je též jedním z důležitých ukazatelů zdravotního stavu obyvatelstva, zejména populace v produktivním věku. Zdravotní závažnost nemocí z povolání je umocňována jejich ekonomickými a sociálními důsledky. Z hlediska zdravotního i společenského představují nemoci z povolání vysoce nežádoucí jev, kterému je třeba předcházet vhodně volenými preventivními kroky. Předpokladem účinných opatření, usilujících o minimalizaci výskytu nemocí z povolání, jsou validní informace z této oblasti.

Legislativa platná pro NRNP:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění zákona č. 156/2004 Sb.

Vyhláška č. 552/2004 Sb., o předávání osobních a dalších údajů do NZIS pro potřeby vedení národních zdravotních registrů.

Zákon č. 258/2000 Sb., o ochraně veřejného zdraví, ve znění pozdějších předpisů.

Nařízení vlády č. 290/1995 Sb., kterým se stanoví seznam nemocí z povolání.

Vyhláška MZ č. 342/1997 Sb., kterou se stanoví postup při uznávání nemocí z povolání a vydává seznam zdravotnických zařízení, která tyto nemoci uznávají, ve znění vyhlášky č. 38/2005 Sb.

Nařízení vlády č. 178/2001 Sb., kterým se stanoví podmínky ochrany zdraví zaměstnanců při práci.

Vyhláška č. 432/2003 Sb., kterou se stanoví podmínky pro zařazování prací do kategorií.

Metodické opatření č. 11/2004 Věstníku MZ

Zajištění jednotného postupu při ověřování podmínek vzniku onemocnění pro účely posuzování nemocí z povolání a ohrožení nemocí z povolání

Národní registr osob nesouhlasících s posmrtným odběrem tkání a orgánů:

Účelem registrace je evidence osob, které nesouhlasí s darováním tkání a orgánů tak, aby nemohlo dojít k neodpovídající manipulaci s jejich tělem po smrti.

Dárcovství orgánů a tkání je vysoce humánním rozhodnutím. Darované orgány a tkáně zachraňují v akutních případech životy mnoha lidem a dalším umožňují prodloužení spokojeného života o mnoho let. To se týká zejména pacientů s onemocněním ledvin, srdce, plic a jater.

Každý člověk má rovněž možnost za svého života vyjádřit nesouhlas s tím, aby po jeho smrti byly ze zemřelého těla odebrány tkáně a orgány vhodné k transplantaci.

Legislativa:

Zákon č. 285/2002 Sb., transplantační zákon.

Vyhláška č. 434/2004 Sb., o podrobnostech rozsahu a obsahu povinně uváděných dat do Národního registru osob nesouhlasících s posmrtným odběrem tkání a orgánů
Věstník MZ č. 10/2004.

Národní registr asistované reprodukce (NRAR):

Národní registr asistované reprodukce (NRAR) je celoplošným populačním registrem. V rámci NRAR jsou evidovány všechny ženy, u kterých byla zahájena ovariální stimulace nebo bylo zahájeno monitorování za účelem léčby sterility (sterility vlastní nebo sterility jiné ženy v případě darování oocytů) metodou mimotělního oplodnění (IVF) nebo příbuznými technikami. Sledování IVF cyklů zajišťuje nezbytné informace o způsobu, průběhu, výsledcích a případných komplikacích pro potřeby odborných zdravotnických pracovníků, Ministerstva zdravotnictví ČR, zdravotních pojišťoven i pro mezinárodní vykazování údajů. Získané informace umožňují hodnocení léčebných postupů a jsou využívány pro řízení a zkvalitňování péče o neplodné páry a pro realizaci státní politiky v oblasti asistované reprodukce a léčby sterility.

Legislativa platná pro NRAR:

Zákon č. 20/1966 Sb., o péči o zdraví lidu, ve znění pozdějších předpisů.

Zákon č. 227/2006 Sb., o výzkumu na lidských embryonálních kmenových buňkách a souvisejících činnostech a o změně některých souvisejících zákonů.

Zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů, ve znění pozdějších předpisů.

5. Zabezpečení serveru NIS a datové komunikace

V následujících kapitolách se budeme zabývat zabezpečením serveru s nemocničním informačním systémem (NIS) CLINICOM a datové komunikaci mezi zdravotnickými zařízeními. Nutno podotknout, že NIS je aplikace, jenž je na serveru provozována obvykle spolu s dalšími aplikacemi. Pokud tedy chceme zabezpečit server s NIS, musíme brát ohled i na ostatní aplikace, které ve výsledku snižují úroveň zabezpečení. Tato skutečnost se týká především povolených portů a vytvořených uživatelských oprávnění. Postup pro zabezpečení serveru tedy nelze nikterak standardizovat, lze pouze dodržovat jistá základní pravidla.

Při datové komunikaci mezi jednotlivými zdravotnickými zařízeními se snažíme dosáhnout alespoň takové úrovně zabezpečení, aby byla zaručena stejná důvěryhodnost a integrita dat jako při listinné komunikaci.

5.1 Zabezpečení serveru NIS

Pro správné zabezpečení přístupu k serveru nemocničního informačního systému (NIS), tedy k datům, jež jsou na serveru uložena, je zapotřebí splnit jisté - základní kritéria.

V prvé řadě je zapotřebí zabezpečit server a data, jenž obsahuje, před neoprávněným přístupem pomocí externích periferií. Je tedy nezbytně důležité využívat silných hesel pro přihlášení do samotného operačního systému, na němž je aplikace NIS instalována. Zabezpečení a oprávnění pro přihlášení se liší dle použitého operačního systému.

Dále musíme server s NIS zabezpečit proti neoprávněnému vniknutí ze sítě LAN (Local Area Network) případně WAN (Wide Area Network), k níž je server s NIS připojen. Pro ochranu serveru před „útoky/zahlcením“ ze síťového připojení lze využít například firewall. Ten by měl, při správném nastavení, samotný server dostatečně ochránit. Primární funkcí firewallu je ochránit lokální-vnitřní síť (LAN) před nebezpečím ze sítě internet. Pokud však nejsme schopni technicky zamezit přístup neoprávněné osoby-zařízení do vnitřní sítě, tak je vhodné aplikovat firewall i v rámci lokální sítě, nejlépe i na jednotlivé pracovní stanice.

Jestliže je samotný server s NIS zabezpečen proti neoprávněnému přístupu-vniknutí, tak je zapotřebí zajistit bezpečnost datové komunikace mezi serverem a klienty (klienty - rozumíme pracovní stanice, které jsou připojeny k serveru). Zde je opět důležité rozlišit, v jaké síti se daný server nachází. Jedna-li se o přenos dat v bezpečné lokální síti nebo intranetu, kde je přístup k datům omezen pouze na uživatele, kteří ho mají mít, tak není zapotřebí přenášená data-soubory, například ve tvaru datového standardu, dále zabezpečovat. Pokud se ovšem server nachází v síti, do které mají přístup i neoprávnění uživatelé, například síť internet, tak je zapotřebí předávaná data zabezpečit. Zde je možné pro přenos dat využít zabezpečení SSL (Secure Socket Layer), S/MIME (Secure/Multipurpose Internet Mail Extensions), PGP (Pretty Good Privacy). Jednotlivým způsobům zabezpečení je věnována kapitola „Zabezpečení datové komunikace“.

5.2 Datová komunikace mezi zdravotnickými zařízeními

Jelikož mohou být při komunikaci mezi jednotlivými zdravotnickými zařízeními přenášena osobní nebo citlivá data (například identita osoby, zdravotní stav, atd.), je zapotřebí, aby ten, kdo s těmito daty manipuluje, učinil taková opatření, aby nedošlo k jejich ztrátě, zničení, neoprávněnému zpracování a ke krokům vedoucím k jejich zneužití.

Je tedy zapotřebí zajistit tři základní aspekty bezpečnosti [9]:

- Důvěrnost - vlastnost, která znemožňuje odhalení informace neoprávněné osobě.
- Integritu - vlastnost, která umožňuje provedení změny pouze určeným způsobem a pouze oprávněnou osobou nebo procesem
- Dostupnost - vlastnost, která zajišťuje použitelnost informace (nebo jiného aktiva) v požadovaném místě a čase podle požadavku oprávněné osoby [10].

Nejdůležitější je při samotném přenosu datových souborů zajištění důvěrnosti a integrity. Nedostupnost by, za určitých podmínek, mohla mít fatální následky, ovšem v rámci bezpečnosti přenosu dat ji řadíme až na poslední místo z výše zmíněných.

Způsobů komunikace mezi jednotlivými nemocničními informačními systémy a zdravotnickými zařízeními je relativně mnoho.

První, pravděpodobně nejstarší způsob komunikace je pomocí formulářů, mluvíme tedy o listinné-papírové komunikaci. Jednotlivá nemocniční zařízení vyplní konkrétní formulář, případně jej NIS vytvoří a uživatel ho následně vytiskne, ten se poté odešle poštou na příslušné místo určení, případně jej tam doručí pověřený pracovník daného zařízení.

Další možností, je sestavení datového souboru nemocničním informačním systémem, nebo jiným zdravotnickým softwarem, který je poté uložen na paměťovém mediu. Data uložená na mediu jsou následně do cílového místa doručena stejným způsobem jako při listinné komunikaci. Při tomto způsobu komunikace mezi zdravotnickými zařízeními, nebo organizacemi, se nevyužívá žádného zvláštního zabezpečení datového souboru, neboť je zde přímá zodpovědnost konkrétního pracovníka. Je zde tedy zaručena stejná úroveň bezpečnosti jako při listinné komunikaci. Pokud by se ovšem jednalo o náhodné odcizení dat, které svojí povahou nevede k jejich zneužití, tak lze samotnou strukturu datového souboru považovat za částečně zabezpečenou.

Třetím variantou komunikace mezi jednotlivými zdravotnickými zařízeními je za pomoci emailové schránky, nebo internetových stránek s možností vložení datového souboru. Obdobně jako v předchozím případě, NIS vytvoří datový soubor, který je poté odeslán prostřednictvím internetové či jiné sítě cílenému zdravotnickému zařízení. V současné době je pro komunikaci mezi zdravotnickými zařízeními upřednostňována právě tato varianta, jelikož vede k výraznému zefektivnění práce.

5.3 Zabezpečení datové komunikace

Jak již bylo zmíněno, při datové komunikaci mezi jednotlivými zdravotnickými zařízeními se snažíme dosáhnout alespoň takové úrovně zabezpečení, aby byla zaručena stejná důvěryhodnost a integrita dat jako při listinné komunikaci. Použitá technika pro zabezpečení datové komunikace se liší dle konkrétní metody přenosu dat.

Při přenosu souborů (např. elektronickou poštou jako příloha, protokolem FTP, atd.) je možné zabezpečit přenosový kanál nebo samotný soubor. Při interaktivní práci, kde se data neukládají a nepřenášají v podobě souborů, je třeba zabezpečit přenosový kanál [9][11].

Na obrázku 5.1 a 5.2 nalezneme souhrn základních metod zabezpečení, jež jsou rozděleny dle způsobu přenosu a vlastností zabezpečení.

Způsob přenosu:	Příklady použitelných metod:
Elektronická pošta	S/MIME, PGP
Přenos souborů (ftp, http)	VPN, SSL, S/MIME, PGP
Interaktivní zpracování	VPN, SSL

Obr. 5.1: Metody zabezpečení dle způsobu přenosu [9]

	Důvěrnost	Integrita	Neodmítnutí (prokazatelnost) původu
VPN	Ano	Ano, částečně	Ne
SSL (https)	Ano	Ano	Ne
S/MIME	Ano (šifrování)	Ano (podpis)	Ano (podpis)
PGP	Ano (šifrování)	Ano (podpis)	Ano (podpis)

Obr. 5.2: Způsoby zabezpečení dle vlastností [9]

5.3.1 VPN (Virtuální Privátní Síť)

Virtuální privátní síť myslíme takovou sítí, která se jeví jako soukromá, tedy oddělená od všech ostatních sítí, avšak po technické stránce využívá síť veřejné. Přenos dat v rámci virtuální privátní sítě je oddělen od sítě veřejné za použití šifrování a dalších bezpečnostních mechanismů na úrovni jednotlivých paketů. VPN lze vytvořit mezi jednotlivými zařízeními, ale i mezi jednotlivými osobními počítači s příslušným softwarem.

Dnes již téměř výhradně používaná technologie IPsec je sadou protokolů, sloužících pro zajištění autenticity a integrity přenášených dat (AH), zajištění důvěrnosti přenášených dat (ESP) a pro ustanovení spojení a dohodnutí klíčů používaných protokoly AH a ESP (IKE). Vzájemná autentizace při navazování spojení a vyjednávání tzv. bezpečnostních asociací protokolu IKE se může provádět pomocí statických sdílených tajných klíčů, které musí být předem uloženy do konfigurace každého zařízení, nebo na základě asymetrických kryptografických algoritmů s využitím certifikátů veřejných klíčů [9][11].

5.3.2 SSL (Secure Socket Layer Protocol)

Protokol SSL byl vytvořen firmou Netscape a slouží k zajištění bezpečné internetové komunikace. V současné době je standardem, jenž je podporovaný celou řadou produktů. SSL využívá jako transportní mechanismus model protokolu TCP/IP. Vrstva SSL protokolu se nachází ihned za aplikační vrstvou, konkrétně tedy mezi aplikační vrstvou a TCP protokolem. SSL tedy nikterak nezkoumá data, jež mu předá aplikační vrstva, ale pouze je zabezpečí a předá dále, ke zpracování, protokolu TCP. Lze tedy implementovat například protokoly HTTP, FTP, Telnet a jiné [12].

5.3.3 S/MIME (Secure/Multipurpose Internet Mail Extension Protocol)

S/MIME je protokol vyvíjený od roku 1995 s cílem zajistit bezpečnost při e-mailové komunikaci. Od roku 1999 je protokol přijat jako standard. Zajišťuje operace elektronického podpisu, šifrování zprávy a management šifrovacích klíčů pro formát MIME. E-mailová zpráva přenášená po Internetu se skládá ze dvou částí, těla a hlavičky. Hlavička obsahuje informace nezbytné k transportu e-mailové zprávy.

Formát MIME, definuje strukturu těla e-mailové zprávy, zahrnuje například rozšířený text, grafiku a audio-soubory, nenabízí ovšem žádné bezpečnostní služby (zajištění důvěrnosti zprávy, ověření původu a nepopiratelnosti).

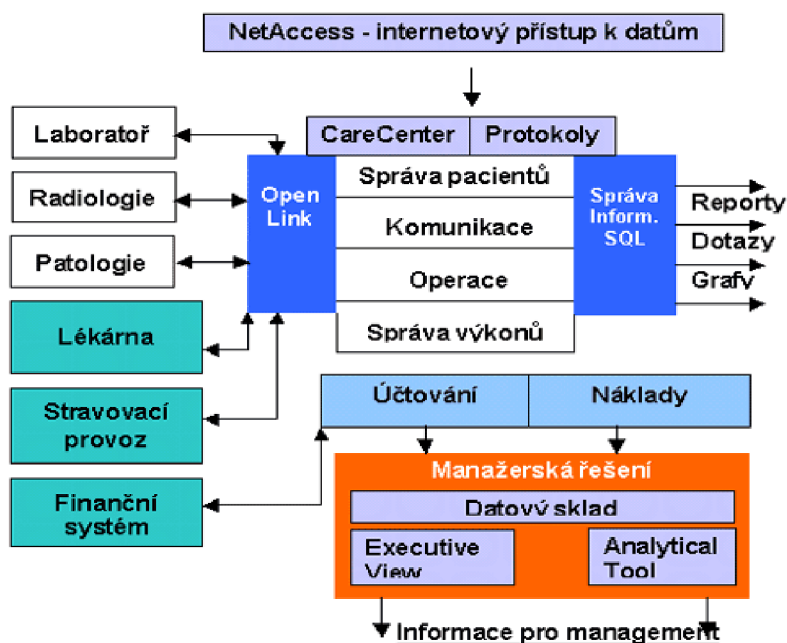
Úkolem protokolu S/MIME je poskytnout tyto bezpečnostní služby podle všeobecně uznávaného standardu. Tělo zprávy ve formátu MIME, je tedy rozšířeno o část se syntaxí, která je výsledkem kryptografických procesů aplikovaných na některou část MIME zprávy. K zajištění důvěrnosti zprávy používá S/MIME, stejně jako protokol SSL, takzvanou digitální obálku, která kombinuje symetrické a asymetrické šifrovací techniky. Vlastní šifrování zprávy je prováděno některou ze symetrických šifer s tajným klíčem, tento tajný klíč je při transportu chráněn pomocí algoritmů s veřejným šifrovacím klíčem. Pro zajištění autenticity a nepopiratelnosti zprávy umožňuje S/MIME připojení digitálního podpisu. Pro výměnu klíčů používá S/MIME certifikáty veřejného klíče. S/MIME podporuje hašovací (matematické) funkce MD5 (Message-Digest Algorithm) a SHA-1 (Secure Hash Algorithm) [9].

5.3.4 PGP (Pretty Good Privacy)

PGP je metoda a současně program pro zabezpečení zpráv elektronické pošty a souborů, který používá vlastní formát dat. Využívá kombinaci symetrických a asymetrických šifrovacích technik podobně jako S/MIME, nerozšiřuje však formát pro e-mailové zprávy, ale pracuje s tělem zprávy nebo samostatným souborem. Obdobně jako S/MIME umožňuje zprávu zašifrovat i připojit digitální podpis. Pro výměnu klíčů jsou používány certifikáty veřejného klíče, nebo původní mechanismus založený na podepisování klíčů ostatními účastníky komunikace a přiřazování důvěry (web of trust). Podporovány jsou symetrické šifrovací algoritmy, asymetrické šifrovací algoritmy a hašovací funkce [9].

6. Nemocniční informační systém CLINICOM

NIS CLINICOM je softwarové řešení firmy SMS, které představuje plně integrovaný komplexní nemocniční informační systém, úspěšně aplikovatelný nezávisle na konkrétní organizaci zdravotní péče a způsobu její úhrady. Je určen pro širokou škálu nemocnic, zaměřených především na poskytování akutní nemocniční péče. Jeho uživatelé jsou nemocnice o velikosti od 100 lůžek až po velké fakultní nemocnice s několika tisíci lůžky. CLINICOM je robustní nemocniční informační systém s modulárním nadčasovým řešením a moderní postrelační databází Caché firmy InterSystems [13].



Obr. 6.1: Schéma Nemocničního Informačního Systému [13]

Financování nemocniční péče prochází v České republice řadou změn. Důležitou vlastností NIS je proto maximální nezávislost na budoucím způsobu účtování péče a plná adaptibilita NIS na platný způsob financování v daném období (výkonový systém, DRG, paušální platby atd.).

NIS CLINICOM vychází ze zkušeností v USA a zemích EU a je proto snadno přizpůsobitelný všem typům účtování péče. NIS CLINICOM je koncipován tak, aby byl v maximální míře postaven na proměnných položkách a číselnících. Pouhým přenastavením hodnot lze potom dosáhnout odlišného chování systému. To oceňují uživatelé především v případě změn účtování lékařské péče. Modulární systém s velkým stupněm možností uživatelského přizpůsobení umožňuje snadné propojení s jinými IS zdravotnického zařízení. Předdefinované nastavení standardních komunikačních protokolů používaných ve zdravotnictví (HL7, EDIFACTS a dalších) umožňuje snadnou komunikaci s dalšími subjekty a nabízí možnost přímého propojení do národních i nadnárodních zdravotnických sítí a on-line výměnu informací [13].

CLINICOM stojí na čtyřech základních pilířích [13]:

Správa pacientů - moderní pojetí centrální správy pacientů, což přináší zásadní výhodu v tom, že všechna data jsou zadávána pouze jedinkrát (snížení chybovosti způsobené vícenásobným vložením).

Správa výkonů - zajišťuje rutinní chod nezávisle na personálu, uživatelé pouze vkládají uskutečněné výkony (neutrální výkony), zatímco systém se stará o správný chod v souladu s legislativou a metodikou (změny legislativy sleduje pouze několik lidí odpovědných za nastavení kmenových souborů).

Komunikace - vystavování žádank, jejich odeslání, příjem výsledků s nastavitelnou úrovní rozlišení a vyúčtování.

Správa operací - účinný nástroj pro sledování zákroků a operací za účelem získání přesných údajů o nákladech a výnosech.

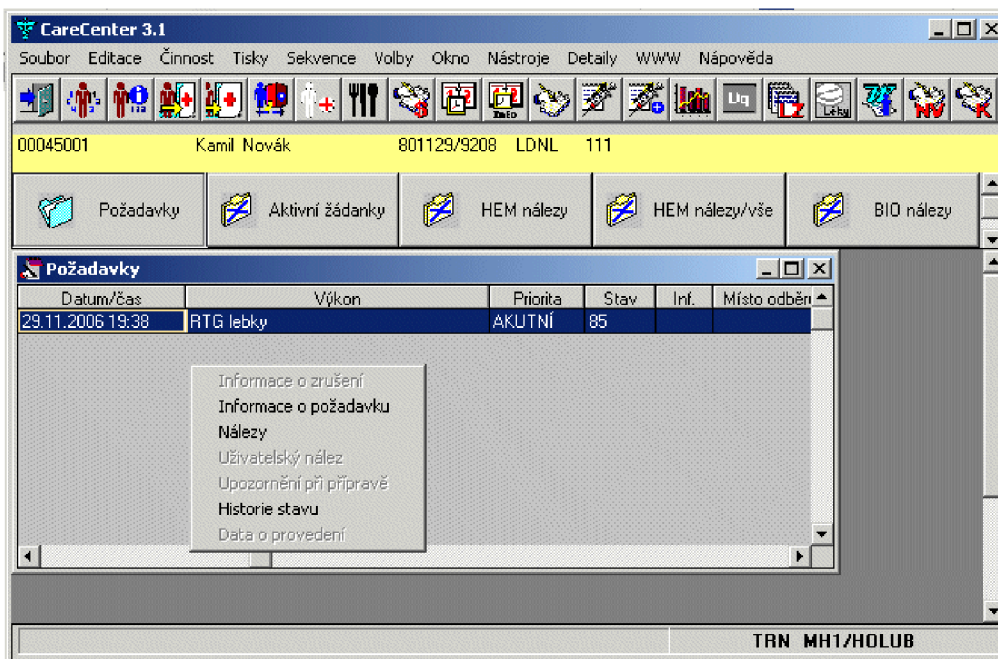
CLINICOM nepředstavuje jen administrativní nástroj lékaře pro usnadnění pořízení a archivace povinné dokumentace, ale je především navržen ke sledování ekonomiky nemocnice [20]. Data z NIS lze dále zpracovat a vyhodnotit prostřednictvím manažerského informačního systému DSS (Decision Support System).

6.1 Uživatelské rozhraní CareCenter systému CLINICOM

CareCenter je grafické uživatelské rozhraní systému CLINICOM, realizované v prostředí Windows. CareCenter zprostředkovává okamžitý přístup k žádankám, nálezům, lékařské dokumentaci a k informacím o protokolech, a to s uspořádáním podle požadavků uživatele. CareCenter poskytuje uživateli jednoduchý pohled na informace o zvoleném pacientovi, ať už je třeba zobrazit nebo vytisknout plán péče, žádanky, nálezy, informace o vyšetření či textovou medicínskou dokumentaci [13].

Jediným přihlášením do systému má uživatel okamžitý přehled o pohybu a stavu pacienta od jeho přijetí, o ordinovaných a provedených vyšetřeních, o výsledcích vyšetření až po ukončení léčby.

Dostupnost urgentních výsledků je automaticky indikována, číselné výsledky vyšetření je možno zobrazit například ve formě grafů a trendů. CareCenter také umožňuje jednoduchý přístup k PC aplikacím třetích stran, např. textovým editorům, apod. Některé aplikace (např. MS Word) jsou plně integrovány, jsou do nich na požádání přenášena patientská data pro další zpracování, případně je jejich práce řízena přímo modulem CareCenter [13].



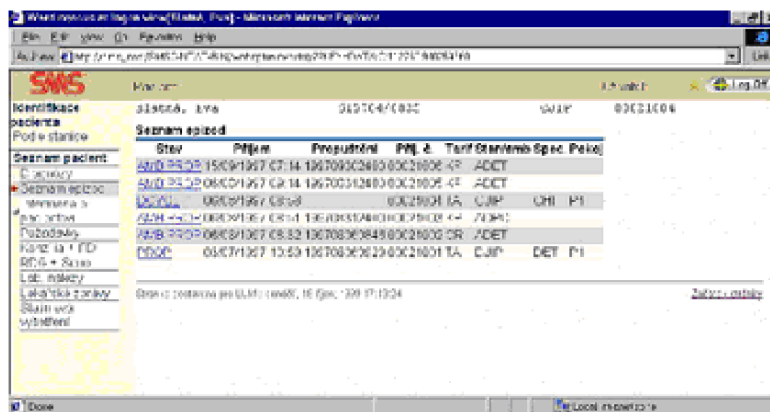
Obr. 6.2: Ukázka grafického rozhraní CareCenter

6.2 Uživatelské rozhraní NetAccess systému CLINICOM

NetAccess představuje jednoduchý zabezpečený externí přístup k lékařským informacím prostřednictvím Internetu/intranetu a umožňuje tak uživatelům přístup k datům NIS nejen z lokální sítě nemocnice, ale také z každého místa, které disponuje připojením na Internet (např. z domova, z konzultačních míst, z ambulance mimo nemocnici, z ordinací praktických lékařů apod.) [13].

NetAccess slouží k zadávání požadavků a zobrazení vybraných údajů. Pracuje se stejnými daty jako CareCenter a uživatelé nabízejí také velmi podobné grafické prostředí pod Windows. Je podporován nejrozšířenějšími internetovými prohlížeči. Rozhraní patří mezi aplikace klient – server, funguje tedy stejně tak jako přístup na email přes webové rozhraní. Výhody tohoto přístupu jsou nesporné, minimalizují se náklady na údržbu koncového zařízení a nevznikají problémy s nekompatibilitou. Samozřejmě musí být vysoký stupeň zabezpečení.

NetAccess využívá standard SSL pro zabezpečené spojení. Veškeré informace a parametry jsou přenášeny v kódované podobě, navíc je zajištěno, že data nejsou ukládána do paměti počítače uživatele, což vede k tomu, že data nemůže nikdo později zobrazit. Další formy zabezpečení lokálních sítí nemocnic, v souvislosti s připojením k Internetu, nabízí SMS v rámci svých služeb [13].



Obr. 6.3: Ukázka rozhraní NetAccess [13]

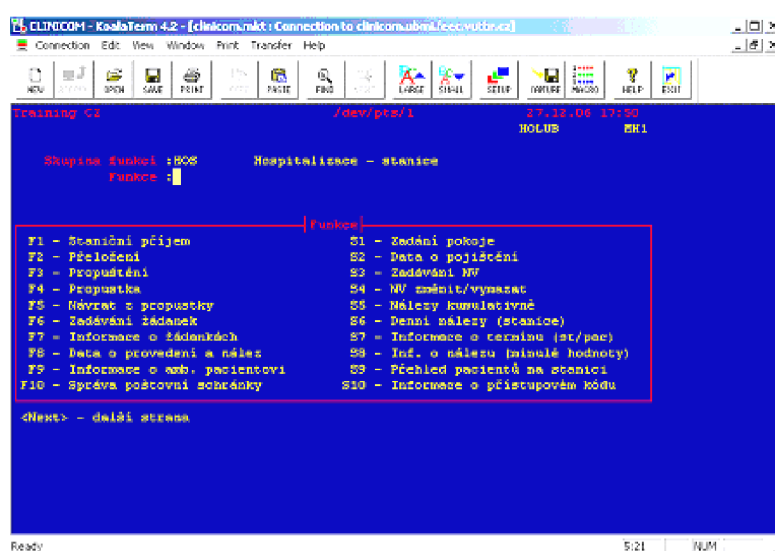
6.3 Uživatelské rozhraní Telnet/SSH systému CLINICOM

Do NIS CLINICOM se dá také plně přistupovat přes zabezpečený protokol SSH (Secure Shell). Postačí nám k tomu terminálové okno v textovém režimu.

Přístup tedy není omezen operačním systémem. V operačním systému Windows můžeme pro přístup na CLINICOM pomocí SSH použít například program KoalaTerm, který lze stáhnout na stránkách jeho výrobce (<http://www.foxitsoftware.com>).

Snad jedinou nevýhodou přístupu přes terminál je menší pohodlnost a přehlednost, kterou nám plně nabízí grafické rozhraní CareCenter.

Výhodou SSH přístupu je jeho minimální náročnost na hardwarové vybavení.



Obr. 6.4: Ukázka přístupu přes SSH pomocí programu KoalaTerm

7. Databázová platforma Caché

Databázové prostředí nemocničních informačních systémů SMS je založeno na postrelační objektové databázi Caché americké firmy InterSystems sídlící v Cambridge v americkém státu Massachusetts. Ta je více než 25 let významným světovým dodavatelem databázových technologií v oblasti zdravotnictví [13].

Společnost InterSystems označuje Caché jako první postrelační databázi. Rozdíl oproti relačním databázovým platformám, které jsou založeny na prostých tabulkách, je v možnosti vkládání vícerozměrných a objektově orientovaných dat.

Postrelační databáze Caché, tedy ukládá data do vícerozměrného datového modelu, což umožňuje efektivnější popis dat. Caché nabízí vysoký výkon při zpracování komplexních transakcí (je až 20x výkonnější než srovnatelné standardní relační databáze). Díky své architektuře vyniká i dalšími vlastnostmi, např. snadnou dostupností dat přes Internet a jednoduchou správou. Databáze Caché byla představena v roce 1997. Je nejen vysoce výkonným databázovým prostředím, ale také prostředím pro rychlý vývoj aplikací. Byla optimalizována pro efektivní tvorbu rychlých transakčních aplikací s možností snadného růstu. Databázi Caché používají nezávislí dodavatelé softwaru, stejně jako IT organizace ve zdravotnictví, finančnictví, veřejné správě a v mnoha dalších oblastech.

V současnosti pracuje na světě více než 100.000 systémů založených na databázi Caché a používá je více než 4 miliony koncových uživatelů v 88 zemích světa [13].

7.1 Vzdálený přístup k uloženým datům v databázi Caché

Vzdálený přístup k databázi téměř vždy využívá různé typy komunikačních rozhraní. U databáze Caché je pro vzdálený přístup k aktuálním datům použito populární rozhraní ODBC (Open DataBase Connectivity), případně JDBC (Java DataBase Connectivity), spolu s dotazovacím jazykem SQL (Structured Query Language) včetně jazyka DDL (Data Definition Language).

Prostřednictvím tohoto rozhraní můžeme z databáze získat data pro další zpracování. Například v aplikaci Microsoft Excel nebo při tvorbě našich vlastních aplikací.

7.1.1 Ovladač ODBC

Jak již bylo zmíněno, jednou z možností, jak získat aktuální data z databáze Caché, je za použití populárního rozhraní otevřené databázové konektivity (ODBC). Toto rozhraní má své výhody i nevýhody. Za největší výhodu můžeme považovat přímé propojení s databázovým serverem, které nám zajišťuje absolutní aktuálnost dat. Ovšem přímé požadavky server zatěžují, což vede k celkovému zpomalení činnosti serveru.

Přestože za tvorbou ODBC stojí společnost Microsoft, není použití ODBC nijak vázáno pouze na platformu Windows. ODBC lze používat i v operačních systémech Unix, OS/2, MacOS a dalších [14].

7.1.2 Architektura ODBC

Model struktury ODBC se dá znázornit pomocí čtyř vrstev [14].

V první nejvrchnější vrstvě se nachází samotná aplikace. Ta v případě, že potřebuje data, provede volání ODBC funkcí (ve formě SQL dotazu).

Druhou vrstvou je tzv. "Správce ODBC ovladačů" (ODBC Driver Manager). Úkolem správce ovladačů je zajistit propojení mezi aplikací a příslušným ODBC ovladačem (ODBC ovladače tvoří třetí vrstvu modelu, podrobněji viz dále).

Jakmile aplikace potřebuje data, správce ovladačů vyhledá a nahraje příslušný ovladač ve formě DLL knihovny. Správce ovladačů také zjistí, jaké konkrétní funkce jsou podporovány jednotlivými ovladači a uschová si jejich adresy v paměti do tabulky. V případě, že aplikace volá konkrétní funkci, správce souborů zjistí, ke kterému ovladači funkce patří a zavolá ji.

Tímto způsobem může být prováděn souběžný přístup k více ovladačům, což se hodí v případě programování aplikací, přistupujících souběžně k několika zdrojům dat.

Třetí již zmíněnou vrstvou jsou ODBC ovladače. Ty provedou zpracování volané ODBC funkce, přeložení požadavku do SQL pro příslušný SRBD (Systém Řízení Báze Dat) a jeho následné poslání.

Poslední vrstvou je SRBD (Systém Řízení Báze Dat), který provede zpracování operace požadované ODBC ovladačem a výsledky této operace mu vrátí.

7.1.3 Instalace ovladače ODBC

Instalaci ovladače ODBC do operačního systému Windows lze provést dvěma způsoby. První způsob instalace je pomocí spuštění instalačního programu, který rozpakuje a posléze nainstaluje ovladač ODBC. Pokud však nemáme ovladač ODBC v podobě instalačního souboru, ale v podobě samotného driveru, musíme označit soubor s koncovkou „.inf“ a vybrat volbu „instalovat“. Tuto volbu nalezneme v menu, které se zobrazí po kliknutí pravého tlačítka myši, na již zmiňovaném souboru. Instalace je v obou případech zcela automatická a není tedy nikterak složitá.

7.1.4 Nastavení ovladače ODBC

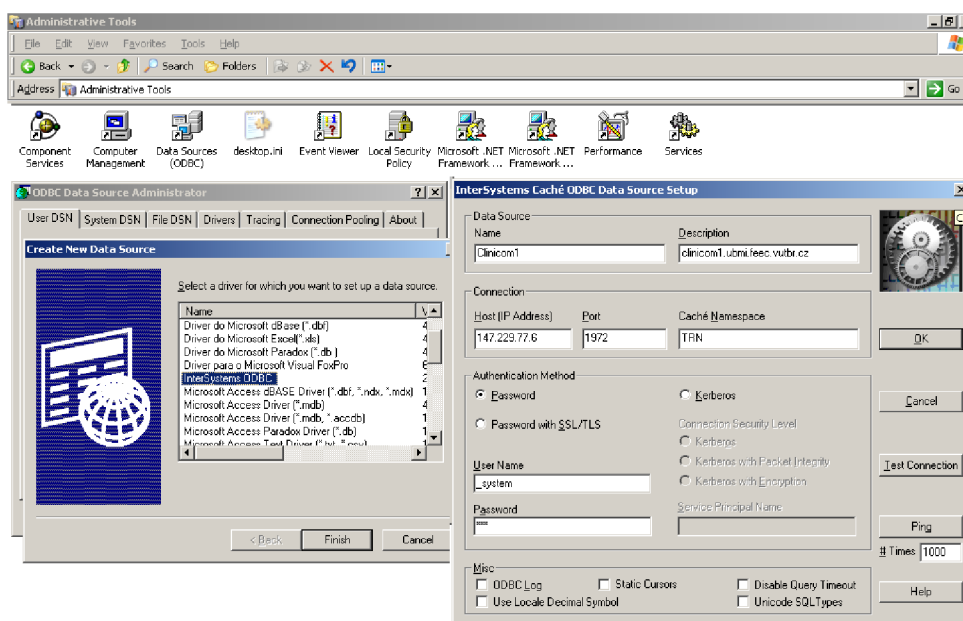
Nastavení ovladače pod systémem Windows provádíme v „Data Sources (ODBC)“, které nalezneme v „Administrative Tools“.

Nejprve musíme patřičný ODBC ovladač přidat. V našem případě vybereme ovladač „InterSystems Caché SQL ODBC“ a potvrdíme výběr. V nastavení ovladače zvolíme pojmenování právě přidávaného ovladače, například „Caché ODBC“, typ připojení vybereme „TCP“.

Dále zadáme server, na němž se nachází databáze, ke které se prostřednictvím ODBC ovladače chceme připojit.

V našem případě je to pokusný server „clinicom1.ubmi.feec.vutbr.cz“, nebo můžeme zadat přímo IP adresu serveru „147.229.77.6“.

Za adresou serveru následuje port, na kterém je daná služba na serveru spuštěná. Na serveru „clinicom1.ubmi.feec.vutbr.cz“ je tento přístup dostupný na standardním portu „1972“, jenž je zvolen při instalaci Caché.
Do položky databáze vyplníme „TRN“. Dále pak uživatelské jméno a heslo přístupu.



Obr. 7.1: Ukázka nastavení ovladače ODBC

7.2 Realizace přístupu do databáze Caché

Při samotné realizaci přístupu do databáze Caché stojíme před zásadní otázkou: Zda pro realizaci přístupu využít již zmíněného populárního ODBC driveru spolu s webovým serverem, pomocí něhož bychom získaná data upravili do požadované podoby a vytvořili tak webovou aplikaci na vzdáleném serveru, nebo využít nástrojů, které nabízí software Caché, pro vytvoření webové aplikace přímo na daném serveru, kde je databáze Caché provozována.

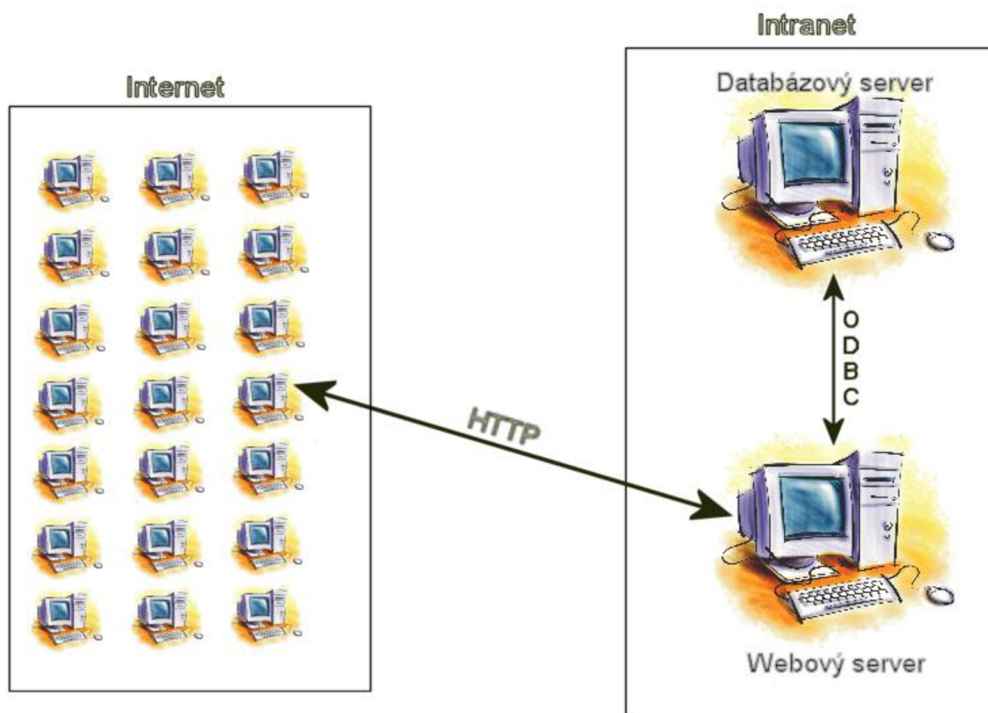
Nutno podotknout, že výše zmíněné možnosti realizace nejsou zdaleka jediné možné. Jelikož chceme, aby bylo možné data z databáze zobrazit kdekoliv, bez složitosti instalování dané aplikace včetně případného doladování a nastavování pro daný operační systém, nabízí se jako ideální právě tvorba webové aplikace, jejímž jediným požadavkem je připojení počítače (případně PDA, SmartPhone, atd.) do společné sítě spolu s databází Caché. Ať již mluvíme o síti veřejné (Internet), privátní (intranet) anebo případně o jejich kombinaci.

Obě výše zmíněné varianty realizace mají své klady a zápory. Proto bychom měli naznačit realizaci obou variant a vybrat tu, která bude lépe vyhovovat našim účelům - naší webové aplikaci.

První varianta využívá již zmiňované rozhraní ODBC a webový server třetí strany (Apache, MS IIS, atd.). Výhodou je možnost oddělit fyzicky databázi od webového serveru. Jinak řečeno, lze provozovat databázový server v intranetu, ale data z něj zobrazovat pomocí webového serveru do Internetu viz obrázek č. 7.2. Ovšem v našem případě, kdy je

databázový server připojen do sítě Internet, je tato výhoda bezvýznamná. Další výhodou, pro nás již podstatnější, je možnost výběru příslušného programovacího jazyka, ve kterém budeme aplikaci vytvářet. Právě díky univerzálnosti a podpoře ODBC driveru je možné si zvolit téměř jakýkoliv běžně využívaný programovací jazyk určený-optimizovaný pro tvorbu webových aplikací (PHP, ASP, atd.).

Nevýhody plynoucí z využití ODBC driveru (zatížení databázového serveru při velké četnosti dotazů, zabezpečení přenosu mezi servery atd.) a provozování dalšího serveru jsou zřejmé.



Obr. 7.2: Ukázka jednoho z možných způsobů zapojení první zmiňované varianty

Druhá varianta spočívá ve využití nástrojů, které nabízí software Caché. Konkrétně budeme mluvit o Caché Server Pages (CSP), jenž je obsažen v instalaci spolu s databází Caché. Výhodou tohoto nástroje je jednoduchost přístupu do databáze. Dále nám pro méně složité webové aplikace odpadne přímá nutnost instalace plnohodnotného webového serveru, jelikož Caché již jednoduchý webový server obsahuje. Bohužel neumožňuje překlad oblíbených programovacích jazyků (PHP, ASP, atd.). Důležité je ovšem to, že Caché umožňuje doinstalování jiných webových serverů (např.: Apache, MS IIS, atd.), čímž získáme modifikovanou první a druhou variantu, kde by plnohodnotný webový server spolu s databází fungoval fyzicky na jediném serveru.

Pro realizaci naší webové aplikace jsme zvolili „čistou“ druhou variantu. Tedy variantu bez narušení chodu a konfigurace databázového serveru a instalace plnohodnotného webového serveru. Tato varianta v sobě sice skrývá úskalí, plynoucí z malého rozšíření formy jazyka, který Caché CSP využívá, ovšem pro naši aplikaci je tato varianta nejzajímavější, jelikož si při ní ukážeme využití softwaru dodávaného s databází Caché.

7.3 Utility Caché (tzv. kostka)

Caché své administrativní nástroje seskupilo do takzvané „kostky“ (Caché cube) viz obr. 7.3. Ta obsahuje všechny potřebné nástroje pro správu celé databáze. Díky těmto utilitám jsme schopni zcela nastavit nejen databázi, která je provozována na daném serveru, ale i vzdálené databáze Caché, ke kterým máme přístup.

Kostka seskupuje řadu aplikací, které si zde nyní uvedeme, avšak podrobněji si popíšeme pouze ty, jež budeme využívat při realizaci naší webové aplikace.



Obr. 7.3: Položky v menu Caché kostky

První položka, kterou můžeme v Caché kostce zvolit, slouží k získání základních informací pro nové uživatele.

Další dvě položky umožňují spustit, případně zastavit, činnost databáze Caché.

Následují položky:

Terminal – slouží k emulaci terminálového prostředí příkazového řádku na lokálním nebo na vzdáleném systému Caché.

Explorer – slouží pro správu databáze, jmenných prostorů spolu s rutinami a globály.

System Management Portal – umožňuje kompletní administraci Caché.

Documentation – obsahuje rozsáhlou dokumentaci pro Caché.

Remote System Access – umožňuje přistupovat na přednastavený vzdálený server se systémem Caché.

Preferred Server – vybírá preferovaný server. Případně umožňuje přidání dalších vzdálených serverů s instalovaným systémem Caché.

About – zobrazí informaci o aktuálně nainstalované verzi Caché.

Exit – zavře kostku Caché.

Položky: Studio a SQL Manager jsme vynechali záměrně, jelikož si je v následujících kapitolách rozvedeme podrobněji.

7.3.1 Caché studio

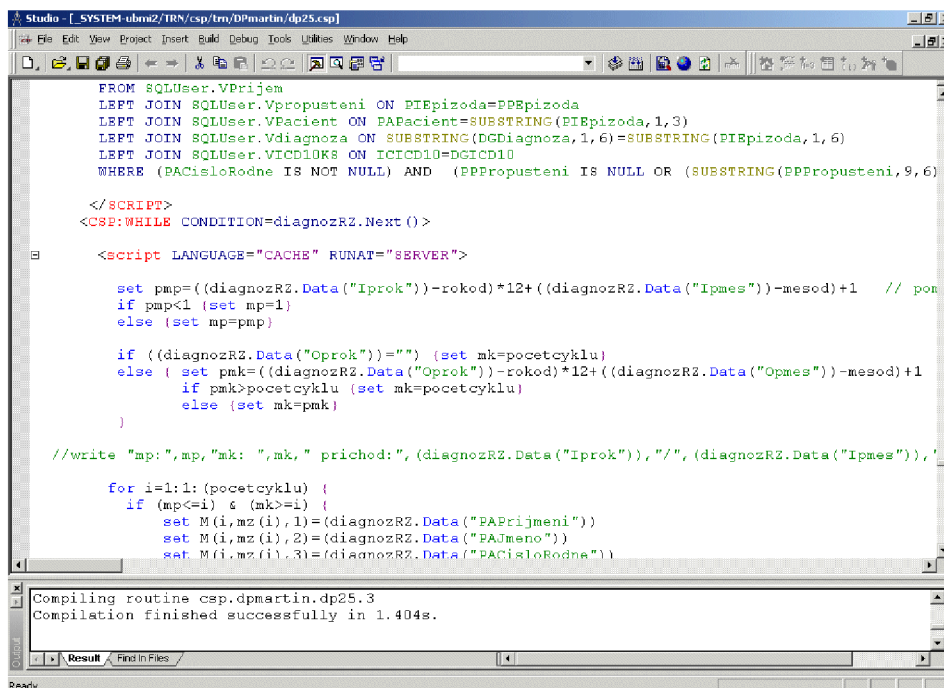
Patří mezi nejdůležitější vývojářský nástroj, který obsahuje kostka Caché viz obr. 7.4. Jedná se o editor s grafickým rozhraním, který umožňuje vytváření tříd, stránek CSP, rutin pomocí jazyků Caché Basic a Caché ObjectScript.

Při tvorbě naší webové aplikace využijeme Caché studio jako editor CSP stránek a poslouží nám také k získání orientace v jednotlivých třídách databáze NIS CLINICOM.

Na samotném počátku tvorby webové aplikace bude zapotřebí vytvořit v Caché Studiu prázdný soubor s koncovkou csp.

Ten můžeme vytvořit v nabídce File -> New -> CSP File -> Caché Server Pages.

Nyní máme vytvořený prázdný csp soubor. Do tohoto souboru budeme zapisovat veškerý potřebný zdrojový kód k naší webové aplikaci. Konkrétní obsah již vytvořené webové aplikace lze nalézt v příloze č. 1.



```
FROM SQLUser.VPrijem
LEFT JOIN SQLUser.Vpropusteni ON PPEpizoda=PEPepizoda
LEFT JOIN SQLUser.VPacient ON PAPacient=SUBSTRING(PIEpizoda,1,3)
LEFT JOIN SQLUser.Vdiagnoza ON SUBSTRING(DGDiagnoza,1,6)=SUBSTRING(PIEpizoda,1,6)
LEFT JOIN SQLUser.VICD10K8 ON ICTCD10=DGICD10
WHERE (PACisloRodne IS NOT NULL) AND (PPPpropusteni IS NULL OR (SUBSTRING(PPPpropusteni,9,6)

</SCRIPT>
<CSP:WHILE CONDITION=diagnozRZ.Next()>

<script LANGUAGE="CACHE" RUNAT="SERVER">

set pmp=((diagnozRZ.Data("Iprok"))-rokod)*12+((diagnozRZ.Data("Ipmez"))-mesod)+1 // pon
if pmp<1 (set mp=1)
else (set mp=pmp)

if ((diagnozRZ.Data("Oprok"))="") (set mk=pocetcyklu)
else ( set pmk=((diagnozRZ.Data("Oprok"))-rokod)*12+((diagnozRZ.Data("Opmez"))-mesod)+1
if pmk>pocetcyklu (set mk=pocetcyklu)
else (set mk=pmk)
)

//write "mp: ",mp,"mk: ",mk," prichod: ",(diagnozRZ.Data("Iprok")),"/", (diagnozRZ.Data("Ipmez")),

for i=1:1:(pocetcyklu) {
if (mp<=i) & (mk>=i) {
set M(i,mz(i),1)=(diagnozRZ.Data("PAPrijmeni"))
set M(i,mz(i),2)=(diagnozRZ.Data("PAJmeno"))
set M(i,mz(i),3)=(diagnozRZ.Data("PACisloRodne"))
}
```

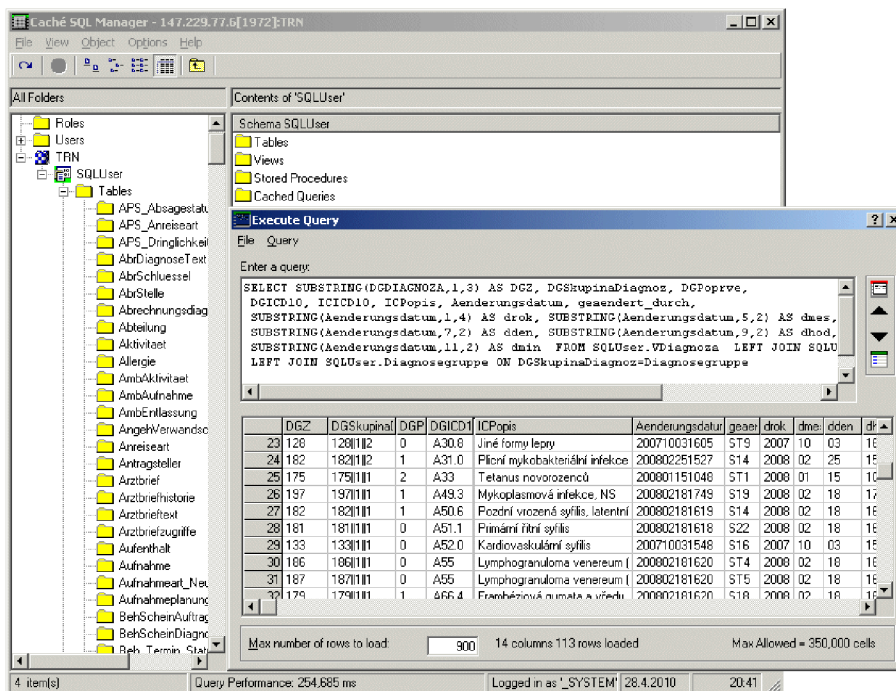
Compiling routine csp.dpmartin.dp25.3
Compilation finished successfully in 1.404s.

Obr. 7.4: Caché studio

7.3.2 Caché SQL Manager

SQL (Structured Query Language) manager (viz obr. 7.5) je dostupný pouze v určitých verzích Caché balíčku v závislosti na parametrech instalace. SQL manager slouží ke kompletní správě relačního přístupu k databázi. Umožňuje prohlížení a editaci jednotlivých tabulek v dané databázi.

To nám umožní získat přehled o rozložení jednotlivých položek mezi tabulkami a jejich vzájemnými vazbami. Jednou z nedocenitelných funkcí Caché SQL Manageru je Execute Query, díky níž můžeme testovat naše SQL dotazy. Výsledek je okamžitě zobrazen a my jej můžeme snadno a rychle zkontrolovat. Tuto funkci využijeme při návrhu a testování SQL dotazů směřovaných na databázi NIS CLINICOM.



Obr. 7.5: Ukázka aplikace Caché SQL Manager spolu se zpracovaným SQL dotazem

7.4 Caché UDA (Unified Data Architecture)

Díky unifikované datové architektuře UDA (Unified Data Architecture) poskytuje Caché jednoduché a přesvědčivé řešení problematických oblastí. Reprezentací dat jako standardních definic tříd propojuje unifikovaná datová architektura starý svět s novým [15].

Jinak řečeno nám UDA umožňuje uložená objektová data lehce převést do relačních tabulek, aniž by byla data v databázi uložena více než jednou.

Relační teorie používá k reprezentaci modelované reality jediný konstrukční prvek: relaci [15]. Pod relací si můžeme představit jednotlivé tabulky, které se skládají z řádků reprezentující jednotlivé záznamy a z jednotlivých atributů, tedy sloupců. Hlavička tabulky zajišťuje přiřazení vlastností sloupcům a každý řádek pak odpovídá instanci nebo n-tici. Řádky tabulky nejsou samy o sobě nikterak seřazeny. To znamená, že datové záznamy nemají konkrétní pořadí. Určitý sloupec nebo kombinace sloupců tvoří tzv. primární klíč (či identifikátor řádku). Ten se používá kdykoli je to možné k nalezení konkrétního řádku. Jedna tabulka se může odkazovat na další za předpokladu, že ve sloupci první tabulky je použit jako hodnota primární klíč druhé (tedy cizí) tabulky. Tento sloupec se pak označuje jako cizí klíč [15].

7.5 Normální formy

Pro relační model jsou stanovena určitá normalizační pravidla, která přesně definují způsob použití tabulek. Normalizace by měla vést ke vzniku tabulek, které lze snadno a efektivně udržovat. Obecně platí, že čím je tabulka ve vyšší normální formě, tím kvalitněji je navržena. Pro tabulky jsou definovány následující stupně - normální formy [16]:

- Nultá normální forma (0NF) - Tabulka spadá do nulté normální formy právě tehdy, pokud existuje alespoň jedna položka, která obsahuje více než jednu hodnotu.
- První normální forma (1NF) - Tabulka spadá do první normální formy tehdy, pokud každý její atribut - sloupec obsahuje jen hodnoty, které jsou z pohledu databáze již nedělitelné - atomické. Jinak řečeno, hodnoty v rámci daného sloupce jsou neredukovatelné, nelze je tedy rozložit na systém jednodušších množin beze ztráty informace.
- Druhá normální forma (2NF) - Tabulka spadá do druhé normální formy, pokud splňuje podmínky 1NF a každá její položka závisí na primárním klíči a nejen na jeho podmnožině.
- Třetí normální forma (3NF) - Tabulka spadá do třetí normální formy, pokud splňuje 2NF a všechny její neklíčové atributy jsou vzájemně nezávislé.
- Čtvrtá normální forma (4NF) - Tabulka spadá do čtvrté normální formy, je-li ve 3NF a popisuje pouze příčinnou souvislost (jeden fakt).
- Pátá normální forma (5NF) - Tabulka spadá do páté normální formy, pokud je ve 4NF a není možné do ní vložit nový sloupec (případně skupinu sloupců) tak, aby se vlivem skrytých závislostí rozpadla na více dílčích tabulek.

Teorie říká, že tabulky by měly být alespoň ve třetí normální formě. V praxi se ukazuje jako nejlepší rozložit vše do co nejvyšší normy a pak dávat zpět na takovou úroveň, kterou je schopen unést databázový stroj [17].

Tabulka, která vyhovuje třetí normální formě, se považuje za normalizovanou.

7.5.1 Vazby mezi tabulkami

Databáze je určitá, uspořádaná množina informací, jež jsou uspořádány do jednotlivých tabulek. Mezi tabulkami jsou vytvořeny vztahy, které dělíme dle vazeb do následujících skupin [16]:

Vazba one-to-one (1:1)

Vyjadřuje vztah, kdy právě jeden záznam má vztah k právě jednomu jinému unikátnímu záznamu. Například každý jeden unikátní občan ČR má jedno unikátní rodné číslo. Tato vazba není často využívána, protože se většinou daří data s touto vazbou ukládat v rámci jedné tabulky.

Vazba one-to-many (1:N)

Jedná se o nejčastěji používanou vazbu. Atribut v tabulce může v tomto případě nabývat právě jedné hodnoty z množiny hodnot definovaných v tabulce druhé. Příkladem je třeba vztah člověka a jeho rodného města. Měst je mnoho, ale každý z nás se narodil pouze v jednom z nich. Stejnou vazbou, ale opačně, je vazba many-to-one (N:1).

Vazba many-to-many (M:N)

Vhodným příkladem této vazby je vztah pacienta a lékaře. Každý lékař má mnoho pacientů a každý z těchto pacientů může mít zároveň několik různých lékařů. Pomocí SQL dotazu se ovšem nedá tento vztah přímo vyjádřit. Využívá se zde takzvané „mezi-tabulky“ s vazbami 1:N na požadované tabulky.

7.6 SQL (Structured Query Language)

Jazyk SQL byl vytvořen jako standardní dotazovací jazyk pro přístup k relačním databázím již v roce 1974 pod názvem Sequel. Patří mezi populární jazyk, sloužící nejen pro dotazování, ale i pro správu celé databáze. V naší webové aplikaci budeme využívat jeho část DQL (Data Query Language) právě pro dotazování (získání) jednotlivých dat z databáze NIS CLINICOM.

Pro vkládání, mazání a upravování jednotlivých dat a struktur v tabulkách, které jsou obsaženy v databázi, využíváme jazyk DML (Data Manipulation Language).

Vytváření, mazání a změny jednotlivých tabulek (datových struktur) provádíme pomocí jazyka DDL (Data Definition Language).

Samotné dotazování na jednotlivé záznamy provádíme pomocí již zmíněného jazyka DQL. Bezpochyby nejznámějším příkazem jazyka DQL je „SELECT“. Tento příkaz (dotaz) vytváří výstupní tabulku, obsahující data z jedné, případně z více tabulek dle zadaných kritérií.

Vybrané parametry pro příkaz SELECT, které budeme využívat, jsou následující:

Nejprve je zadán seznam výrazů, který většinou obsahuje odkaz na sloupce příslušných tabulek. Můžeme použít kombinaci sloupců nebo vybrat všechny sloupce z příslušných tabulek pomocí zástupného znaku „*“. Pokud však chceme vybrat pouze část obsahu z daného sloupce, například počátek rodného čísla, lze využít funkce „SUBSTRING“.

Za výběrem sloupců následuje klauzule „FROM“, za kterou je uveden zdroj dat, kterým je nejčastěji tabulka, případně seznam tabulek. Určuje tedy, z kterých zdrojů se mají zadané sloupce vybírat. Pokud zvolíme více jak jeden zdroj, dojde k provedení kartézského součinu, který sloučí zadané zdroje dohromady.

Pro přehlednost výsledného dotazu ovšem využijeme klauzuli „JOIN“. Ta umožňuje spojení zdrojů a aplikaci predikátů, které se povinně vkládají za klíčové slovo „ON“. Samotné spojení dělíme na vnější a vnitřní.

Pro vysvětlení si uvedeme příklad z databáze NIS CLINICOM. Pokud chceme spojit tabulku, obsahující informace o pacientech, s tabulkou, která obsahuje jejich jednotlivé diagnózy, musíme použít vnějšího spojení, jelikož každý pacient nemusí mít zadanou diagnózu. V případě vnitřního spojení by takový záznam nebyl v celkovém výběru zobrazen. Jestliže by měl každý pacient povinně zadanou diagnózu, mohli bychom použít spojení vnitřní. Klíčovým slovem „ON“ následně určíme, podle jaké podmínky se mají nalezená data z tabulky sloučit (například podle id). Dále může následovat klauzule „WHERE“, sloužící k zobrazení řádků, které vyhovují zadané podmínce.

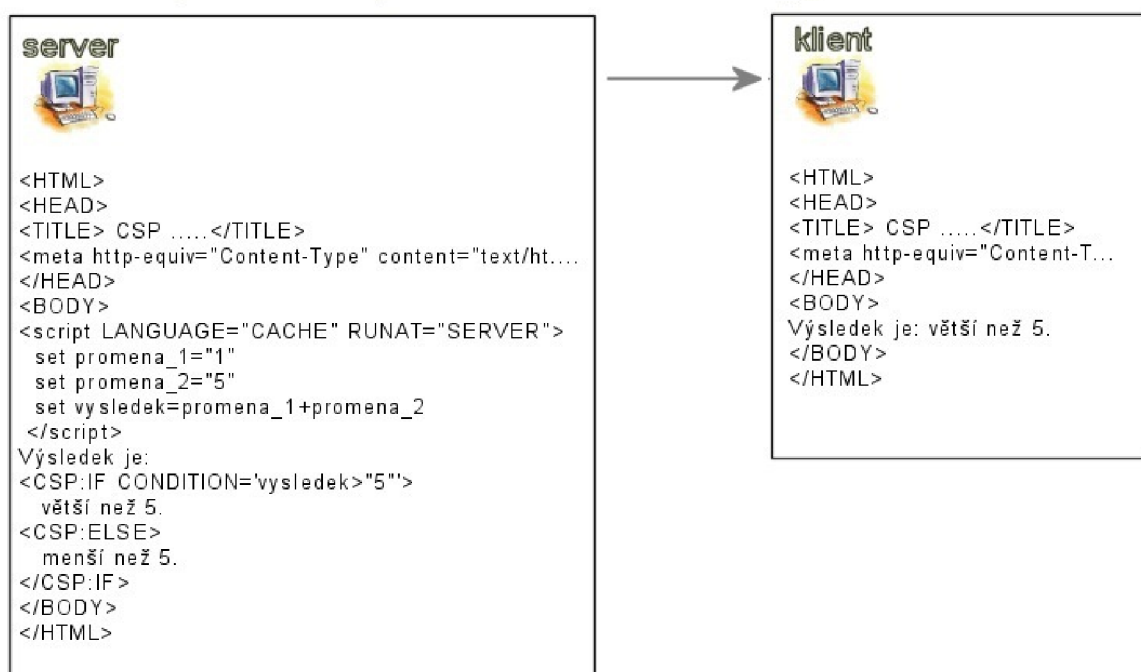
Výsledné nalezené záznamy našeho dotazu můžeme rovněž seskupovat do řádků, které mají stejnou (zadanou) hodnotu pomocí „GROUP BY“. Případně je můžeme řadit pomocí klauzule „ORDER BY“, která definuje, jak budou řádky vráceny na výstup (do výstupní tabulky příkazu SELECT). Například zajistí abecední seřazení záznamu podle jména. Ukázka příkazu SELECT, jazyka Caché SQL:

```
SELECT SUBSTRING(RODNECISLO,1,6) AS NAROZENI, ICICD10, ICPOPIS  
FROM SQLUser.VDiagnoza  
LEFT JOIN SQLUser.VICD10KS ON DGICD10=ICICD10  
WHERE ICICD10 LIKE 'A%' GROUP by ICICD10 ORDER by DGICD10
```

8. Caché Server Pages (CSP)

Caché Server Pages (CSP) je v Caché podporováno od verze 4. Umožňuje vytvářet dynamický webový obsah závislý na čase, vazbách v uložených datech, uživatelských skupin apod. a dokonce i na obsahu, který byl vygenerován náhodně. Umožňuje to princip stránek HTML (HyperText Markup Language) se specifickými značkami (tagy), které jsou prováděny na straně serveru Caché pokaždé, když je stránka vyvolána a vrací individuální obsah stránky Caché Server Pages. V CSP může být integrován nejen jazyk HTML, ale i obrázky a libovolné další binární soubory [15]. Jinak řečeno, umožňuje CSP vytváření dynamických internetových stránek. Jednoduchý příklad zobrazuje následující obrázek č. 8.1.

Příklad zpracování dynamické webové stránky na straně serveru.



Obr. 8.1: Ukázka zpracování jednoduchého skriptu na straně serveru

CSP nejprve zpracuje skripty, určené ke zpracování na straně serveru, a poté je uživateli přenesen výsledek jejich činnosti. Přičemž výsledek je začleněn přímo do struktury jazyka HTML. Při tvorbě dynamického obsahu nám CSP nabízí sadu jazykových prvků, mezi které patří: provádění skriptů, databázové dotazy, cykly, větvení kódu, substituce hodnot atd. Níže si popíšeme ty jazykové prvky, které budeme při tvorbě naší webové aplikace aktivně využívat.

8.1 Substituce hodnot

CSP umožňuje využívání proměnných, které budou při běhu programu nahrazeny aktuální hodnotou dané proměnné. Příkladem jednoduché substituce hodnot je proměnná „vysledek“ v našem vzorovém příkladu na obrázku č. 8.1.

Pokud bychom chtěli zobrazit výsledek proměnné na straně klienta, použili bychom na straně serveru výraz v následujícím tvaru „#(vysledek)#“. Hodnota proměnné bude

při generování nahrazena za její obsah, tedy za číslo „6“. K substituci hodnot ovšem nemusí docházet pouze při generování stránky, ale i při samotné kompilaci. Takovou substituci rozeznáme podle výrazu „##(...)##“. V naší webové aplikaci je takto řešena položka „Poslední úprava provedena:“.

8.2 Provádění skriptů

Provádění skriptů je rovněž ukázáno na vzorovém obrázku č. 8.1. Skript samotný rozeznáme díky značce „script“. Za touto značkou musíme zapsat skriptovací jazyk, kterým bude daný skript napsán. Caché Server Pages podporuje tyto skriptovací jazyky: Caché ObjectScript, Caché SQL, Caché Basic, VBScript a JavaScript [18]. Dále určíme, na jaké straně bude skript zpracován, zda na straně serveru či klienta. Poté následuje samotný kód skriptu, který je ukončen značkou „</script>“.

8.3 Cykly

Cykly všeobecně známe téměř z každého programovacího jazyka. Obsah cyklu je vykonáván do doby, dokud neskončí zadaná podmínka. Stejně tak je tomu u CSP cyklu. Značkovací cykly jazyka CSP jsou například „CSP:LOOP“ a „CSP:WHILE“. Při tvorbě webových aplikací s přístupem do databáze se bez druhého jmenovaného cyklu pouze stěží obejdeme. V naší aplikaci jej budeme využívat pro cyklické opakování výpisu dat z databáze, které nám umožní vypsat jednotlivé řádky. Například výběr diagnózy je rovněž řešen cyklem WHILE. Blok celého cyklu je ukončen značkou „</CSP:WHILE>“, za kterou se již cyklus neprovádí.

8.4 Větvení kódu

Podobně jako cykly podporuje CSP i větvení kódu. Díky větvení kódu jsme schopni zpracovávat chod aplikace dle zadaných podmínek. Mezi základní značky pro větvení kódu patří „CSP:IF“, který je možné doplnit značkou „CSP:ELSE“ případně „CSP:ELSEIF“. Blok obsahu podmínky IF je pak zakončen značkou „</CSP:IF>“. Jednoduchý příklad využití větvení programu je na obrázku 8.1, kde dle vyhodnocení podmínky IF, na serveru, dojde k výpisu na straně klienta. Pokud by byl obsah proměnné „vysledek“ nižší než zadaná podmínka, vyhodnotil by jí server při zpracování jako nepravdivou a zpracoval by obsah, který je uložen v bloku značky CSP:ELSE. Jednotlivé podmínky můžeme vkládat „do sebe“, čímž bude docházet k neustále rozsáhlejšímu větvení programu. Ovšem je velmi důležité, abychom při realizaci větvení programu nezapomněli, každou z podmínek řádně ukončit.

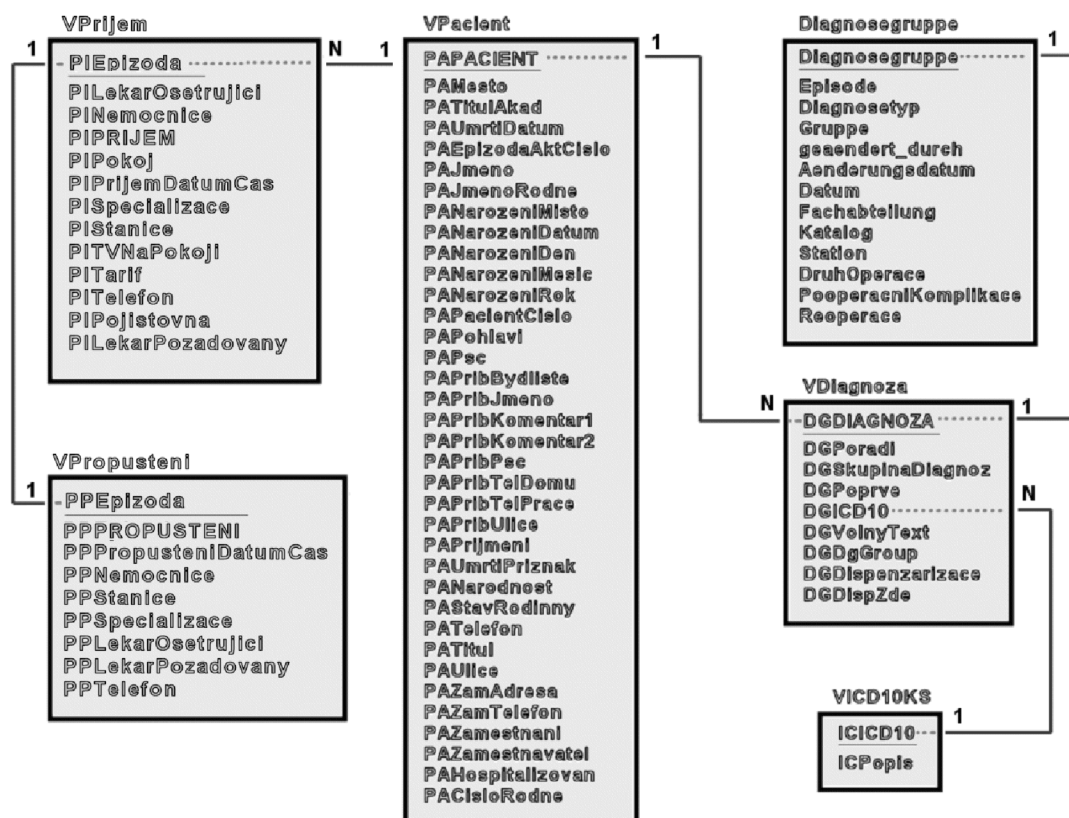
8.5 Databázové dotazy

Databázové dotazy nám umožňují zobrazit konkrétní data z naší databáze. CSP podporuje dvě formy zápisu. Dynamické databázové dotazy, které zapisujeme stejně jako skripty jazyka Caché SQL. Předdefinované databázové dotazy pak vkládáme do značek „SQL:QUERY“. Obě varianty využívají již zmiňovaný jazyk SQL. Konkrétním zápisem SQL dotazů se již zabývat nebudeme.

9. Realizace webové aplikace

Nyní, po objasnění základních struktur pro tvorbu webové aplikace, můžeme přistoupit k samotné realizaci. Jak již bylo zmíněno, aplikace bude realizována pouze pomocí jednoho serveru, na němž je plně funkční InterSystems Caché včetně podpory Caché Server Pages.

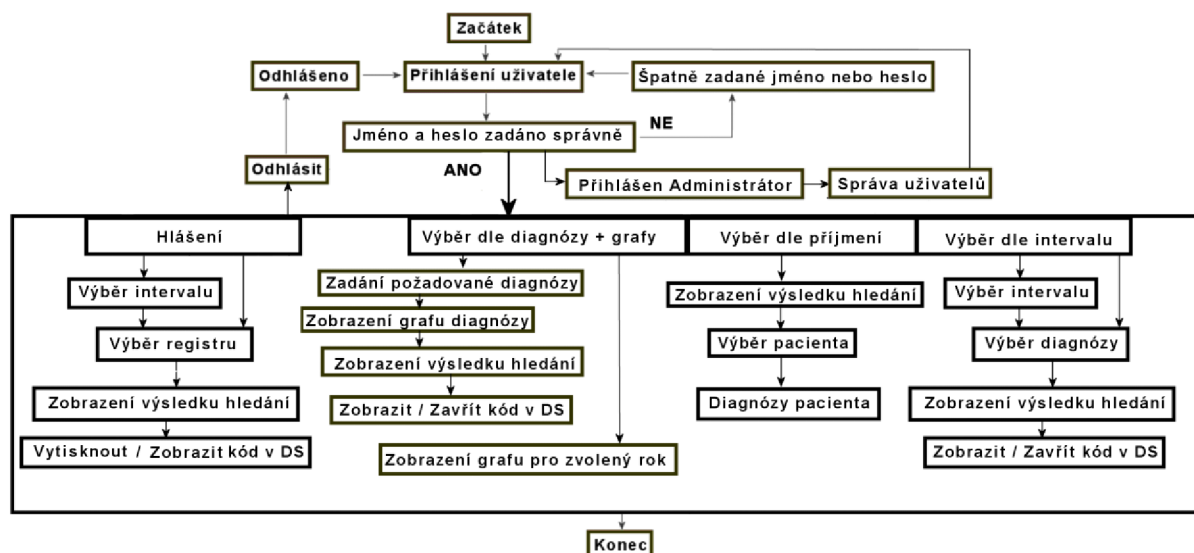
Nejprve jsme „prozkoumali“ strukturu databáze NIS CLINICOM, abychom mohli správně určit jednotlivé vztahy mezi tabulkami (relační přístup). Částečná struktura tabulek, které budeme pro jednotlivé výpisy potřebovat, je včetně vzájemných vztahů vyobrazena na obrázku 9.1. Jedná se o strukturu, zjištěnou pomocí Caché SQL Manageru, který je součástí kostky Caché.



Obr. 9.1: E-R diagram vybraných tabulek NIS CLINICOM

Po dohledání všech potřebných tabulek z rozvětvené struktury databáze NIS CLINICOM, je důležité správně navrhnout programový diagram výsledné webové aplikace.

Diagram na obrázku 9.2 vyobrazuje návrh dosavadní blokové struktury programu. Jedná se o modifikovanou strukturu předchozího projektu, který byl dle zadání diplomové práce inovován a rozšířen o další funkce.



Obr. 9.2: Návrh programového diagramu webové aplikace

Výsledný zdrojový kód webové aplikace je uložen ve dvou souborech, přičemž odkazování probíhá pouze na „hlavní“ soubor, ve kterém je dle přeposlané proměnné aktivována vždy jen určitá část-větev programu. Druhý soubor slouží pouze pro generování kódu ve formátu datového standardu MZČR verze 3.

V praxi, kde by tato webová aplikace byla pouhou součástí rozsáhlého projektu, bychom pravděpodobně použili rozdělení kódu dle jednotlivých částí do více souborů. Například kaskádové styly (CSS), sloužící převážně pro úpravu vzhledu stránky, by byly uloženy v jediném centrálním souboru pro celý projekt. Poté by se stačilo pouze odkazovat na soubor, který by kaskádové styly obsahoval. Toto řešení značně zjednodušuje zápis a případné opravy/úpravy vzhledu převážně v rozsáhlých projektech. Ve vytvořené webové aplikaci by však nenašlo většího uplatnění.

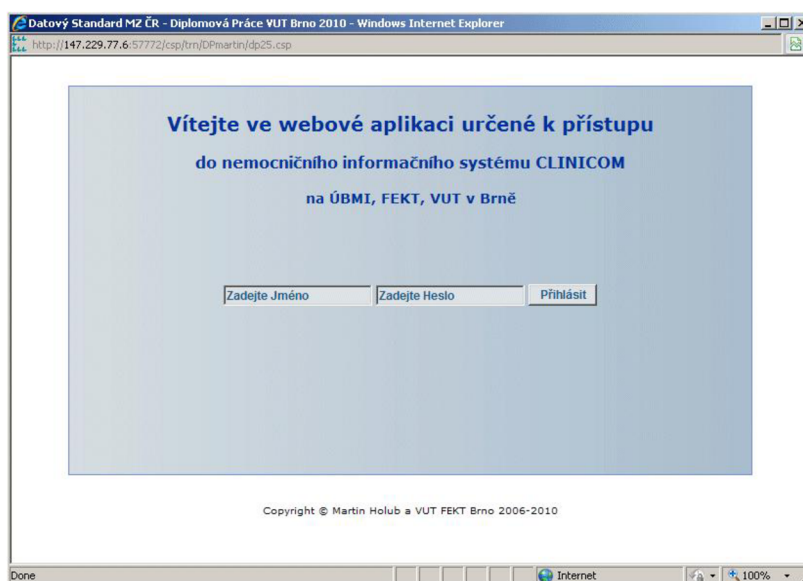
U vytvořené aplikace byl kladen důraz na příjemný, nenáročný a přehledný vzhled, jenž nebude působit rušivým dojmem.

9.1 Ukázka webové aplikace

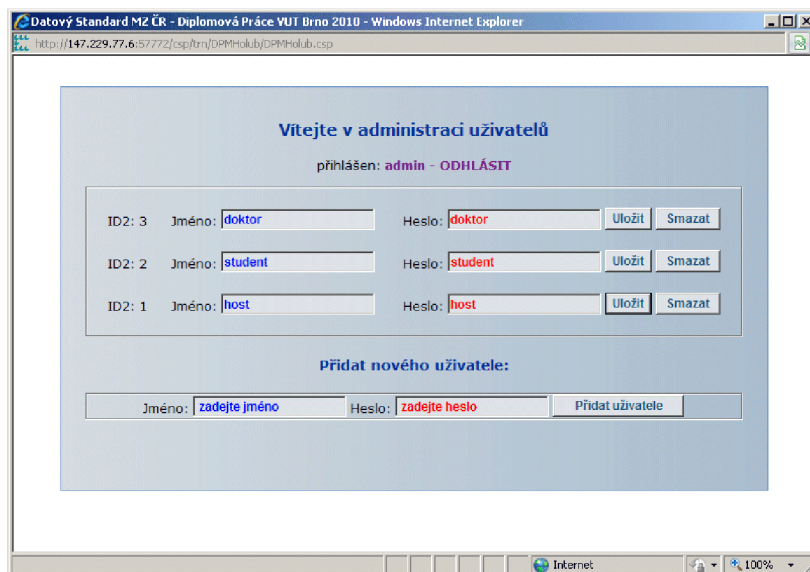
Nyní si ukážeme výběr nejdůležitějších částí vytvořené webové aplikace.

Na úvodní stránce (<http://147.229.77.6:57772/csp/trn/DPMHolub/DPMHolub.csp>) jsme vyzváni k přihlášení (viz obr. 9.3). Přihlašovací údajům je vyhrazena tabulka „uživatelé“ v databázi Caché. Pokud se do webové aplikace přihlásíme pod administrátorským heslem, tak můžeme přidávat, editovat a mazat jednotlivé uživatele (viz obr. 9.4). Jediný uživatel, kterého nelze smazat, je administrátor. Pro volný vstup do aplikace můžeme využít přednastaveného uživatelského jména „doktor“ a hesla rovněž „doktor“.

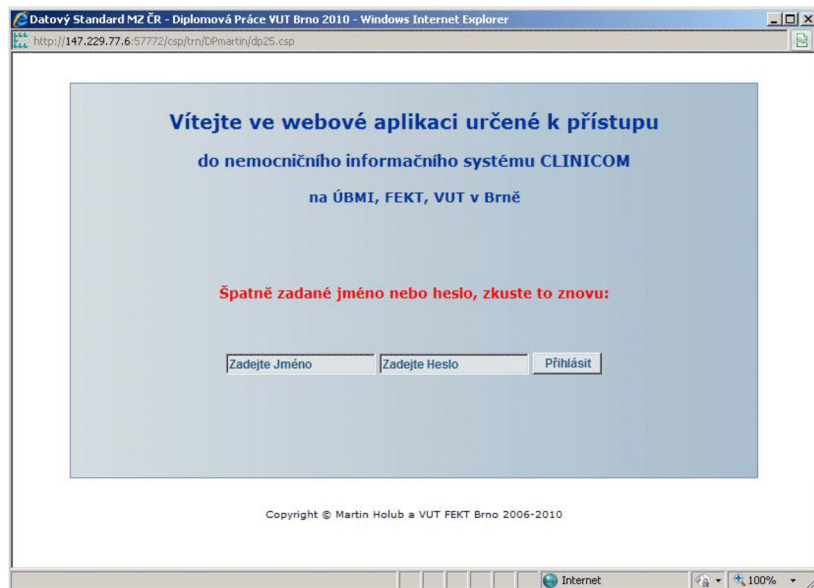
V případě, že zadáme neplatné jméno nebo heslo, budeme vyzváni k opětovnému zadání přihlašovacích údajů (viz obr. 9.5). Po zadání platného jména a hesla jsme přeměrováni na hlavní stránku aplikace (viz obr. 9.6). Odhlásit se, po přihlášení můžeme kdykoliv kliknutím na tlačítko „ODHLÁSIT“ jenž nalezneme v levém horním rohu.



Obr. 9.3: Přihlášení do webové aplikace



Obr. 9.4: Administrace uživatelů



Obr. 9.5: Výzva k zadání správného jména a hesla



Obr. 9.6: Úvodní stránka webové aplikace

Hlavní stránka aplikace obsahuje menu na levé straně, horní, dolní informační panel a obsahovou část. Horní panel nás informuje o datu, přihlášeném uživateli a nabízí možnost odhlášení. Dolní panel má pouze informativní charakter. Menu aplikace, nacházející se na levé straně, je rozděleno na tři části: „Hlášení“, „O aplikaci“ a „Doplňkové funkce“. Položka „O aplikaci“ nás informuje pouze o účelu vzniku této aplikace. Webová aplikace je tedy rozdělena na dvě hlavní části: „Hlášení“ a „Doplňkové funkce“.

9.1.1 Menu „Hlášení“

První část „Hlášení“ je věnována povinným hlášením do registrů Národního zdravotnického informačního systému. Zde jsou prozatím zpracovány dva zvolené registry. Konkrétně se jedná o Národní registr hospitalizovaných (NRHOSP) a Národní onkologický registr (NOR). Přihlášený uživatel si může zvolit požadovaný registr a časový interval, v němž si přeje nalezené výsledky z databáze NIS CLINICOM zobrazit (viz obr. 9.7).

Dnes je: 17/05/2010
Uživatel: Doktor - ODHLÁŠIT

Hlášení
O aplikaci
Doplňkové funkce

Zadejte prosím požadovaný Národní zdravotní registr NRZ a časový interval

Zvolte prosím požadovaný registr: Zvolte prosím konkrétní registr

Zvolte počátek intervalu: rok měsíc

Zvolte konec intervalu: rok měsíc

Vyhledat zvolená data z databáze

Poznámka: Zvolte požadovaný interval od-do u nezvoleného údaje se bere automaticky minimum/maximum.

Copyright © Martin Holub a VUT FEKT Brno 2006-2010
Poslední úprava provedena: 17/05/2010 20:05:51

Obr. 9.7: Položka „Hlášení“ – výběr požadovaného registru a intervalu

Aplikace je ošetřena proti nezadání nebo nekorektnímu zvolení jednotlivých údajů následujícím způsobem. Dojde-li k nezadání konkrétního registru, doplní program automaticky první registr z výběru. Pokud nedojde k zadání jednoho či více časových údajů, program automaticky provede doplnění roku případně měsíce dle nalezených záznamů v databázi. Výběr je generován dynamicky, čímž odpadá nutnost dopisování kódu pro následující roky (2011, 2012 atd.) Pokud je hledaný interval například od února 2008 po současnost, tak není nutné zvolit konec intervalu. Rovněž hledáme-li údaj od začátku záznamů v databázi NIS CLINICOM, po únor 2008, zvolíme pouze konec intervalu tedy únor 2008. O veškeré automatické doplnění se stará vytvořený ověřovací skript na straně serveru. O doplnění údajů jsme informováni po stisku tlačítka „Vyhledat zvolená data z databáze“ (viz obr. 9.8) ihned na začátku vypsání dat. Jestliže zadáme nekorektní interval (například: od roku 2010 do 2007 nebo interval od srpna po leden v tomtéž roce), tak jsme vyzváni k novému zadání korektních údajů (viz obr. 9.9).

Dnes je: 16/05/2010
Uživatel: Doktor - ODHLÁŠIT

Nezadali jste: Rok DO >> vyplňuji nejvyšší rok z DB: 2010
Nezadali jste: Měsíc DO >> vyplňuji 05. měsíc
Nezadali jste požadovaný registr >> vyplňuji NRHOSP

Registr: NRHOSP - 09/2007

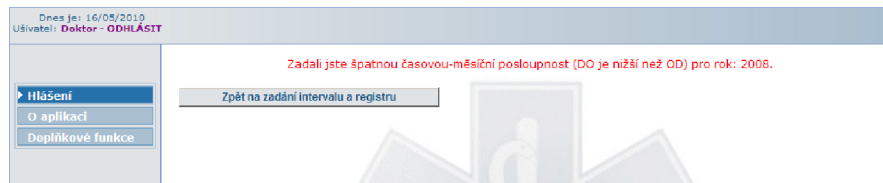
Příjmení	Jméno	Rodné číslo	Přijat	Pokoj	Stanice	Tarif
Novák - detaily	Petr	8509183551	02.10.2005 v 08:53	M1	INTM	T1H1
Nemeth - detaily	Štefan	8510251233	05.09.2006 v 18:01	4	INTZ	T1H1
Karlavák - detaily	Milan	8512208243	11.05.2007 v 12:50	M1	INTM	T1H1
Prouza - detaily	Pavel	8710024257	14.09.2007 v 19:01	M9	INTM	T1H1
Vocásek - detaily	Květoslav	7801156330	24.09.2007 v 23:01	M9	INTM	T1H1
Vybul - detaily	John	8310025599	25.09.2007 v 02:07	M3	INTM	T1H1
Vomáčka - detaily	Petr	6406152137	25.09.2007 v 15:40	11	CH1J	T3IP
První - detaily	Pacient	8009269862	26.09.2007 v 13:21	M1	INTM	T1H1
Asd - detaily	Asdas	8603084270	28.09.2007 v 00:01	1	PSYL	T3H5
Tománek - detaily	Martin	8011256385	28.09.2007 v 21:01	M1	INTM	T1H1
Pokorný - detaily	Radek	7506160011	29.09.2007 v 00:12	P1	PL1L	T3H6
Pokec - detaily	Jan	1111112222	30.09.2007 v 03:01	M4	INTM	T1H1

Registr: NRHOSP - 10/2007

Příjmení	Jméno	Rodné číslo	Přijat	Pokoj	Stanice	Tarif
Novák - detaily	Petr	8509183551	02.10.2005 v 08:53	M1	INTM	T1H1
Nemeth - detaily	Štefan	8510251233	05.09.2006 v 18:01	4	INTZ	T1H1
Karlavák - detaily	Milan	8512208243	11.05.2007 v 12:50	M1	INTM	T1H1

Copyright © Martin Holub a VUT PEKT Brno 2006-2010
Poslední úprava provedena: 16/05/2010 13:13:40

Obr. 9.8: Vyhledaná data dle zvolených parametrů



Obr. 9.9: Výzva ke změně-opravě zvolených parametrů

Vyhledané záznamy z databáze NIS CLINICOM (viz obr. 9.8) obsahují údaje, jež jsou zapotřebí vždy pro zvolený registr. Například pro registr hospitalizovaných (NRHOSP) jsou záznamy řazeny po měsících a obsahují: příjmení, jméno, rodné číslo pacienta, dále pak datum a čas přijetí k hospitalizaci, pokoj na němž je pacient ubytován, stanice a tarif.

Nalezené záznamy z databáze je možné vytisknout, nebo převést do vzorového kódu v DS MZČR verze 3. Pro tento účel slouží příslušná tlačítka, které nalezneme na konci vypsaných záznamů (viz obr. 9.10).

Při stisku tlačítka „Zobrazit nalezené výsledky v DS 3“ dojde k otevření nového okna, ve kterém je provedena autorizace přístupu (viz obr. 9.11). Pokud je přístup povolen, dojde k vygenerování vzorového kódu v DS MZCR verze 3.

Pokud zvolíme tlačítko „Vytisknout nalezená data“, tak dojde k tisku vyhledaných záznamů. Aplikace je pro tisk ošetřena tak, že nedochází k tisku menu, záhlaví, zápatí, v nalezených výsledcích sekce hlášení jsou tisku ušetřena rovněž tlačítka a odkazy (včetně odkazu „detaily“ u jednotlivých záznamů).



Obr. 9.10: Tlačítka: „Vytisknout nalezená data“ a „Zobrazit nalezené výsledky v DS 3“

Autorizace: ..Přístup povolen..

Nalezená data pro zvolený registr/interval/diagnózu, zobrazená ve formátu DS 3:

```
<?xml version="1.0" encoding="Windows-1250" standalone="no" ?>
<!DOCTYPE data SYSTEM "ds030701.dtd">
<data id_soubor="VUT_UBMI_00_00000_2010-05-17T13:08:02" verze_ds="03.01.01" ur="T" potvrzeni="N">
<zdroj_is_kod_firmy="VUT_UBMI" kod_prog="CSP" verze_prog="xx" licis_prog="xx" />
<pm lco="12345678" >
<as typ="E" >
<obsah>adresat@vutbr.cz</obsah>
```

Obr. 9.11: Autorizace a zobrazení vzorového kódu s datovým standardem verze 3.

DG kód	Popis diagnózy	DG Poprvé	Příjmení	Jméno	Rodné číslo	Datum a čas	Lékař
A30.5	Lepra lepromatózní	Ano	House	Gregory	6207141237	13.10.2008 v 13:41	S26
I15.0	Renovaskulární hypertenze	Ano	House	Gregory	6207141237	22.10.2008 v 09:32	S27
I46.1	Náhle srdeční smrt	Nezjištěno	House	Gregory	6207141237	22.10.2008 v 09:32	S27
J11.1	Chřipka s jinými projevy ne dýchacím ústrojí, virus	Ne	House	Gregory	6207141237	15.12.2008 v 11:50	ST1
K38.0	Hyperplazie appendixu	Ano	House	Gregory	6207141237	01.10.2008 v 09:11	S27
M84.3	Fraktura z námahy, nezařazená jinde	Ano	House	Gregory	6207141237	24.11.2008 v 11:12	ST1
S92.0	Zlomenina palce kosti - fracture osse-calcanis	Ano	House	Gregory	6207141237	01.10.2008 v 09:13	S27
V01.1	Provozní nehoda (silniční)	Ano	House	Gregory	6207141237	01.10.2008 v 09:13	S27

Obr. 9.12: Diagnózy vybraného pacienta

Pokud si přejeme zobrazit informace o zadaných diagnózách, zvolíme možnost detaily u požadované osoby (viz obr. 9.8 a 9.12).

Detailní informace obsahují následující položky:

DG kód - jedná se o kód diagnózy dle Mezinárodní klasifikace nemocí (MKN10).

Popis diagnózy – obsahuje přesný název diagnózy dle MKN10.

DG Poprvé - informuje o tom, zda byla diagnóza zjištěna poprvé.

Možnosti jsou: Ano, Ne, Nezjištěno, Nežadáno.

Za jejich korektnost je odpovědný zadavatel diagnózy (např. lékař).

Příjmení, Jméno, Rodné číslo - jsou příslušné identifikační údaje pacienta.

Datum a čas - informuje o datu a času zadání diagnózy, případně její aktualizaci.

Lékař - obsahuje zkratku zadavatele diagnózy, lékaře, případně, v rámci výuky na UBMI, se jedná o studenty.

V levém spodním rohu, pod výpisem jednotlivých diagnóz, nalezneme tlačítko „Zpět na výsledky hledání“, jenž nás vrátí o krok zpět.

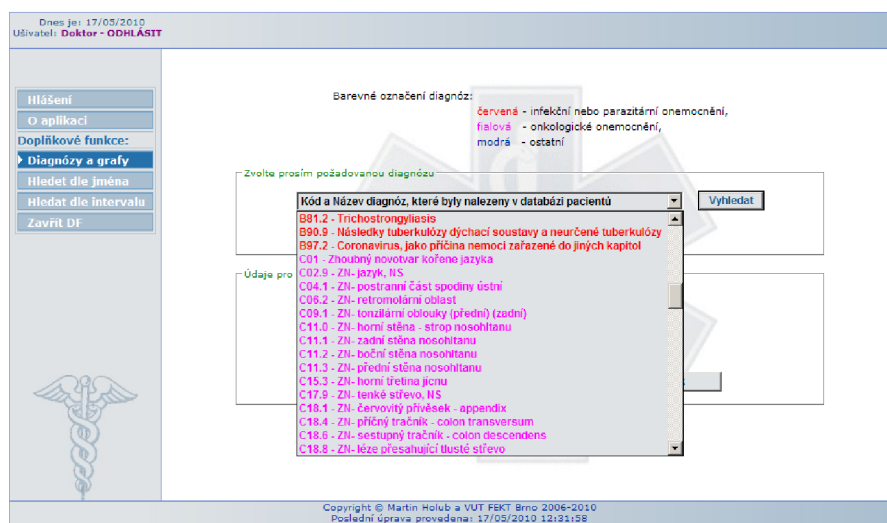
9.1.2 Doplnkové funkce webové aplikace

Druhá část „Doplnkové funkce“ rozšiřuje webovou aplikaci o další užitečné funkce. Konkrétně se jedná o sestavení grafu s výskytem vybrané diagnózy, případně všech infekčních nebo onkologických diagnóz, sestavení grafu s celkovým výskytem jednotlivých diagnóz. Dále pak hledání záznamů dle příjmení. Rovněž lze hledat diagnózy dle zadaného intervalu, filtrovat infekční, onkologické diagnózy a zobrazit některé z nalezených výsledků v datovém standardu MZČR.

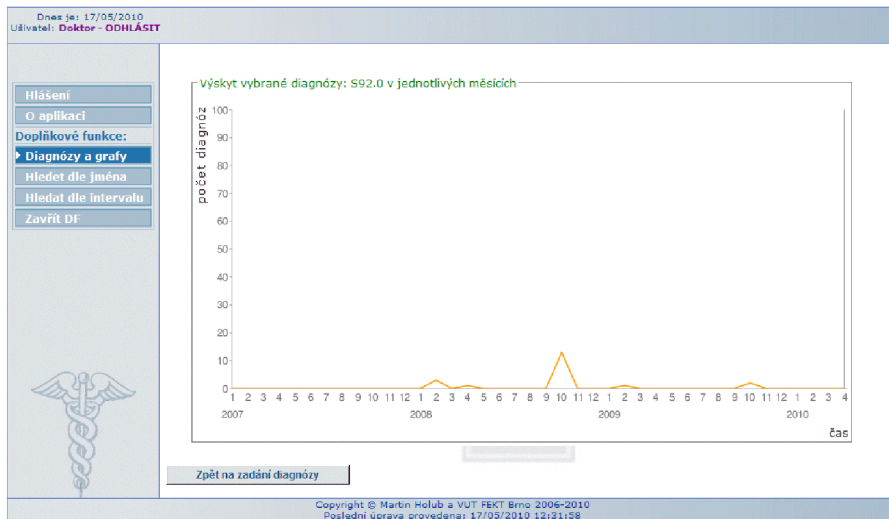
První položkou v menu doplňkových funkcí je: „Diagnózy a grafy“. Jedná se o výběr diagnózy, pro kterou budou zobrazeny nalezené záznamy pacientů, případně generován graf, jenž zobrazuje výskyt vybrané diagnózy v průběhu předchozích let.

Seznam diagnóz (viz obr. 9.13) je generován dynamicky, obsahuje tedy pouze diagnózy, které jsou nalezeny v databázi u jednoho či více pacientů. V případě, že by pacientovi byla zadána diagnóza, která by doposud nebyla nalezena u jiného pacienta, výběr by se automaticky, po novém načtení stránky, o tuto diagnózu rozšířil. Pro přehlednost jsou infekční-parazitární diagnózy označeny červeně a onkologické diagnózy jsou označeny fialově.

Po vybrání požadované diagnózy a stisknutí tlačítka „Vyhledat“, budou zobrazeny záznamy pacientů se zadanou diagnózou. Pokud však chceme místo záznamů zobrazit graf, ve kterém bude zakreslen výskyt vybrané diagnózy (viz obr. 9.14), zaškrtneme „ANO“ u položky „Přeji si zobrazit graf zvolené diagnózy:“.



Obr. 9.13: Ukázka výběru požadované diagnózy



Obr. 9.14: Ukázka výskytu zvolené diagnózy

Tabulka (viz obr. 9.15) nalezených výsledků hledání dle zvolené diagnózy obsahuje totožné položky, jako detailní zobrazení diagnóz vybraného pacienta u povinných hlášení. Jedná se tedy o položky: dg kód, popis diagnózy, dg poprvé, příjmení, jméno, rodné číslo, datum, čas a lékař.

Pod tabulkou, s výsledky hledání zadané diagnózy, je zobrazen počet nalezených záznamů a počet záznamů s rozdílnou diagnózou. Celkový počet záznamů v databázi je dán jejich součtem. Pokud chceme vyhledat jinou diagnózu, můžeme tak učinit po kliknutí na tlačítko „Zpět na zadání diagnózy“. Jestliže si přejeme zobrazit nalezené záznamy ve formátu datového standardu MZČR, klikneme na tlačítko „Zobrazit data ve vzorovém DS“.

DG kód	Popis diagnózy	DG Poprvé	Příjmení	Jméno	Rodné číslo	Datum a čas	Lékař
B20.0	Onemocnění HIV projevující se mykobakteriálními infekcemi	Ano	Cucy	Pan	0602181239	11.03.2009 v 09:37	S25
B20.0	Onemocnění HIV projevující se mykobakteriálními infekcemi	Ano	Žlutěnka	Petr	0410074929	07.10.2009 v 11:04	ST5
B20.0	Onemocnění HIV projevující se mykobakteriálními infekcemi	Nezadáno	Jošt	Fero	0010071919	07.10.2009 v 11:03	ST9

3 - počet nalezených záznamů se zadanou diagnózou
701 - počet nalezených záznamů s rozdílnou diagnózou

Nalezená data pro zvolenou diagnózu: B20.0, zobrazená ve formátu DS3:

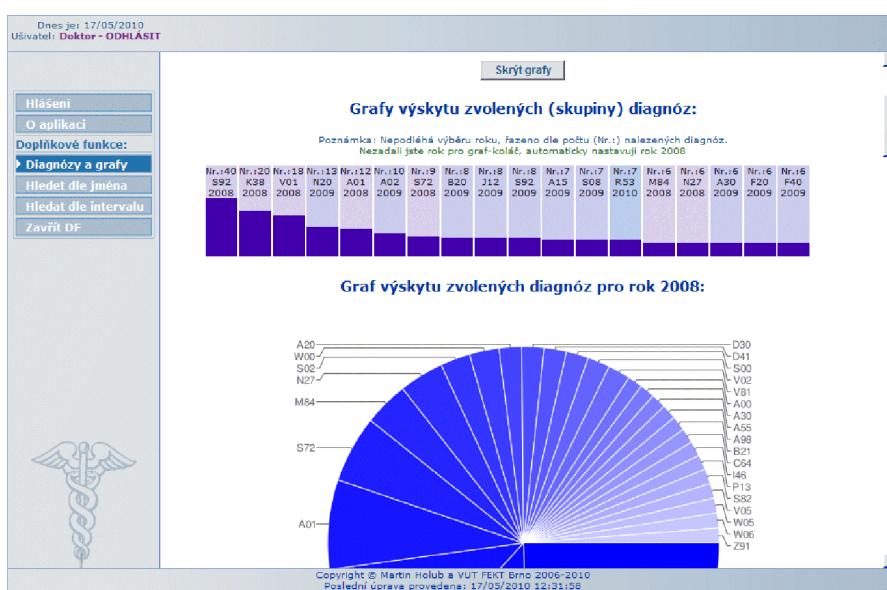
```
<?xml version="1.0" encoding="Windows-1250" standalone="no" ?>
<DOCTYPE data SYSTEM "ds30701.dtd">
<data id_soubor="VUT_UBMI_00_00000_2010-05-17T13:24:51"
verze_ds="03.01.01" ur="T" potvrzeni="N">
<zdroj_is_kod_firmy="VUT_UBMI_kod_prog="CSP" verze_prog="30x"
Copyright © Martin Holub a VUT FKT Brno 2006-2010
Poslední úprava provedena: 17/05/2010 12:31:58
```

Obr. 9.15: Ukázka výsledku hledání dle zadané diagnózy a vygenerování vzorového DS

Vygenerovaný kód obsahuje údaje z výše umístěné tabulky s výsledky hledání. Jednotlivé bloky datového standardu jsou vypsány dle specifikace verze 3.01.01.

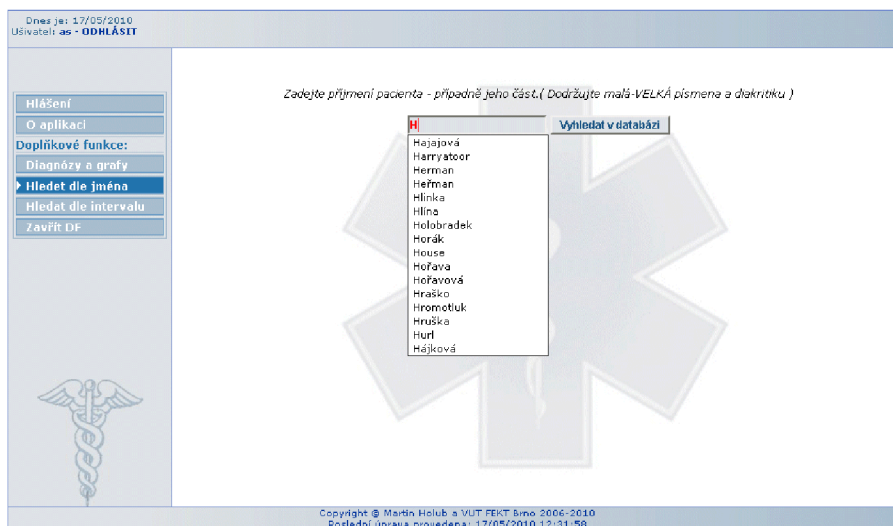
Hlavička je pouze ukázková. Záznamy o pacientech jsou zobrazeny dle zadaných kritérií a nalezených údajů. Pokud by u pacienta byla nalezená diagnóza zadaná více než jednou, bude o ni blok pacienta automaticky doplněn, aniž by byl ukončen a opětovně otevřen. Zobrazený kód je možné zavřít tlačítkem „Zavřít vzorové DS“, nebo případným odhlášením.

Pokud bychom si přáli zobrazit graf všech diagnóz, případně pouze infekčních či onkologických, nevybereme žádnou diagnózu, ale zvolíme rok a stiskneme tlačítko „Zobraz grafy pro zvolený rok“. Výsledkem jsou dva grafy, první (sloupcový) zobrazuje celkový výskyt diagnóz (jednotlivé roky jsou od sebe rozlišeny změnou barvy pozadí), druhý (koláčový) graf zobrazuje výskyt diagnóz pouze pro zvolený rok. Oba grafy jsou seřazeny od nejvyššího výskytu po nejnižší (viz obrázek 9.16). Pokud si přejeme zobrazit jiný graf, stiskneme tlačítko „Skrýt grafy“ a provedeme nový výběr.



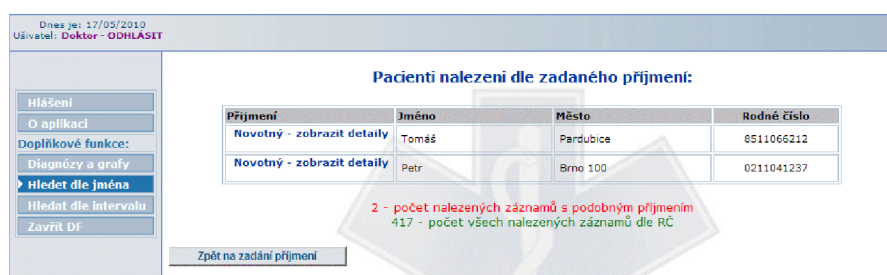
Obr. 9.16: Grafy výskytu zvolené skupiny diagnóz

Druhou položkou v menu doplňkových funkcí je: „Hledat dle jména“. Zde je umožněno vyhledat pacienta z databáze NIS CLINICOM dle příjmení. Všechna nalezená příjmení v databázi jsou předem dynamicky generována do skriptu, který zjednodušuje výběr konkrétního pacienta takzvaným našeptáváním (viz obr. 9.17).

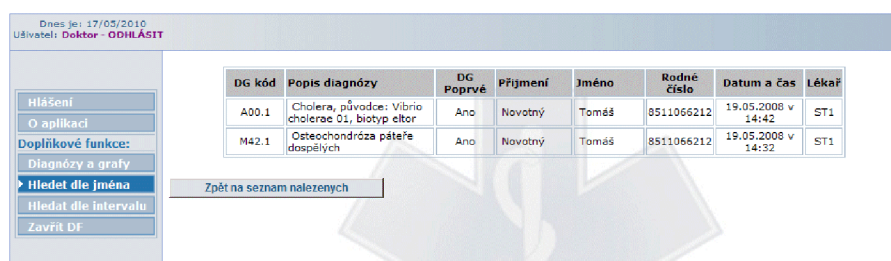


Obr. 9.17: Vyhledání pacienta z NIS CLINICOM dle příjmení

Po stisku tlačítka „Vyhledat v databázi“ dojde k vyhledání příjmení pacienta, případně pacientů s vybraným příjmením. Poté vybereme konkrétního pacienta, například dle jména, bydliště nebo rodného čísla. Teprve pak je zobrazen kompletní seznam nalezených diagnóz u námi zvoleného pacienta (viz obrázek 9.18 a 9.19).



Obr. 9.18: Výběr konkrétního pacienta ze seznamu



Obr. 9.19: Diagnózy vybraného pacienta

Poslední položkou v menu doplňkových funkcí je „Hledat dle intervalu“ viz obrázek 9.20. V této části programu můžeme vyhledávat diagnózy v zadaném intervalu. Pokud nezadáme jeden z intervalů, případně oba, bude interval doplněn automaticky stejně tak, jako je tomu u povinných hlášení viz výše.

Dále je výběr diagnózy rozdělen na dvě části.

V první části lze vybrat konkrétní diagnózu nebo jen počátek kódu diagnóz, které mají být vyhledány. Zde je nám opět ulehčen výběr díky dynamicky generovanému našeptávání. Je tedy možné vyhledat všechny záznamy pro diagnózy začínající například kódem A3 atd. V druhé části lze vybrat předdefinovaný rozsah diagnóz (všechny, infekční, onkologické).

Dnes je: 17/05/2010
Uživatel: Doktor - ODHLÁSIT

Hlášení
O aplikaci
Doplňkové funkce:
Diagnózy a grafy
Hledat dle jména
Hledat dle intervalu
Zavřít DF

Zde máte možnost zadat konkrétní interval (nepovinný údaj)
Interval od: rok měsíc do: rok měsíc
Poznámka: Zvolte požadovaný rozsah od-do u nevoleného údaje se bere automaticky minimum/maximum.

Výběr diagnózy (nepovinný údaj)
Kód a Název diagnóz, které byly nalezeny v databázi pacientů
počátek kódu diagnózy Vyhledat zadanou diagnózu v intervalu

Nepodílejte výběru diagnózy - hledá se dle intervalu
Vyhledat infekční diagnózy dle intervalu
Vyhledat onkologické diagnózy dle intervalu
Vyhledat VŠECHNY použité diagnózy dle intervalu

Copyright © Martin Holub a VUT FEKT Brno 2006-2010
Poslední úprava provedena: 17/05/2010 12:31:58

Obr. 9.20: Hledání diagnóz dle intervalu

Nalezené záznamy lze rovněž zobrazit v datovém standardu MZČR verze 3.

Pokud si přejeme zavřít položky doplňkových funkcí v menu, zvolíme tlačítko „Zavřít DF“. Jak již bylo zmíněno, tak aplikaci můžeme opustit kdykoliv stiskem tlačítka „ODHLÁSIT“ (viz obrázek 9.21).

Vítejte ve webové aplikaci určené k přístupu
do nemocničního informačního systému CLINICOM
na ÚBMI, FEKT, VUT v Brně

Proběhlo úspěšné odhlášení uživatele.

Zadejte Jméno Zadejte Heslo Přihlásit

Copyright © Martin Holub a VUT FEKT Brno 2006-2010

Obr. 9.21: Odhlášení uživatele z webové aplikace

Závěr

V diplomové práci jsme se nejprve zaměřili na poměrně složitou strukturu bloků datového standardu Ministerstva zdravotnictví České republiky a jeho postupný vývoj. Rovněž jsme se zabývali značkovacími jazyky od jejich vzniku až po současné využití v datovém standardu spolu s definicí typu dokumentu. Právě spojení značkovacího jazyka XML a DTD plně splňuje potřeby datového standardu.

V práci jsme popsali základní kroky vedoucí k zabezpečení serveru s nemocničním informačním systémem. Problematika zabezpečení byla řešena i pro komunikaci mezi jednotlivými zdravotnickými organizacemi. Zde se při datové komunikaci snažíme dosáhnout alespoň takové úrovně zabezpečení, aby byla zaručena stejná důvěryhodnost a integrita dat jako při listinné komunikaci.

Poté jsme stručně popsali možnosti přístupu do nemocničního informačního systému CLINICOM firmy SMS pomocí grafického rozhraní CareCenter, NetAccess, včetně textového/terminálového rozhraní. Terminálové rozhraní sice nenabízí takový komfort jako rozhraní grafické, ale působí rovněž přehledně a uceleně.

Dále jsme se v práci zaměřili na databázovou platformu Caché včetně poskytovaných utilit. Právě databázi Caché využívá NIS CLINICOM pro ukládání svých dat. Z utilit Caché jsme si podrobněji ukázali ty, které jsme použili při realizaci naší webové aplikace.

K realizaci webové aplikace jsme využili jednoduchého webového serveru s podporou CSP, jenž je součástí instalačního balíčku Caché americké firmy InterSystems.

Tímto nám odpadla nutnost instalace dalšího webového serveru.

Stávající konfiguraci serveru s databází Caché bylo potřeba upravit tak, aby podporovala české znaky. Úpravou konfigurace došlo k jejich správnému zobrazování v záznamech, které jsme získali SQL dotazem z databáze NIS CLINICOM. Znaky s českou diakritikou, které byly zapsány přímo ve zdrojovém kódu jazyka CSP pomocí Caché Studia, se bohužel nezobrazovaly korektně. Problém se zobrazováním českých znaků v celé webové aplikaci byl vyřešen použitím HTML entit, které jsou součástí standardu HTML a slouží k reprezentaci znaků nezávisle na nastavení kódové stránky.

Výsledkem této práce je plně funkční zabezpečená dynamická webová aplikace realizovaná pomocí jazyka CSP. Oprávněnému uživateli umožňuje aplikace vyhledat a zobrazit záznamy pacientů z databáze NIS CLINICOM dle námi zvoleného Národního zdravotního registru (NRZ), zvoleného kódu diagnózy (Mezinárodní klasifikace nemocí verze 10), příjmení nebo časového intervalu. Nalezené záznamy je umožněné vytisknout za účelem archivace, nebo je převést do vzorového kódu datového standardu Ministerstva zdravotnictví České republiky.

Aplikace rovněž umožňuje zobrazení grafu výskytu zvolené diagnózy, případně celkový roční přehled infekčních nebo onkologických diagnóz. Všechna patientská data využívaná při řešení této práce byla získána ze školní databáze NIS CLINICOM VUT Brno.

Použitá literatura

- [1] *Česká společnost zdravotnické informatiky a vědeckých informací* [online].c1996, [cit. 2010-01-05].<<http://cszivi.cls.cz/doc/standroz.htm>>
- [2] *Standardy informatiky a telematiky ve zdravotnictví* [online].c2004, [cit.2010-01-9].<http://labe.felk.cvut.cz/~falc/IKTZ_2004L/slides/Standardy.html>
- [3] *Divize SLP® Správná Laboratorní Práce* [online].[cit. 2010-03-05].<http://www.sekk.cz/SLP/SLP_DS_NCLP_info.htm>
- [4] *Datový standard MZ ČR 03.08.01* [CD-ROM].Praha, 2006.[cit. 2010-04-05]. Adresář: DS_3/ds/ds_definice_1.rtf.
- [5] *Seznam základních dokumentů a textů k DS a NČLP* [online].[cit. 2010-12-20].<<http://www.mzd.cz/data/c764/lib/dsbck.htm>>
- [6] *Seriál o XML pro Softwarové noviny* [online].[cit. 2010-04-10].<<http://www.kosek.cz/clanky/sw-nxml>>
- [7] Mlýnková I., et al.: *XML technologie. Principy a aplikace v praxi*. Grada Publishing, a.s. Praha, 2008. ISBN 978-80-247-2725-7.
- [8] *Ústav zdravotnických informací a statistiky ČR* [online].[cit. 2010-03-20].<http://www.uzis.cz/info.php?article=13&mnu_id=7300>
- [9] *Datový standard MZ ČR 03.13.01* [CD-ROM].Praha, 2009.[cit. 2010-04-04]. Adresář: DS_3/hypertext/indexdsbbe.htm.
- [10] Hanáček P., Staudek J.: *Definice bezpečnostních funkcí pro předávání dat ve zdravotnictví*, Ministerstvo zdravotnictví České republiky, 1998.
- [11] Zámečník M., Papp R., Seiner M., Kosiner I.: *Problematika ochrany zdravotnických dat, klasifikace citlivosti zdravotnických dat a doporučené bezpečnostní funkce pro jejich přenos*,

Ministerstvo zdravotnictví České republiky, 1998.

- [12] Dostálek L a kol.: *Velký průvodce protokoly TCP/IP Bezpečnost*. Computer Press. Praha, 2003.

- [13] *SMS Informační technologie pro české a slovenské zdravotnictví* [online].[cit. 2008-12-23]. <<http://www.smed.cz>>

- [14] *Magazín Informačních Technologií* [online].[cit. 2010-03-20]. <<http://reboot.cz/howto/databaze/odbc-v-kostce>>

- [15] Kursten W., Ihringer M., Kühn M., Röhring B.: *Caché: Databáze postrelačního typu a tvorba aplikací*. Přeložil Kocan M., Havel R., Pupík J. 1.vydání. Brno: CP Books, a.s., 2005. 400s. ISBN 80-251-0491-5.

- [16] Žák K.: *Modelování databází – Root.cz* [online].[cit. 2010-03-20]. <<http://www.root.cz/clanky/modelovani-databazi/>>

- [17] *Normální formy – NaWEBka* [online].[cit. 2010-04-05]. <<http://www.rydval.cz/phprs/view.php?cislocclanku=2005123127>>

- [18] Intersystems. *Using Caché Server Pages (CSP)* [online]. 2005, [cit. 2010-02-09]. <<http://www.intersystems.com/cache/documentation/cache5docs/PDFS/GCSP.pdf>>

- [19] Zvárová J., Hanzlíček P. a kol.: *Biomedicínská informatika I. – Základy informatiky pro biomedicínu a zdravotnictví*. EuroMISE. Praha, 2002. ISBN 80-246-0609-7.

- [20] *Nemocniční informační systém Clinicom*. Příručka firmy SMS, Brno, 2002.

Abecední seznam použitých zkratk

ANSI	-	American National Standards Institute
ASP	-	Active Server Pages
CSP	-	Caché Server Pages
CSS	-	Cascading Style Sheets – Kaskádové styly
ČNR	-	České národní rada
ČSR	-	Československá republika
DDL	-	Data Definition Language
DLL	-	Dynamic Linking Library
DML	-	Data Manipulation Language
DOS	-	Disk Operating System - Diskový Operační Systém
DQL	-	Data Query Language
DRG	-	Diagnosis related group
DS	-	Datový Standard
DSS	-	Decision Support System
DTD	-	Document Type Definition - Definice Typu Dokumentu
EDIFACTS	-	Electronic Data Interchange For Administration
EU	-	Evropská Unie
GML	-	Generalized Markup Language
HL7	-	Health Level Seven
HTML	-	Hypertext Markup Language - Hypertextový značkovací jazyk
IP	-	Internet Protocol
IT	-	Information Technology – Informační Technologie
IVF	-	In vitro fertilizace – Mímotělní-umělé oplodnění
JDBC	-	Java DataBase Connectivity
LAN	-	Local Area Network – Lokální-místní síť
MD5	-	Message-Digest Algorithm
MKN	-	Mezinárodní Klasifikace Nemocí
MZ	-	Ministerstvo zdravotnictví
MZČR	-	Ministerstva zdravotnictví České republiky
MZSV	-	Ministerstvo zdravotnictví a sociálních věcí
NČLP	-	Národní Číselník Laboratorních Položek
NIS	-	Nemocniční Informační Systém
NKCHR	-	Národní kardiologický registr
NOR	-	Národní onkologický registr
NRAR	-	Národní registr asistované reprodukce
NRCCH	-	Národní registr cévní chirurgie
NRHOSP	-	Národní registr hospitalizovaných
NRKI	-	Národní registr kardiiovaskulárních intervencí
NRKN	-	Národní registr kloubních náhrad
NRNAR	-	Národní registr novorozenců
NRPN	-	Národní registr nemocí z povolání
NRPOT	-	Národní registr potratů
NRULISL	-	Národní registr uživatelů lékařsky indikovaných substitučních látek
NRROD	-	Národní registr rodiček

NRVV	-	Národní registr vrozených vad
NZIS	-	Národní Zdravotnický Informační Systém
NZR	-	Národní Zdravotní Registry
ODBC	-	Open DataBase Connectivity
OECD	-	Organisation for Economic Co-operation and Development
PC	-	Personal Computer – Osobní počítač
PDA	-	Personal Digital Assistant - Osobní digitální pomocník
PGP	-	Pretty Good Privacy
PHP	-	Personal Home Page - Hypertext Preprocessor
RLZF	-	Registr lékařů, zubních lékařů a farmaceutů
SGML	-	Standard Generalized Markup Language
SHA	-	Secure Hash Algorithm
SQL	-	Structured Query Language
SŘBD	-	Systém Řízení Báze Dat
SSH	-	Secure Shell
SSL	-	Secure Socket Layer
S/MIME	-	Secure/Multipurpose Internet Mail Extensions
TCP	-	Transmission Control Protocol
UBMI	-	Ústav biomedicínkého inženýrství
UDA	-	Unified Data Architecture
UPT	-	Uměle přerušené těhotenství
USA	-	United States of America – Spojené státy americké
ÚZIS	-	Ústav Zdravotnických Informací a Statistiky
WAN	-	Wide Area Network
WHO	-	World Health Organisation - Světová zdravotnická organizace
XML	-	eXtensible Markup Language - Rozšiřitelný značkovací jazyk

Příloha č.1 - Zdrojový kód webové aplikace

Soubor DPMHolub.csp:

```
<script LANGUAGE="CACHE" RUNAT="SERVER">
  //deklarace jmena souboru
  set namef="DPMHolub.csp"
</script>
<html><head>
<title>Datový Standard MZ &#268;R - Diplomová Práce VUT Brno 2010 </title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250" />
<CSP:COMMENT> <cs:include Page="x.csp"> </CSP:COMMENT>
<cs:class private=0 encoded=1> <!-- encoded 0=no, 1=yes 1,2=yes -->
<style type="text/css"><!--
table { font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12px; font-style: normal; }
table.tb1 { border-style: solid; border-width: 0px; border-color:#2470AA; font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 11px; font-style: normal; }
table.tb1 td { border-style: solid; border-width: 0px; border-color:#2470AA; font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 11px; font-style: normal; }
table.tb2 { border-style: solid; border-width: 1px; border-color:#A9BECF;font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 11px; font-style: normal; }
table.tb2 td { border-style: solid; border-width: 1px; border-color:#A9BECF;font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 11px; font-style: normal; }
input { background-color: #DFE3E6; font-family: Arial, Helvetica, sans-serif; font-size: 12px; font-weight: bold; color:#195481; text-decoration: none; }
select { border: 1px #000033 solid; background-color: #DFE3E6; font-family: Arial, Helvetica, sans-serif; font-size: 12px; font-weight: bold; color:#195481; text-decoration: none; }
BODY { SCROLLBAR-FACE-COLOR: #DFE3E6; SCROLLBAR-HIGHLIGHT-COLOR: #DFE3E6; SCROLLBAR-SHADOW-COLOR: #003399; SCROLLBAR-3DLIGHT-COLOR: #ffffff; SCROLLBAR-ARROW-COLOR: #003399; SCROLLBAR-TRACK-COLOR: #ffffff; SCROLLBAR-DARKSHADOW-COLOR: #003399;
font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 12px; font-style: normal;
}
H1,H2,H3,H4 { color: #003399; font-family: Verdana, Arial, Helvetica, sans-serif; font-style: normal; font-weight: bold }
H1 { font-size: 20px; } H2 { font-size: 16px; } H3 { font-size: 14px; } H4 { font-size: 11px; }
a:link { color: #003399; text-decoration:none; font-weight: bold }
a:visited {text-decoration:none; font-weight: bold}
a:hover {color: red; text-decoration:none; font-weight: bold }
/* styly pro autocomplete */
.autocomplete {
font-family: Verdana;
font-size: 8pt;
background-color: white;
border: 1px solid black;
position: absolute;
cursor: default;
overflow: auto;
overflow-x: hidden;
}
.autocomplete_item {
padding: 1px;
padding-left: 5px;
color: black;
width: 100%;
}
.autocomplete_item_highlighted {
padding: 1px;
padding-left: 5px;
color: white;
background-color: #0A246A;
}
/* styly pro graf */
#graf2 dl { background: #fff; width: 700px; }
#graf2 dt { text-align: center; }
#graf2 dd { margin: 10 1px; width: 35px; height: 100px; background: #40a; float: left; }
#graf2 dd div.popisek { display: block; background: #ace; text-align: center; font-size: 80%; }
#graf2 dd.p1 div.popisek { height:98%; }
#graf2 dd.p2 div.popisek { height:95%; }
```

```
#graf2 dd.p3 div.popisek { height:92%; }  
#graf2 dd.p4 div.popisek { height:90%; }  
#graf2 dd.p5 div.popisek { height:88%; }  
#graf2 dd.p6 div.popisek { height:85%; }  
#graf2 dd.p7 div.popisek { height:82%; }
```

```

#graf2 dd.p8 div.popisek { height:80%; }
#graf2 dd.p9 div.popisek { height:78%; }
#graf2 dd.p10 div.popisek { height:75%; }
#graf2 dd.p11 div.popisek { height:72%; }
#graf2 dd.p12 div.popisek { height:70%; }
#graf2 dd.p13 div.popisek { height:68%; }
#graf2 dd.p14 div.popisek { height:65%; }
#graf2 dd.p15 div.popisek { height:62%; }
#graf2 dd.p16 div.popisek { height:60%; }
#graf2 dd.p17 div.popisek { height:58%; }
#graf2 dd.p18 div.popisek { height:55%; }
#graf2 dd.p19 div.popisek { height:52%; }
#graf2 dd.p20 div.popisek { height:50%; }
#graf2 dd.p21 div.popisek { height:48%; }
#graf2 dd.p22 div.popisek { height:45%; }
#graf2 dd.p23 div.popisek { height:42%; }
#graf2 dd.p24 div.popisek { height:40%; }
#graf2 dd.p25 div.popisek { height:38%; }
#graf2 dd.p26 div.popisek { height:35%; }
#graf2 dd.p27 div.popisek { height:32%; }
#graf2 dd.p28 div.popisek { height:30%; }
#graf2 dd.p29 div.popisek { height:28%; }
#graf2 dd.p30 div.popisek { height:25%; }
#graf2 dd.p31 div.popisek { height:22%; }
#graf2 dd.p32 div.popisek { height:20%; }
div.cl /* cistic */ { clear: both; }
table.popisek {
width: 680px;
margin: 0 auto 0 auto;
border: 1px solid black;
border-collapse: collapse;}
/* styly pro layout */
#hlavni {
width: 980px;
margin: 0 auto 0 auto;
border: 1px solid #718BCD;}
#inout {
margin: 30px auto ;
height: 400px;
width: 700px;
background-color: #DFE3E6;
border: 1px solid #718BCD;
background-image:url("");background-repeat: repeat-y; background-position:right top;
overflow: auto;}
#head {
height: 40px;
background-color: #DFE3E6;
border-bottom: 1px solid #718BCD;
background-image:url("");background-repeat: repeat-y; background-position:right top;}
#leftmenu {
float: left;
width: auto;
height: 500px;
border-right: 1px solid #718BCD;
background-color: #DFE3E6;
background-image:url("");background-repeat: no-repeat; background-position:center bottom;}
#obsah {
height: 500px;
overflow: auto;
background-image:url("");background-repeat: no-repeat; background-position:center center;}
#end {
height: 30px;
border-top: 1px solid #718BCD;
background-color: #DFE3E6;
background-image:url("");background-repeat: repeat-y; background-position:right bottom;
clear: both;}
/* styly pro tisk */
#print { display: none;}
@media print {
div#hlavni {border: 0}
div#head {display: none;}
div#noprint {display: none;}
div#leftmenu {display: none;}
div#end {display: none;}
div#print {display: block;}
div#obsah {overflow: visible; height: auto; background-image:none }

```



```
<li><a href="#"(namef)?jmeno=#(%request.Data("jmeno",1))#&heslo=#(%request.Data("heslo",1))#>Zav&#345;it DF</a></li>
<CSP:ELSE>
<li><a href="#"(namef)?jmeno=#(%request.Data("jmeno",1))#&heslo=#(%request.Data("heslo",1))#&obsah=df">Dopl&#328;ková
funkce</a></li>
</CSP:IF> </ul> </div>
  <div id="obsah"> <br>
<CSP:COMMENT> obsah dle DIAG .. obsah=d </CSP:COMMENT>
<CSP:IF CONDITION=$G(%request.Data("obsah",1))["dfd"]">
<CSP:COMMENT> pokud není zvolena diagnóza, umožní jí zvolit</CSP:COMMENT>
<CSP:IF CONDITION=$G(%request.Data("zvoldiag",1))="">
  <br><br><table class="tbl1"><tr><td>Barevné ozna&#269;ení diagnóz:
</td><td>&nbsp;</td><tr><td>&nbsp;</td><td><font color="red">&#269;erVENÁ</font> - infek&#269;ní nebo parazitární
onemocn&#283;ní,</td><tr><td>&nbsp;</td><td><font color="fuchsia">fialová</font> &nbsp;</td><td>&nbsp;</td><td><font color="modrá">modrá</font> &nbsp;</td><td>&nbsp;</td><td><font color="#003399">modrá</font> &nbsp;</td><td>&nbsp;</td><tr><td>&nbsp;</td><tr></table><br>
  <CSP:COMMENT> vyberovej select.. vygeneruje roky, jenž DB obsahuje </CSP:COMMENT>
<SCRIPT LANGUAGE="SQL" NAME="kodydrok">
SELECT SUBSTRING(Aenderungsdatum,1,4) AS drok
from Diagnosegruppe Group by SUBSTRING(Aenderungsdatum,1,4) order by SUBSTRING(Aenderungsdatum,1,4) ASC
</SCRIPT>
<CSP:COMMENT> vyberovy select .. spojí diag a vcid10, seskupí diag group by, vytvoření cyklu pro option</CSP:COMMENT>
<SCRIPT LANGUAGE="SQL" NAME="kodyd">
SELECT DGICD10, ICICD10, ICPopis
FROM SQLUser.VDiagnoza
LEFT JOIN SQLUser.VICD10KS ON dgicd10=ICICD10 GROUP by ICICD10 Order by dgicd10
</SCRIPT>
<form name="form" method="post" action="#"(namef)">
<fieldset style="width: 650px;"><legend style="color: green; font-size: 80%;">Zvolte prosím po&#382;adovanou
diagnózu</legend><br>
  <select name="zvoldiag">
    <option style="color: black;" value="">Kód a Název diagnóz, které byly nalezeny v databázi pacient&#367;</option>
    <CSP:WHILE CONDITION=kodyd.Next()>
      <CSP:IF CONDITION=(kodyd.Data("DGICD10"))[" A"]">
        <option style="color: red;" value="#"(kodyd.Data("DGICD10"))#>#(kodyd.Data("DGICD10"))# -
#(kodyd.Data("ICPopis"))#</option>
      <CSP:ELSE>
        <CSP:IF CONDITION=(kodyd.Data("DGICD10"))[" B"]">
          <option style="color: red;" value="#"(kodyd.Data("DGICD10"))#>#(kodyd.Data("DGICD10"))# -
#(kodyd.Data("ICPopis"))#</option>
        <CSP:ELSE>
          <CSP:IF CONDITION=(kodyd.Data("DGICD10"))[" C"]">
            <option style="color: fuchsia;" value="#"(kodyd.Data("DGICD10"))#>#(kodyd.Data("DGICD10"))# -
#(kodyd.Data("ICPopis"))#</option>
          <CSP:ELSE>
            <CSP:IF CONDITION=(kodyd.Data("DGICD10"))[" D"]">
              <option style="color: fuchsia;" value="#"(kodyd.Data("DGICD10"))#>#(kodyd.Data("DGICD10"))# -
#(kodyd.Data("ICPopis"))#</option>
            <CSP:ELSE>
              <option value="#"(kodyd.Data("DGICD10"))#>#(kodyd.Data("DGICD10"))# - #(kodyd.Data("ICPopis"))#</option>
            </CSP:IF>
          </CSP:IF>
        </CSP:IF>
      </CSP:WHILE>
    </select>
    #(.InsertHiddenField("#(namef)", "jmeno", (%request.Data("jmeno",1))))#
    #(.InsertHiddenField("#(namef)", "heslo", (%request.Data("heslo",1))))#
    #(.InsertHiddenField("#(namef)", "obsah", (%request.Data("obsah",1))))#
    <input type="submit" name="Submit" value="Vyhledat">
    <br><br> P&#345;ejí si zobrazit graf zvolené diagnózy:
    <input type="checkbox" name="graf" value="Ano"> - ANO <br><br>
  </fieldset> <br><br>
  <CSP:IF CONDITION=$G(%request.Data("Tgrafy",1))="">
  <fieldset style="width: 650px;">
  <legend style="color: green; font-size: 80%;">Údaje pro graf - dle zvoleného roku</legend><br>
  <table><tr><td>
    <input type="radio" name="Tgrafvyber" value="2" checked> - Výb&#277;r infek&#269;ních diagnóz
  </td><tr><td>
    <input type="radio" name="Tgrafvyber" value="3"> - Výb&#277;r onkologických diagnóz
  </td><tr><td>
    <input type="radio" name="Tgrafvyber" value="1"> - Výb&#277;r v&#353;ech diagnóz - group
  </td></tr></table><br>
  <select name="TgrafROK">
    <option style="color: black;" value="">Zvolte po&#382;adovaný rok: </option>
    <CSP:WHILE CONDITION=kodydrok.Next()>
      <option style="color: black;" value="#"(kodydrok.Data("drok"))#>#(kodydrok.Data("drok"))#</option>
    </CSP:WHILE>
  </select>
</form>
```



```

        for gm=1:1:12 { // deklarace mesicu
if pomcar'=0 write "," // omezeni aby nedelal carku pred prvni hodnotou
write grafmat(gr, gm)
set pomcar=pomcar+1
    }
    else { // pokud je nacten aktualni rok omezi mesice do akt-1
        for gm=1:1:(mesakt-1) { // deklarace mesicu
if pomcar'=0 write "," // omezeni aby nedelal carku pred prvni hodnotou
write grafmat(gr, gm)
set pomcar=pomcar+1
        }
    }
}
write "&chx|=0:"
    // generovani mesicu na osu x
    for gr=grafodr:1:grafdor { // 3-koncovy rok.gen.
if gr'=grafdor { // pokud neni aktualni rok...
for gm=1:1:12 { // deklarace mesicu
write "|" // omezeni aby nedelal carku pred prvni hodnotou
write gm
    }
}
else { // pokud je nacten aktualni rok omezi mesice do akt-1
    for gm=1:1:(mesakt-1) { // deklarace mesicu
write "|" // omezeni aby nedelal carku pred prvni hodnotou
write gm
    }
}
}
// generovani roku na osu x
write "|2:|3:"
    for gr=grafodr:1:grafdor { // 3-koncovy rok.gen.
if gr'=grafdor { // pokud neni aktualni rok...
for gm=1:1:12 { // deklarace mesicu
write "|" // omezeni aby nedelal carku pred prvni hodnotou
if gm=1 { write gr }
    }
}
else {
    for gm=1:1:(mesakt-1) { // deklarace mesicu
write "|" // omezeni aby nedelal carku pred prvni hodnotou
if gm=1 { write gr }
    }
}
}
write "></td></tr><tr><td> </td><td align=right>&#269;as </td></tr></table>"
</script></fieldset>
</td></tr>
</table>
<br>
<CSP:COMMENT> BIND - tlacitka zpět </CSP:COMMENT>
<div align="left">
<form name="form1" method="post" action="#(namef)#">
<input type="hidden" name="zvoldiag" value="">
#(.InsertHiddenField("#(namef)#", "jmeno", (%request.Data("jmeno", 1))))#
#(.InsertHiddenField("#(namef)#", "heslo", (%request.Data("heslo", 1))))#
#(.InsertHiddenField("#(namef)#", "obsah", (%request.Data("obsah", 1))))#
&nbsp;<input type="submit" name="zpet" value="Zp&#283t na zadání diagnózy">
</form>
</div>
<CSP:COMMENT> Jestliže je vybrána diagnóza a ne grafy . pokračuje.... </CSP:COMMENT>
<CSP:ELSE>
<center><h2>Pacienti nalezeni dle zadané diagnózy:</h2></center>
<table class="tb1" width="85%">
<td><table class="tb2" width="100%" align="center">
<tr style="color: black; background: silver;">
<td align="center"><b>DG kód</b></td>
<td align="center"><b>Popis diagnózy</b></td>
<td align="center"><b>DG Poprvé</b></td>
<td align="center"><b>P&#345;ijmení</b></td>
<td align="center"><b>Jméno</b></td>
<td align="center"><b>Rodné &#269;islo</b></td>
<td align="center"><b>Datum a &#269;as </b></td>
<td align="center"><b>Léka&#345;</b></td>
</tr>
<script LANGUAGE="CACHE" RUNAT="SERVER">
//deklarace pocitadla a prace s oprakovanim dg
set a=0
set b=0
set pacopak=0
set zakoncitip="ne"
</script>
<CSP:COMMENT> Sloucení diagnóz tentokrát bez group by, sloučení diag. s pacienty, vyber pacientu podle diag </CSP:COMMENT>
<SCRIPT LANGUAGE="SQL" NAME="diagnoz">

```

```

SELECT SUBSTRING(DGDIAGNOZA,1,3) AS DGZ, DGSkupinaDiagnoz, DGPoprve, DGICD10, ICICD10, ICPopis,
Aenderungsdatum, geaendert_durch,
SUBSTRING(Aenderungsdatum,1,4) AS drok, SUBSTRING(Aenderungsdatum,5,2) AS dmes, SUBSTRING(Aenderungsdatum,7,2)
AS dden,
SUBSTRING(Aenderungsdatum,9,2) AS dhod, SUBSTRING(Aenderungsdatum,11,2) AS dmin, PACisloRodne,PAJmeno,PAPrijeni
FROM SQLUser.VDiagnoza
LEFT JOIN SQLUser.VICD10KS ON dgid10=ICICD10
LEFT JOIN SQLUser.Vpacient ON PAPACIENT=SUBSTRING(DGDIAGNOZA,1,3)
LEFT JOIN SQLUser.Diagnosegruppe ON DGSkupinaDiagnoz=Diagnosegruppe ORDER by ICICD10
</SCRIPT>
<CSP:WHILE CONDITION=diagnoz.Next()>
  <CSP:IF CONDITION=(diagnoz.Data("DGICD10"))=$G(%request.Data("zvoldiag",1))>
    <script LANGUAGE="CACHE" RUNAT="SERVER">
      // Uprava poprve nul nezadana, 0 nezjistena, 1 Ano, 2 Ne
      if (diagnoz.Data("DGPoprve")="0" { set poprve="Nezji&#353;t&#283no" }
      elseif (diagnoz.Data("DGPoprve")="1" { set poprve="Ano" }
      elseif (diagnoz.Data("DGPoprve")="2" { set poprve="Ne" }
      else { set poprve="Nczadáno" }
    </script>
  <tr>
    <td align="center" width="10%">#(diagnoz.Data("DGICD10"))#</td>
    <td width="25%">&nbsp;#(diagnoz.Data("ICPopis"))# &nbsp;</td>
    <td align="center" width="10%">#(poprve)#</td>
    <td width="13%">&nbsp;#(diagnoz.Data("PAPrijeni"))#</td>
    <td width="12%">&nbsp;#(diagnoz.Data("PAJmeno"))#</td>
    <td align="center" width="10%">#(diagnoz.Data("PACisloRodne"))#</td>
    <td align="center" width="15%">#(diagnoz.Data("dden"))#.#(diagnoz.Data("dmes"))#.#(diagnoz.Data("drok"))# v
    #(diagnoz.Data("dhod"))#.#(diagnoz.Data("dmin"))#</td>
    <td align="center" width="5%">#(diagnoz.Data("geaendert_durch"))#</td>
  </tr>
  <script LANGUAGE="CACHE" RUNAT="SERVER">
    set b=b+1 // zapis pocit. nalez. diag
  </script>
  <CSP:ELSE>
    <script LANGUAGE="CACHE" RUNAT="SERVER">
      set a=a+1 // zapis pocitadla nenalez. diag
    </script>
  </CSP:IF>
</CSP:WHILE>
</table>
</td></table><br>
<font color="red" size="-1"> #(b)# - po&#269;et nalezených záznam&#367; se zadanou diagnózou </font><br>
<font color="green" size="-1"> #(a)# - po&#269;et nalezených záznam&#367; s rozdílnou diagnózou </font><br>
<CSP:COMMENT> BIND - tlačítko zpět (výsledek nalezených diagnóz) </CSP:COMMENT>
<div align="left">
  <form name="form1" method="post" action="#(namef)#">
    <input type="hidden" name="zvoldiag" value="">
    #(.InsertHiddenField("#(namef)#", "jmeno", (%request.Data("jmeno",1))))#
    #(.InsertHiddenField("#(namef)#", "heslo", (%request.Data("heslo",1))))#
    #(.InsertHiddenField("#(namef)#", "obsah", (%request.Data("obsah",1))))#
    &nbsp;<input type="submit" name="zpet" value="Zp&#283t na zadání diagnózy">
  </form></div>
  <CSP:COMMENT> nabídka schovat DS pokud je zobrazen </CSP:COMMENT>
  <CSP:IF CONDITION=$G(%request.Data("showds",1))["Zo">
    <form name="form7" method="post" action="#(namef)#">
      <input type="hidden" name="zvoldiag" value="#(%request.Data("zvoldiag",1))#">
      #(.InsertHiddenField("#(namef)#", "heslo", (%request.Data("heslo",1))))#
      #(.InsertHiddenField("#(namef)#", "jmeno", (%request.Data("jmeno",1))))#
      #(.InsertHiddenField("#(namef)#", "obsah", (%request.Data("obsah",1))))#
      <input type="submit" name="showds" value="Zav&#345;ít vzorové DS">
    </form><center>
  <CSP:COMMENT> nabídka zobraz DS pokud není zobrazeno </CSP:COMMENT>
  <CSP:ELSE>
    <form name="form6" method="post" action="#(namef)#">
      <input type="hidden" name="zvoldiag" value="#(%request.Data("zvoldiag",1))#">
      #(.InsertHiddenField("#(namef)#", "heslo", (%request.Data("heslo",1))))#
      #(.InsertHiddenField("#(namef)#", "jmeno", (%request.Data("jmeno",1))))#
      #(.InsertHiddenField("#(namef)#", "obsah", (%request.Data("obsah",1))))#
      <input type="submit" name="showds" value="Zobrazit data ve vzorovém DS">
    </form><center>
  </CSP:IF>
  <CSP:COMMENT> Zobrazit-Generovat vybranou DIAG v DS </CSP:COMMENT>
  <CSP:IF CONDITION=$G(%request.Data("showds",1))["Zo">
    <br><u>Nalezená data pro zvolenou diagnózu:#(%request.Data("zvoldiag",1))#, zobrazená ve formátu DS3:</u><br>
    <table class="tb2" width="65%">
      <tr><td align="center"><b>V<br>Z<br>O<br>R<br>O<br>V<br>Ý<br>&nbsp;<br>V<br>Y<br>G<br>E<br>N<br>E<br>R<br>O<br>V<br>

```

```

>a<br>n<br>ý<br>&nbsp;<br>k<br>ó<br>d<br></b></td>
  &nbsp;<br>&lt;?xml version='1.0' encoding='Windows-1250' standalone='no' ?&gt;
  <br>&nbsp;<br>&lt;DOCTYPE dasta SYSTEM "ds030701.dtd"&gt;
  <br>&nbsp;<br>&lt;dasta
id_soubor="VUT_UBMI_00_00000_#($ZDATE($Horolog,3))#T#($ZTIME($PIECE($HOROLOG,"",2),1))#" verze_ds="03.01.01"
ur="T" potvrzeni="N"&gt;
  <br>&nbsp;<br>&nbsp;<br>&lt;zdroj_is kod_firmy="VUT_UBMI" kod_prog="CSP" verze_prog="xx" liccis_prog="xx" /&gt;
  <br>&nbsp;<br>&lt;pm ico="12345678" &gt;
  <br>&nbsp;<br>&nbsp;<br>&lt;as typ="E" &gt; <!-- blok adresa-spojeni cil -->
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;obsah&gt;adresat@vutbr.cz&lt;/obsah&gt;
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;as&gt;
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;a typ="X" &gt; <!-- blok adresa X-jina -->
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;jmeno&gt;Jmeno Prijmeni (firma) &lt;/jmeno&gt;
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;psc&gt;630 00&lt;/psc&gt; <!-- max 9 znaku -->
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;mesto&gt;Brno&lt;/mesto&gt;
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;stat&gt;CZ&lt;/stat&gt;
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;a&gt;
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;pm&gt;
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;is icz="12345678"&gt; <!-- odesilatel -->
  <!-- Slouzeni diagnoz tentokrat bez group by, priradit pacienty, vybrat dle diagnozy -->
  <SCRIPT LANGUAGE="SQL" NAME="diagnozDS">
  SELECT SUBSTRING(DGDIAGNOZA,1,3) AS DGZ, DGSKupinaDiagnoz, DGPoprve, DGICD10, ICICD10, ICPopis,
Aenderungsdatum, geaendert_durch,
  SUBSTRING(Aenderungsdatum,1,4) AS dsdrok, SUBSTRING(Aenderungsdatum,5,2) AS dsdmes,
SUBSTRING(Aenderungsdatum,7,2) AS dsdden
  FROM SQLUser.VDiagnoza
  LEFT JOIN SQLUser.VICD10KS ON dgicd10=ICICD10
  LEFT JOIN SQLUser.Diagnosegruppe ON DGSKupinaDiagnoz=Diagnosegruppe ORDER by ICICD10
</SCRIPT>
  <CSP:WHILE CONDITION=diagnozDS.Next()>
  <SCRIPT LANGUAGE="SQL" NAME="pacientDS">
  SELECT SUBSTRING(PAUmrtiDatum,1,4) AS PAUmrtiRok,SUBSTRING(PAUmrtiDatum,5,2)
AS PAUmrtiMesic,SUBSTRING(PAUmrtiDatum,7,2) AS PAUmrtiDen,
  SUBSTRING(PANarozeniDatum,1,4) AS PANarRok,SUBSTRING(PANarozeniDatum,5,2)
AS PANarMesic,SUBSTRING(PANarozeniDatum,7,2) AS PANarDen,
  PAPACIENT, PAPacientCislo, PACisloRodne, PAJmeno, PAPrijmeni, PATitulAkad,
  PANarozeniDatum, PAUmrtiDatum,PAPohlavi, PAJmenoRodne, PAUlice, PAMesto, PAPsc
  From SQLUser.Vpacient where PAPACIENT=#(diagnozDS.Data("DGZ"))#
  </SCRIPT>
  <CSP:WHILE CONDITION=pacientDS.Next()>
  <CSP:IF CONDITION=(diagnozDS.Data("DGICD10"))=#G(%request.Data("zvoldiag",1))>
  <script LANGUAGE="CACHE" RUNAT="SERVER">
  // diag opakovani pri stejnem pacientovi... neopakovat blok
  if pacopak=(pacientDS.Data("PACisloRodne")) {
  set opak="ano"
  }
  else { set opak="ne"
  if pacopak="0" { set zakoncitip="ano" }
  }
  </script><CSP:IF CONDITION=!opak="ano">
  <br>&nbsp;<br>&nbsp;<br>&lt;dgz ind_oprav_sd="N"&gt;
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;diag poradi="1" &gt; #(diagnozDS.Data("DGICD10"))#&lt;/diag&gt;
  <!-- kod diagnozy mkn-10 -->
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;dat_du format="D" &gt; #(diagnozDS.Data("dsdrok"))#-#(diagnozDS.Data("dsdmes"))#-
#(diagnozDS.Data("dsdden"))#&lt;/dat_du&gt; <!-- // datum diagnozy -->
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;spec_dg&gt; #(diagnozDS.Data("ICPopis"))#&lt;/spec_dg&gt;
  <!-- popis diagnozy ! -->
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;autor&gt;
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;jmeno&gt; #(diagnozDS.Data("geaendert_durch"))#&lt;/jmeno&gt;
  <!-- s27 atd -->
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;autor&gt;
  <br>&nbsp;<br>&nbsp;<br>&nbsp;<br>&lt;dgz&gt;
  <CSP:ELSE>
  <CSP:IF CONDITION='zakoncitip="ano"'>
  <br>&nbsp;<br>&nbsp;<br>&lt;/ip&gt;
  </CSP:IF>
  <br>&nbsp;<br>&nbsp;<br>&lt;ip id_pac="#(pacientDS.Data("PAPACIENT"))#000#(pacientDS.Data("PAPacientCislo"))#" &gt; <!-- data pacienta
v IS !!!PAPACIENT PAPacientCislo -->
  <br>&nbsp;<br>&nbsp;<br>&lt;rodcis&gt; #(pacientDS.Data("PACisloRodne"))#&lt;/rodcis&gt;
  <br>&nbsp;<br>&nbsp;<br>&lt;jmeno&gt; #(pacientDS.Data("PAJmeno"))#&lt;/jmeno&gt;
  <br>&nbsp;<br>&nbsp;<br>&lt;prijmeni&gt; #(pacientDS.Data("PAPrijmeni"))#&lt;/prijmeni&gt;
  <CSP:IF CONDITION=(pacientDS.Data("PATitulAkad"))="">
  <br>&nbsp;<br>&nbsp;<br>&lt;titul_pred&gt; #(pacientDS.Data("PATitulAkad"))#&lt;/titul_pred&gt; <!-- PATitulAkad
je nebo null -->
  </CSP:IF>
  <CSP:IF CONDITION=(pacientDS.Data("PANarozeniDatum"))="">

```



```

SELECT COUNT(DGICD10) AS pocet, SUBSTRING(DGDIAGNOZA,1,3) AS DGZ,SUBSTRING(DGICD10,2,5) AS cCD10,
DGSkupinaDiagnoz, DGICD10, ICICD10, ICPopis, Aenderungsdatum, SUBSTRING(Aenderungsdatum,1,4) AS diagROK
FROM SQLUser.VDiagnoza
LEFT JOIN SQLUser.VICD10KS ON dgid10=ICICD10
LEFT JOIN SQLUser.Diagnosegruppe ON DGSkupinaDiagnoz=Diagnosegruppe where (DGICD10 LIKE 'A%' OR DGICD10 LIKE
'B%') GROUP BY DGICD10,SUBSTRING(Aenderungsdatum,1,4) order by pocet DESC
</SCRIPT>
<CSP:COMMENT> data pro Tgraf - onkologicke </CSP:COMMENT>
<CSP:ELSEIF CONDITION='$G(%request.Data("Tgrafvyber",1))="3"'>
<SCRIPT LANGUAGE="SQL" NAME="diagnozTgraf">
SELECT COUNT(DGICD10) AS pocet, SUBSTRING(DGDIAGNOZA,1,3) AS DGZ,SUBSTRING(DGICD10,2,5) AS cCD10,
DGSkupinaDiagnoz, DGICD10, ICICD10, ICPopis, Aenderungsdatum, SUBSTRING(Aenderungsdatum,1,4) AS diagROK
FROM SQLUser.VDiagnoza
LEFT JOIN SQLUser.VICD10KS ON dgid10=ICICD10
LEFT JOIN SQLUser.Diagnosegruppe ON DGSkupinaDiagnoz=Diagnosegruppe where (DGICD10 LIKE 'C%' OR DGICD10 LIKE
'D%') GROUP BY DGICD10,SUBSTRING(Aenderungsdatum,1,4) order by pocet DESC
</SCRIPT>
</CSP:IF>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set i=0
</script>
<table><tr><td style="background-color: white;" align="center">
<div id="graf2">
<dl>
<dt><h2>Grafy výskytu zvolených (skupiny) diagnóz:</h2><font size="-2" color="#195481">Poznámka: Nepodléhá výb&#283;ru roku,
&#345;azeno dle po&#269;tu (Nr.) nalezených diagnóz.<br></font></dt>
<CSP:WHILE CONDITION=diagnozTgraf.Next()>
<script LANGUAGE="CACHE" RUNAT="SERVER">
// ulozeni hodnot pro rok z vyberu (2008-default) do vektoru ( pro google chart )
if (%request.Data("Tgrafvyber",1))="1" { // 1-vsechny diag
if (%request.Data("TgrafROK",1))="" {set (%request.Data("TgrafROK",1))="2008" write "<font color=green size=-2>Nezadali jste rok
pro graf-kolá&#269;., automaticky nastavuji rok 2008 </font>" }
if (diagnozTgraf.Data("diagROK"))=(%request.Data("TgrafROK",1)) {
if (diagnozTgraf.Data("pocet"))="1" // u vyberu vseh diagnoz omezi na vyskyt-pocet 2a vice
{ set MyArrayPocet(i)=(diagnozTgraf.Data("pocet"))
set MyArrayDiagnoza(i)=(diagnozTgraf.Data("cCD10"))
set i=i+1
} } }
else {
if (%request.Data("TgrafROK",1))="" {set (%request.Data("TgrafROK",1))="2008" write "<font color=green size=-2>Nezadali jste rok
pro graf-kolá&#269;., automaticky nastavuji rok 2008 </font>" }
if (diagnozTgraf.Data("diagROK"))=(%request.Data("TgrafROK",1)) {
set MyArrayPocet(i)=(diagnozTgraf.Data("pocet"))
set MyArrayDiagnoza(i)=(diagnozTgraf.Data("cCD10"))
set i=i+1
} }
} }
</script>
<CSP:IF CONDITION=(diagnozTgraf.Data("diagROK"))="2006">
<dd class="p#(diagnozTgraf.Data("pocet"))#"><div class="popisek" style="background:
#fce">Nr.:(diagnozTgraf.Data("pocet"))#<br>#(diagnozTgraf.Data("cCD10"))#<br>#(diagnozTgraf.Data("diagROK"))#</div></dd>
<CSP:ELSEIF CONDITION=(diagnozTgraf.Data("diagROK"))="2007">
<dd class="p#(diagnozTgraf.Data("pocet"))#"><div class="popisek" style="background:
#ece">Nr.:(diagnozTgraf.Data("pocet"))#<br>#(diagnozTgraf.Data("cCD10"))#<br>#(diagnozTgraf.Data("diagROK"))#</div></dd>
<CSP:ELSEIF CONDITION=(diagnozTgraf.Data("diagROK"))="2008">
<dd class="p#(diagnozTgraf.Data("pocet"))#"><div class="popisek" style="background:
#dce">Nr.:(diagnozTgraf.Data("pocet"))#<br>#(diagnozTgraf.Data("cCD10"))#<br>#(diagnozTgraf.Data("diagROK"))#</div></dd>
<CSP:ELSEIF CONDITION=(diagnozTgraf.Data("diagROK"))="2009">
<dd class="p#(diagnozTgraf.Data("pocet"))#"><div class="popisek" style="background:
#cce">Nr.:(diagnozTgraf.Data("pocet"))#<br>#(diagnozTgraf.Data("cCD10"))#<br>#(diagnozTgraf.Data("diagROK"))#</div></dd>
<CSP:ELSEIF CONDITION=(diagnozTgraf.Data("diagROK"))="2010">
<dd class="p#(diagnozTgraf.Data("pocet"))#"><div class="popisek" style="background:
#bce">Nr.:(diagnozTgraf.Data("pocet"))#<br>#(diagnozTgraf.Data("cCD10"))#<br>#(diagnozTgraf.Data("diagROK"))#</div></dd>
<CSP:ELSE>
<dd class="p#(diagnozTgraf.Data("pocet"))#"><div class="popisek" style="background:
#ace">Nr.:(diagnozTgraf.Data("pocet"))#<br>#(diagnozTgraf.Data("cCD10"))#<br>#(diagnozTgraf.Data("diagROK"))#</div></dd>
</CSP:IF>
</CSP:WHILE>
<div class="cl"></div>
</dl>
</div>
<br><h2>Graf výskytu zvolených diagnóz pro rok #( %request.Data("TgrafROK",1) )#:</h2><br><br>
<script LANGUAGE="CACHE" RUNAT="SERVER">
//generovani dat pro graf (= nerovnost !!!) posledni udaj nema , a |
set i=i-1
write "<img src= http://chart.apis.google.com/chart?cht=p&chs=650x450&chco=0000FF&chl="
for g=0:1:i {

```

```

Write MyArrayDiagnoza(g)
if g'=i write "|"
}
write "&chd=t:"
for g=0:1:i {
Write MyArrayPocet(g)
if g'=i write ","
}
write ">"
</script>
<CSP:COMMENT> Zavrit TGRAfY </CSP:COMMENT>
<div align="left">
<form name="form1" method="post" action="#(namef)#">
<input type="hidden" name="zvolddiag" value="">
<input type="hidden" name="Tgrafy" value="">
#(..InsertHiddenField("#(namef)#","jmeno",(%request.Data("jmeno",1))))#
#(..InsertHiddenField("#(namef)#","heslo",(%request.Data("heslo",1))))#
#(..InsertHiddenField("#(namef)#","obsah",(%request.Data("obsah",1))))#
&nbsp;<input type="submit" name="zpet" value="Skřýt Grafy">
</form></div></td></tr></table>
</CSP:IF>
<CSP:COMMENT> obsah dle intervalu obsah=dfi </CSP:COMMENT>
<CSP:ELSEIF CONDITION=$G(%request.Data("obsah",1))["dfi"]>
<CSP:COMMENT> umozni zvolit-vybrat rozsah hledani-pokud neni jeste zvolen </CSP:COMMENT>
<CSP:IF CONDITION=$G(%request.Data("hledat",1))="">
<br><br>
<CSP:COMMENT> vyberovej select.. vygeneruje roky, jenz DB obsahuje </CSP:COMMENT>
<SCRIPT LANGUAGE="SQL" NAME="kodyd">
SELECT SUBSTRING(Aenderungsdatum,1,4) AS drok
from SQLUser.Diagnosegruppe Group by SUBSTRING(Aenderungsdatum,1,4) order by SUBSTRING(Aenderungsdatum,1,4) ASC
</SCRIPT>
<CSP:COMMENT> vyberovy select (i pro naplneni DiagText...) .. spoji diag a vicd10, seskupi diag group by, vytvoreni cyklu pro
option</CSP:COMMENT>
<SCRIPT LANGUAGE="SQL" NAME="kodyd2">
SELECT DGICD10, ICICD10, ICPopis, SUBSTRING(DGICD10,2,5) AS DGICD100
FROM SQLUser.VDiagnoza
LEFT JOIN SQLUser.VICD10KS ON dgicd10=ICICD10 GROUP by ICICD10 Order by dgicd100 ASC
</SCRIPT>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set DiagTextPole=0 // deklarace pole diagvektoru pro select
set DiagTextI=0 // deklace naplneni
write "<script language=javascript>"
write "if (location.href.indexOf('') != -1) { "
write "window.onload = function () { "
write "data = ["
</script>
<CSP:WHILE CONDITION=kodyd2.Next()><script LANGUAGE="CACHE" RUNAT="SERVER">
set DiagTextPole(DiagTextI,(DiagTextI))=(kodyd2.Data("DGICD100"))
set DiagTextPole(DiagTextI,(DiagTextI+1))=(kodyd2.Data("ICPopis"))
set DiagTextI=DiagTextI+1
write """,kodyd2.Data("DGICD100"),"" </script></CSP:WHILE>' ].sort();
AutoComplete_Create('diagnozy', data);
}
}
if (top.location.href != location.href) {
top.location.href = location.href;
}
</script>

<script LANGUAGE="CACHE" RUNAT="SERVER">
set irok=0
</script>
<form name="form" method="post" action="#(namef)#">
#(..InsertHiddenField("#(namef)#","jmeno",(%request.Data("jmeno",1))))#
#(..InsertHiddenField("#(namef)#","heslo",(%request.Data("heslo",1))))#
#(..InsertHiddenField("#(namef)#","obsah",(%request.Data("obsah",1))))#
<br>
<fieldset style="width: 700px;">
<legend style="color: green; font-size: 80%;">Zde máte mo&#382;nost zadat konkrétní interval (nepovinný údaj)</legend>
<br>Interval od:
<select name="rokod">
<option style="color: black;" value="">rok</option>
<CSP:WHILE CONDITION=kodyd.Next()>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set rokvyb(irok)=kodyd.Data("drok")
set irok=irok+1

```



```

set rokod=$G(%request.Data("nejmrok",1))
else { set rokod=$G(%request.Data("rokod",1)) }
if $G(%request.Data("rokdo",1))="" {
write "<font color=green size=-2>Nezadali jste: Rok DO >> vypl&#328;uji nejvy&#353;&#353;i rok z DB:
",($G(%request.Data("nejvrok",1))),"</font><br>"
set rokdo=$G(%request.Data("nejvrok",1))
else { set rokdo=$G(%request.Data("rokdo",1)) }
if rokod>rokdo {
write "<font color=red>Zadali jste &#353;patn&#283; interval: ROK OD-DO (OD je VET&#352;i ne&#382; DO)</font><br>"
set pokračovat=0
}
else {
if $G(%request.Data("mesod",1))="" {
write "<font color=green size=-2>Nezadali jste: M&#283;sic OD >> vypl&#328;uji 01. m&#277;sic </font><br>"
set mesod="01" }
else { set mesod=$G(%request.Data("mesod",1)) }
if $G(%request.Data("mesdo",1))="" {
if rokdo=($Extract($ZDate($Horolog,4),7,10)) {set mesdo=($Extract($ZDate($Horolog,4),4,5)) } // DO AKTUALNIHO MESICE (
mozna bude lepsi mesic - 1..)
else { set mesdo="12" }
write "<font color=green size=-2>Nezadali jste: M&#283;sic DO >> vypl&#328;uji ",mesdo,". m&#277;sic </font><br>"
}
else { if (rokdo=($Extract($ZDate($Horolog,4),7,10)) & ($G(%request.Data("mesdo",1))>($Extract($ZDate($Horolog,4),4,5))) {
set mesdo=($Extract($ZDate($Horolog,4),4,5))
else { set mesdo=$G(%request.Data("mesdo",1)) }
}
}
set pokračovat=1
if (rokod=rokdo) & (mesod>mesdo) {
write "<font color=red>Zadali jste &#353;patnou &#269;asovou-m&#277;si&#269;ni posloupnost (DO je ni&#382;&#353;i ne&#382;
OD) pro rok: ",rokod,".</font><br>"
set pokračovat=0
}}
</script>
<CSP:COMMENT> pokud vyhovuji podminky(predchozi kontrola) povoli pokračovani </CSP:COMMENT>
<CSP:IF CONDITION="pokracovat="1">
<center><h2>Pacienti nalezeni dle zadaného intervalu:</h2></center>
<table class="tb1" width="85%"><td><table class="tb2" width="100%" align="center">
<tr style="color: black; background: silver;">
<td align="center"><b>DG kód</b></td>
<td><b> Popis diagnózy</b></td>
<td align="center"><b>DG Poprvé</b></td>
<td><b> P&#345;ijmení</b></td>
<td><b> Jméno</b></td>
<td align="center"><b>Rodné &#269;islo</b></td>
<td align="center"><b>Datum a &#269;as </b></td>
<td align="center"><b>Léka&#345;</b></td>
</tr>
</table>
</table>
<script LANGUAGE="CACHE" RUNAT="SERVER">
//deklarace pocitadla
set b=0
set a=0
</script>
<!-- Sloucení diagnóz tentokrát bez group by, sloučení diag. s pacienty, vyber pacientu podle diag -->
<SCRIPT LANGUAGE="SQL" NAME="diagnoz">
SELECT SUBSTRING(DGDIAGNOZA,1,3) AS DGZ, DGSkupinaDiagnoz, DGPoprve, DGICD10,SUBSTRING(DGICD10,2,5)
AS CD10, ICICD10, ICPopis, Aenderungsdatum, geaendert_durch,
SUBSTRING(Aenderungsdatum,1,4) AS drok, SUBSTRING(Aenderungsdatum,5,2) AS dmes, SUBSTRING(Aenderungsdatum,7,2)
AS dden, SUBSTRING(Aenderungsdatum,9,2) AS dhod, SUBSTRING(Aenderungsdatum,11,2) AS dmin,
SUBSTRING(PAUmrtiDatum,1,4) AS PAUmrtiRok,SUBSTRING(PAUmrtiDatum,5,2)
AS PAUmrtiMesic,SUBSTRING(PAUmrtiDatum,7,2) AS PAUmrtiDen,
SUBSTRING(PANarozeniDatum,1,4) AS PANarRok,SUBSTRING(PANarozeniDatum,5,2)
AS PANarMesic,SUBSTRING(PANarozeniDatum,7,2) AS PANarDen,
PAPACIENT, PAPatientCislo, PACisloRodne, PAJmeno, PAPrijmeni, PATitulAkad,
PANarozeniDatum, PAUmrtiDatum,PAPohlavi, PAJmenoRodne, PAUlice, PAMesto, PAPsc
FROM SQLUser.VDiagnoza
LEFT JOIN SQLUser.VICD10KS ON dgicd10=ICICD10
LEFT JOIN SQLUser.Vpatient ON PAPACIENT=SUBSTRING(DGDIAGNOZA,1,3)
LEFT JOIN SQLUser.Diagnosegruppe ON DGSkupinaDiagnoz=Diagnosegruppe where (SUBSTRING(Aenderungsdatum,1,6)>={fn C
ONCAT("#(rokod)#,#(mesod)#"}) AND SUBSTRING(Aenderungsdatum,1,6)<={fn CONCAT("#(rokdo)#,#(mesdo)#")})
order by SUBSTRING(Aenderungsdatum,1,12) ASC
</SCRIPT>
<CSP:WHILE CONDITION=diagnoz.Next()>
<script LANGUAGE="CACHE" RUNAT="SERVER">
// !!!! zajisti vypis vybranych-NASTAVENYCH diagnóz pres promenu vypsát !!!!
if $G(%request.Data("hledatinf",1))="" {
if (diagnoz.Data("CD10"))["A" { set vypsát="1" }

```



```

elseif (diagnoz.Data("CD10"))["B" { set vypsats="1" }
else { set vypsats="0" }
}
elseif $G(%request.Data("hledatonk",1))="" {
if (diagnoz.Data("CD10"))["C" { set vypsats="1" }
elseif (diagnoz.Data("CD10"))["D" { set vypsats="1" }
else { set vypsats="0" }
}
elseif $G(%request.Data("hledatvse",1))="" {
set vypsats="1" }
else { set vypsats="0" }
// pokud vyberem diagnozu-pocatek..pripadne da prednost hledatvybd2.. take osetreni value=pocatek kodu...vyhleda vse
if $G(%request.Data("hledatvybd2",1))="" { set (%request.Data("hledatvybd",1))=(%request.Data("hledatvybd2",1)) } // pokud je
vyber hledatvybd2 vybran uprednostni ho pred hledatvybd
if $G(%request.Data("hledatvybd",1))["po" { set %request.Data("hledatvybd",1)="" }
if $G(%request.Data("hledatvyb",1))="" {
set hledatvybdiag=$G(%request.Data("hledatvybd",1))
if (diagnoz.Data("CD10"))[hledatvybdiag { set vypsats="1" }
else { set vypsats="0" }
}
}
</script>
<CSP:IF CONDITION='vypsats="1"'>
<script LANGUAGE="CACHE" RUNAT="SERVER">
// Uprava poprve nul nezadana, 0 nezjistena, 1 Ano, 2 Ne
if (diagnoz.Data("DGPoprve"))="0" { set poprve="Nezji&#353;t&#283no" }
elseif (diagnoz.Data("DGPoprve"))="1" { set poprve="Ano" }
elseif (diagnoz.Data("DGPoprve"))="2" { set poprve="Ne" }
else { set poprve="Nezadano" }
</script>
<tr>
<td align="center" width="10%">#(diagnoz.Data("DGICD10"))#</td>
<td width="25%">&nbsp;#(diagnoz.Data("ICPopis"))# &nbsp;#</td>
<td align="center" width="10%">#(poprve)#</td>
<td width="13%">&nbsp;#(diagnoz.Data("PAPrijmeni"))#</td>
<td width="12%">&nbsp;#(diagnoz.Data("PAJmeno"))#</td>
<td align="center" width="10%">#(diagnoz.Data("PACisloRodne"))#</td>
<td align="center" width="15%">#(diagnoz.Data("dden"))#.#(diagnoz.Data("dmes"))#.#(diagnoz.Data("drok"))# v
#(diagnoz.Data("dhod"))#.#(diagnoz.Data("dmin"))#</td>
<td align="center" width="5%">#(diagnoz.Data("geaendert_durch"))#</td>
</tr>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set b=b+1 // zapis pocit. nalez. diag
</script>
<CSP:ELSE>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set a=a+1 // zapis pocitadla nenalez.-nevyhovujich zaznamu
</script>
</CSP:IF>
</CSP:WHILE>
</table></td></table><br>
<font color="red" size="-1"> #(b)# - po&#269;et nalezených záznam&#367; v zadaném rozsahu </font><br>
<font color="green" size="-1"> #(a+b)# - po&#269;et vsech nalezených záznam&#367; v databázi </font><br>
<br><br>
<CSP:IF CONDITION='b>0'>
<table class="tbl"><tr><td style="border-width: 1px; background-
color:silver;"><a href="ds.csp?ove=#(((($ZDATE($HOROLOG,8))*1.35)*$EXTRACT($ZTIME($PIECE($HOROLOG,".",2),1,9),1,2))#&
rokod=#(rokod)#&mesod=#(mesod)#&rokdo=#(rokdo)#&mesdo=#(mesdo)#&hledatinf=#($G(%request.Data("hledatinf",1)))#&hledatonk
=#($G(%request.Data("hledatonk",1)))#&hledatvse=#($G(%request.Data("hledatvse",1)))#&hledatvyb=#($G(%request.Data("hledatvyb",1
)))#&hledatvybd=#($G(%request.Data("hledatvybd",1)))# target="_blank" class="link">&nbsp;#&nbsp; Zobrazit nalezené výsledky v DS 3
&nbsp;#</a></td></tr></table><br>
</CSP:IF>
</CSP:IF>
<CSP:COMMENT> BIND - tlacitko zpět </CSP:COMMENT>
<div align="left">
<form name="form1" method="post" action="#(namef)#">
<input type="hidden" name="hledat" value="">
#(..InsertHiddenField("#(namef)#", "jmeno", (%request.Data("jmeno",1)))#
#(..InsertHiddenField("#(namef)#", "heslo", (%request.Data("heslo",1)))#
#(..InsertHiddenField("#(namef)#", "obsah", (%request.Data("obsah",1)))#
&nbsp;#<input type="submit" name="zpet" value="Zp&#283t na zadání intervalu">
</form>
</div>
</CSP:IF>
<CSP:COMMENT> obsah dle jmena .. obsah=dfj </CSP:COMMENT>
<CSP:ELSEIF CONDITION=$G(%request.Data("obsah",1))["dfj"]>
<CSP:COMMENT> umozni zvolit jmeno .. pokud není vybráno </CSP:COMMENT>

```

```

<CSP:IF CONDITION=$G(%request.Data("hledat",1))="">
<br><br><i>Zadejte p&#345;jmeni pacienta - p&#345;ipadn&#283; jeho &#269;ást.( Dodr&#382;ujte malá-VELKÁ písmena a
diakritiku )</i><br>
<CSP:COMMENT> vyberovy select.. vygeneruje prijmeni jenz DB obsahuje - pro autocomplete </CSP:COMMENT>
<SCRIPT LANGUAGE="SQL" NAME="kodyd">
SELECT PAPrijmeni, PACisloRodne
from SQLUser.Vpacient Group By PACisloRodne order by PAPrijmeni ASC
</SCRIPT>
<script LANGUAGE="CACHE" RUNAT="SERVER">
write "<script language=javascript>"
write "if (location.href.indexOf('') != -1) { "
write "window.onload = function () { "
write "data = ["
</script>
<CSP:WHILE CONDITION=kodyd.Next()> <script LANGUAGE="CACHE" RUNAT="SERVER">
write ""',kodyd.Data("PAPrijmeni"),''," </script></CSP:WHILE>''],sort();
AutoComplete_Create('prijmeni', data);
}
}
if (top.location.href != location.href) {
top.location.href = location.href;
}
</script>
<form name="form" method="post" action="#"(namef)#">
#(..InsertHiddenField("#(namef)#","jmeno",(%request.Data("jmeno",1))))#
#(..InsertHiddenField("#(namef)#","heslo",(%request.Data("heslo",1))))#
#(..InsertHiddenField("#(namef)#","obsah",(%request.Data("obsah",1))))#
<br>
<table border="0" align="center"><tr><td>
<div align="left">
<input type="text" style="color: red;" id="prijmeni" name="hledatjm" value=" zadejte
p&#345;jmeni:" onfocus="this.value="">
<input type="submit" name="hledat" value="Vyhledat v databázi">
</div>
</td></tr>
</table>
</form>
<CSP:COMMENT> Jestlize je jiz vybran pacient - zobrazi jeho detaily..diagnozy atd. </CSP:COMMENT>
<CSP:ELSEIF CONDITION=$G(%request.Data("hledatpac",1))="1">
<table class="tb1" width="85%"><tr>
<td align="center"><b>DG kód</b></td>
<td align="center"><b>Popis diagnózy</b></td>
<td align="center"><b>DG Poprvé</b></td>
<td align="center"><b>P&#345;jmeni</b></td>
<td align="center"><b>Jméno</b></td>
<td align="center"><b>Rodné &#269;islo</b></td>
<td align="center"><b>Datum a &#269;as </b></td>
<td align="center"><b>Léka&#345;</b></td>
</tr>
<script LANGUAGE="CACHE" RUNAT="SERVER">
//deklarace nenalezene diagnozy.. nize
set nenalezenadg=0
</script>
<!-- Sloucení diagnóz tentokrát bez group by, sloučení diag. s pacienty, vyber pacientu podle diag -->
<SCRIPT LANGUAGE="SQL" NAME="diagnoz">
SELECT SUBSTRING(DGDIAGNOZA,1,3) AS DGZ, DGSkupinaDiagnoz, DGPoprve, DGICD10, ICICD10, ICPopis,
Aenderungsdatum, geaendert_durch,
SUBSTRING(Aenderungsdatum,1,4) AS drok, SUBSTRING(Aenderungsdatum,5,2) AS dmes, SUBSTRING(Aenderungsdatum,7,2)
AS dden,
SUBSTRING(Aenderungsdatum,9,2) AS dhod, SUBSTRING(Aenderungsdatum,11,2) AS dmin, PACisloRodne,PAJmeno,PAPrijmeni
FROM SQLUser.VDiagnoza
LEFT JOIN SQLUser.VICD10KS ON dgicd10=ICICD10
LEFT JOIN SQLUser.Vpacient ON PAPACIENT=SUBSTRING(DGDIAGNOZA,1,3)
LEFT JOIN SQLUser.Diagnosegruppe ON DGSkupinaDiagnoz=Diagnosegruppe ORDER by ICICD10
</SCRIPT>
<CSP:WHILE CONDITION=diagnoz.Next()>
<CSP:COMMENT> Vybere info. pacienta dle RC - preposlane z predchoziho vyberu </CSP:COMMENT>
<CSP:IF CONDITION=(diagnoz.Data("PACisloRodne"))=$G(%request.Data("hledatpacrc",1))>
<script LANGUAGE="CACHE" RUNAT="SERVER">
// Uprava poprve nul nezadana, 0 nezjistena, 1 Ano, 2 Ne
if (diagnoz.Data("DGPoprve"))="0" { set poprve="Nezji&#353;t&#283;no" }
elseif (diagnoz.Data("DGPoprve"))="1" { set poprve="Ano" }
elseif (diagnoz.Data("DGPoprve"))="2" { set poprve="Ne" }
else { set poprve="Nezadáno" }

```

```

// Kontrola prazdne diagnozy
if (diagnoz.Data("DGICD10"))="" { set nenalezenadg=nenalezenadg+1 }
</script>
<tr>
<td align="center" width="10%">#{diagnoz.Data("DGICD10")}#</td>
<td width="25%">&nbsp;#{diagnoz.Data("ICPopis")}# &nbsp;#</td>
<td align="center" width="10%">#{poprve}#</td>
<td width="13%">&nbsp;#{diagnoz.Data("PAPrijmeni")}#</td>
<td width="12%">&nbsp;#{diagnoz.Data("PAJmeno")}#</td>
<td align="center" width="10%">#{diagnoz.Data("PACisloRodne")}#</td>
<td align="center" width="15%">#{diagnoz.Data("dden")}#.#(diagnoz.Data("dmes"))#.#(diagnoz.Data("drok"))# v
#{diagnoz.Data("dhod")}#:#(diagnoz.Data("dmin"))#</td>
<td align="center" width="5%">#{diagnoz.Data("geaendert_durch")}#</td>
</tr>
</CSP:IF>
</CSP:WHILE>
</table></td>
</table><br>
<script LANGUAGE="CACHE" RUNAT="SERVER">
// pri nenalezene diag. u pacienta vypise
if nenalezenadg=0 { write "<font color=red size=-1> U pacienta nebyla nalezena &#382;ádná diagnóza </font>" }
//prehozeni promeny pro zpetny seznam pac. podle hledani
set hledatpacc=$G(%request.Data("hledatpacc",1))
</script>

<CSP:COMMENT> BIND - tlacitko zpet </CSP:COMMENT>
<div align="left">
<form name="form1" method="post" action="#(namef)#">
<input type="hidden" name="hledat" value="1">
<input type="hidden" name="hledatjm" value="#(hledatpacc)#">
#{..InsertHiddenField("#(namef)#","jmeno",(%request.Data("jmeno",1)))#}
#{..InsertHiddenField("#(namef)#","heslo",(%request.Data("heslo",1)))#}
#{..InsertHiddenField("#(namef)#","obsah",(%request.Data("obsah",1)))#}
&nbsp;#<input type="submit" name="zpet" value="Zp&#283t na seznam nalezenych">
</form>
</div>
<CSP:COMMENT> pokud je vybrano prijmeni .. zobrazi seznam </CSP:COMMENT>
<CSP:ELSE>
<script LANGUAGE="CACHE" RUNAT="SERVER">
// pokud neni zadane jmeno neumozni pokracovat !!!!
if $G(%request.Data("hledatjm",1))="" {
write "<font color=green>Zadejte prosím P&#345;ijmeni nebo jeho po&#269;átek</font>"
set pokracovat="0" }
elseif $G(%request.Data("hledatjm",1))[" zadejte p" {
write "<font color=green>Zadejte prosím P&#345;ijmeni nebo jeho po&#269;átek</font>"
set pokracovat="0" }
else { set hledatjm=$G(%request.Data("hledatjm",1))
set pokracovat="1" }
</script>
<CSP:IF CONDITION=pokracovat="1">
<center><h2>Pacienti nalezeni dle zadaného p&#345;ijmeni:</h2></center>
<table class="tb1" width="85%"><td><table class="tb2" width="100%" align="center">
<tr style="color: black; background: silver;">
<td><b> P&#345;ijmeni</b></td>
<td><b> Jméno</b></td>
<td><b> M&#277;sto</b></td>
<td align="center"><b>Rodné &#269;islo</b></td>
</tr>
<script LANGUAGE="CACHE" RUNAT="SERVER">
//deklarace pocitadla nalezenych pacientu dle prijmeni
set b=0
set a=0
</script>
<SCRIPT LANGUAGE="SQL" NAME="pacient">
SELECT * From SQLUser.Vpacient Group by PACisloRodne Order by PAPrijmeni ASC
</SCRIPT>
<CSP:WHILE CONDITION=pacient.Next()>
<script LANGUAGE="CACHE" RUNAT="SERVER">
// zajisti vypis vybranych prijmeni pres promenu vypsati
if (pacient.Data("PAPrijmeni"))[hledatjm { set vypsati="1" }
else { set vypsati="0" }
</script>
<CSP:IF CONDITION=vypsati="1">
<tr>
<td width="13%">
<a href="#(namef)#?jmeno=#(%request.Data("jmeno",1))#&heslo=#(%request.Data("heslo",1))#&obsah=dfj&hledat=1&hledatpacc=1&hled

```

```

atpacjm=#(pacient.Data("PAPrijmeni"))#&#hledatpacrc=#(pacient.Data("PACisloRodne"))#&#hledatpacc=#(hledatjm)#">&nbsp;#(pacient.D
ata("PAPrijmeni"))# - zobrazit detaily &nbsp;</a></td>
<td width="12%">&nbsp;#(pacient.Data("PAJmeno"))#</td>
<td width="12%">&nbsp;#(pacient.Data("PAMesto"))#</td>
<td align="center" width="10%">#(pacient.Data("PACisloRodne"))#</td>
</tr>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set b=b+1 // zapis pocit. nalez. prijmeni
</script>
<CSP:ELSE>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set a=a+1 // zapis pocitadla nenalez. prij
</script>
</CSP:IF>
</CSP:WHILE>
</table></td></table><br>
<font color="red" size="-1"> #(b)# - po#269;et nalezených záznam&#367; s podobným p&#345;ijmením </font><br>
<font color="green" size="-1"> #(a+b)# - po#269;et v&#353;ech nalezených záznam&#367; dle R&#268; </font><br>
</CSP:IF>
<CSP:COMMENT> tlačitko zpět </CSP:COMMENT>
<div align="left">
<form name="form1" method="post" action="#(namef)#">
<input type="hidden" name="hledat" value="">
#(..InsertHiddenField("#(namef)#","jmeno",(%request.Data("jmeno",1)))#
#(..InsertHiddenField("#(namef)#","heslo",(%request.Data("heslo",1)))#
#(..InsertHiddenField("#(namef)#","obsah",(%request.Data("obsah",1)))#
&nbsp;<input type="submit" name="zpet" value="Zp&#283;t na zadání p&#345;ijmení">
</form></div>
</CSP:IF>
<CSP:COMMENT> obsah dle jmena obsah=h </CSP:COMMENT>
<CSP:ELSEIF CONDITION="$G(%request.Data("obsah",1))["h"]">
<CSP:COMMENT> zobrazení informací pro vybraného pacienta </CSP:COMMENT>
<CSP:IF CONDITION="$G(%request.Data("infopac",1))="1"">
<table class="tb1" width="85%"><td>
<table class="tb2" width="100%" align="center">
<tr style="color: black; background: silver;">
<td align="center"><b>DG kód</b></td>
<td><b> Popis diagnózy</b></td>
<td align="center"><b>DG Poprvé</b></td>
<td><b> P&#345;ijmení</b></td>
<td><b> Jméno</b></td>
<td align="center"><b>Rodné &#269;islo</b></td>
<td align="center"><b>Datum a &#269;as </b></td>
<td align="center"><b>Léka&#345;</b></td>
</tr>
<script LANGUAGE="CACHE" RUNAT="SERVER">
//deklarace nenalezene diagnozy.. nize
set nenalezenadg=0
</script>
<!-- Sloučení diagnóz tentokrát bez group by, sloučení diag. s pacienty, vyber pacientu podle diag -->
<SCRIPT LANGUAGE="SQL" NAME="diagnoz">
SELECT SUBSTRING(DGDIAGNOZA,1,3) AS DGZ, DGSkupinaDiagnoz, DGPoprve, DGICD10, ICICD10, ICPopis,
Aenderungsdatum, geaendert_durch,
SUBSTRING(Aenderungsdatum,1,4) AS drok, SUBSTRING(Aenderungsdatum,5,2) AS dmes, SUBSTRING(Aenderungsdatum,7,2)
AS dden,
SUBSTRING(Aenderungsdatum,9,2) AS dhod, SUBSTRING(Aenderungsdatum,11,2) AS dmin, PACisloRodne,PAJmeno,PAPrijmeni
FROM SQLUser.VDiagnoza
LEFT JOIN SQLUser.VICD10KS ON dgicd10=ICICD10
LEFT JOIN SQLUser.Vpacient ON PAPACIENT=SUBSTRING(DGDIAGNOZA,1,3)
LEFT JOIN SQLUser.Diagnosegruppe ON DGSkupinaDiagnoz=Diagnosegruppe ORDER by ICICD10
</SCRIPT>
<CSP:WHILE CONDITION=diagnoz.Next()>
<CSP:COMMENT> Vybere info. pacienta dle RC - preposlane z predchoziho vyberu </CSP:COMMENT>
<CSP:IF CONDITION="(diagnoz.Data("PACisloRodne"))=$G(%request.Data("hledatpacrc",1))">
<script LANGUAGE="CACHE" RUNAT="SERVER">
// Uprava poprve nul nezadana, 0 nezjistena, 1 Ano, 2 Ne
if (diagnoz.Data("DGPoprve"))="0" { set poprve="Nezji&#353;t&#283;no" }
elseif (diagnoz.Data("DGPoprve"))="1" { set poprve="Ano" }
elseif (diagnoz.Data("DGPoprve"))="2" { set poprve="Ne" }
else { set poprve="Nezadáno" }
// Kontrola prazdne diagnozy
if (diagnoz.Data("DGICD10"))="" { set nenalezenadg=nenalezenadg+1 }
</script>
</tr>
<td align="center" width="10%">#(diagnoz.Data("DGICD10"))#</td>
<td width="25%">&nbsp;#(diagnoz.Data("ICPopis"))# &nbsp;</td>

```



```

<td align="center" width="10%">#(poprve)#</td>
<td width="13%">&nbsp;#(diagnoz.Data("PAPrijmeni"))#</td>
<td width="12%">&nbsp;#(diagnoz.Data("PAJmeno"))#</td>
<td align="center" width="10%">#(diagnoz.Data("PACisloRodne"))#</td>
<td align="center" width="15%">#(diagnoz.Data("dden"))#.#(diagnoz.Data("dmes"))#.#(diagnoz.Data("drok"))#
#(diagnoz.Data("dhod"))#.#(diagnoz.Data("dmin"))#</td>
<td align="center" width="5%">#(diagnoz.Data("geaendert_durch"))#</td>
</tr>
</CSP:IF>
</CSP:WHILE>
</table></td>
</table><br>
<script LANGUAGE="CACHE" RUNAT="SERVER">
// pri nenalezene diag. u pacienta vypise
if nenalezenadg=0 { write " <font color=red size=-1> U pacienta nebyla nalezena &#382;ádná diagnóza </font>" }
</script>
<CSP:COMMENT> BIND - tlačitko zpet </CSP:COMMENT>
<div align="left">
<form name="formback" method="post" action="#(namef)#">
<input type="hidden" name="hledat" value="1">
<input type="hidden" name="rokod" value="#($G(%request.Data("brokod",1)))#">
<input type="hidden" name="rokdo" value="#($G(%request.Data("brokdo",1)))#">
<input type="hidden" name="mesod" value="#($G(%request.Data("bmesod",1)))#">
<input type="hidden" name="mesdo" value="#($G(%request.Data("bmesdo",1)))#">
<input type="hidden" name="registrNR" value="#($G(%request.Data("bregistrNR",1)))#">
#(..InsertHiddenField("#(namef)#","jmeno",(%request.Data("jmeno",1)))#
#(..InsertHiddenField("#(namef)#","heslo",(%request.Data("heslo",1)))#
#(..InsertHiddenField("#(namef)#","obsah",(%request.Data("obsah",1)))#
&nbsp;#<input type="submit" name="zpet" value="Zp&#283; na výsledky hledání">
</form>
</div>
<CSP:ELSE>
<CSP:COMMENT> umozni zvolit-vybrat rozsah hledani-pokud neni jeste zvolen </CSP:COMMENT>
<CSP:IF CONDITION='!$G(%request.Data("hledat",1))="">
<CSP:COMMENT> vyberovej select.. vygeneruje roky, jenz DB obsahuje </CSP:COMMENT>
<SCRIPT LANGUAGE="SQL" NAME="kodyd">
SELECT SUBSTRING(Aenderungsdatum,1,4) AS drok
from SQLUser.Diagnosegruppe Group by SUBSTRING(Aenderungsdatum,1,4) order by SUBSTRING(Aenderungsdatum,1,4) ASC
</SCRIPT>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set irok=0
</script>
<form name="form" method="post" action="#(namef)#">
#(..InsertHiddenField("#(namef)#","jmeno",(%request.Data("jmeno",1)))#
#(..InsertHiddenField("#(namef)#","heslo",(%request.Data("heslo",1)))#
#(..InsertHiddenField("#(namef)#","obsah",(%request.Data("obsah",1)))#
<br>
<fieldset style="width: 700px;">
<legend style="color: green;">Zadejte prosím po&#382;adovaný Národní zdravotní registr NRZ a &#269;asový interval</legend>
<br>Zvolte prosím po&#382;adovaný registr:
<select name="registrNR">
<option style="color: black;" value="">Zvolte prosím konkrétní registr</option>
<option style="color: black;" value="NRHOSP">Národní registr hospitalizovaných (NRHOSP)</option>
<option style="color: black;" value="NOR">Národní onkologický registr (NOR)</option>
</select><br><br><table><tr><td>
Zvolte po&#269;átek intervalu:</td></tr>
<select name="rokod">
<option style="color: black;" value="">rok</option>
<CSP:WHILE CONDITION=kodyd.Next()>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set rokvyb(irok)=kodyd.Data("drok")
set irok=irok+1
</script>
<option style="color: black;" value="#(kodyd.Data("drok"))#">#(kodyd.Data("drok"))#</option>
</CSP:WHILE>
</select>
<select name="mesod">
<option style="color: black;" value="">m&#283;sic</option>
<option style="color: black;" value="01">leden</option><option style="color: black;" value="02">únor</option><option style="color:
black;" value="03">b&#345;ezen</option><option style="color: black;" value="04">duben</option><option style="color:
black;" value="05">kv&#283;ten</option><option style="color: black;" value="06">&#269;erven</option><option style="color:
black;" value="07">&#269;ervenec</option><option style="color: black;" value="08">srpen</option><option style="color:
black;" value="09">zá&#345;i</option><option style="color: black;" value="10">&#345;ijen</option><option style="color:
black;" value="11">listopad</option><option style="color: black;" value="12">prosinec</option>
</select>
</td></tr><tr><td>Zvolte konec intervalu:</td></tr>

```

```

<select name="rokdo">
  <option style="color: black;" value="">rok</option>
  <script LANGUAGE="CACHE" RUNAT="SERVER">
    for i=0:1:(irok-1) {
      write "<option style=color:black; value=",rokvyb(i),">,<option>" }
    </script>
  </select>
  <select name="mesdo">
    <option style="color: black;" value="">m&#283;sic</option>
    <option style="color: black;" value="01">leden</option><option style="color: black;" value="02">únor</option><option style="color:
black;" value="03">b&#345;ezen</option><option style="color: black;" value="04">duben</option><option style="color:
black;" value="05">kv&#283;ten</option><option style="color: black;" value="06">&#269;erven</option><option style="color:
black;" value="07">&#269;ervenec</option><option style="color: black;" value="08">srpen</option><option style="color:
black;" value="09">zá&#345;i</option><option style="color: black;" value="10">&#345;ijen</option><option style="color:
black;" value="11">listopad</option><option style="color: black;" value="12">prosinec</option>
  </select>
</td></tr></table><br>
<input type="hidden" name="nejmrok" value=#(rokvyb(0))#>
<input type="hidden" name="nejvrok" value=#(rokvyb(irok-1))#>
<input type="hidden" name="hledat" value="1">
<input type="submit" name="hledatvyb" value="Vyhledat zvolená data z databáze">
<br><br>
<font size="-2" color="#195481">Poznámka: Zvolte po&#382;adovaný interval od-do u nezvoleného údaje se bere automaticky
minimum/maximum.</font><br><br>
</fieldset> <br> </form>
<CSP:COMMENT> Jeli interval zvolen ... pokračuje </CSP:COMMENT>
<CSP:ELSE>
  <div id="noprint">
    <script LANGUAGE="CACHE" RUNAT="SERVER">
      // kontrola roku jestli je zadany spravny interval
      if $G(%request.Data("rokod",1))="" {
        write "<font color=green size=-2>Nezadali jste: Rok OD >> vypl&#328;uji nejni&#382;&#353;i rok z DB:
",($G(%request.Data("nejmrok",1))),"</font><br>"
        set rokod=$G(%request.Data("nejmrok",1))
      } else { set rokod=$G(%request.Data("rokod",1))
      if $G(%request.Data("rokdo",1))="" {
        write "<font color=green size=-2>Nezadali jste: Rok DO >> vypl&#328;uji nejvy&#353;&#353;i rok z DB:
",($G(%request.Data("nejvrok",1))),"</font><br>"
        set rokdo=$G(%request.Data("nejvrok",1))
      } else { set rokdo=$G(%request.Data("rokdo",1))
      if rokod>rokdo {
        write "<font color=red>Zadali jste &#353;patn&#283; interval: ROK OD-DO (OD je VET&#353;i ne&#382; DO)</font><br>"
        set pokračovat=0
      }
      else {
      if $G(%request.Data("mesod",1))="" {
        write "<font color=green size=-2>Nezadali jste: M&#283;sic OD >> vypl&#328;uji 01. m&#277;sic </font><br>"
        set mesod="01"
      } else { set mesod=$G(%request.Data("mesod",1))
      }
      if $G(%request.Data("mesdo",1))="" {
        if rokdo=($Extract($ZDate($Horolog,4),7,10)) {set mesdo=($Extract($ZDate($Horolog,4),4,5)) } // DO AKTUALNIHO MESICE (
mozna bude lepsi mesic - 1..)
        else { set mesdo="12"
        write "<font color=green size=-2>Nezadali jste: M&#283;sic DO >> vypl&#328;uji ",mesdo,". m&#277;sic </font><br>"
        }
      } else { if (rokdo=($Extract($ZDate($Horolog,4),7,10))) & (($G(%request.Data("mesdo",1))>($Extract($ZDate($Horolog,4),4,5)))) {
        set mesdo=($Extract($ZDate($Horolog,4),4,5))
        else { set mesdo=$G(%request.Data("mesdo",1))
        }
      }
      if $G(%request.Data("registrNR",1))="" {
        write "<font color=green size=-2>Nezadali jste po&#382;adovaný registr >> vypl&#328;uji NRHOSP </font><br>"
        set registrNR="NRHOSP"
      } else { set registrNR=$G(%request.Data("registrNR",1))
      set pokračovat=1
      if (rokod=rokdo) & (mesod>mesdo) {
        write "<font color=red>Zadali jste &#353;patnou &#269;asovou-m&#277;st&#269;ní posloupnost (DO je ni&#382;&#353;i ne&#382;
OD) pro rok: ",rokod,".</font><br>"
        set pokračovat=0
      }
    }
  </script>
</div>
<CSP:COMMENT> pokud vyhovuji podminky(predchozi kontrola) povoli pokračovani </CSP:COMMENT>
<CSP:IF CONDITION="pokracovat="1">
  <br><br>
  <div id="print">Ti&#353;t&#277;ná podoba dat ze dne #($ZDate($Horolog,4))#, registr: #(registrNR)# <br></div>

```

```

<CSP:IF CONDITION="registrNR="NRHOSP">
<script LANGUAGE="CACHE" RUNAT="SERVER">
//deklarace pomocnych promennych a vypocet mesicu pro sestaveni matic
set pocetcyklu=((rokdo-rokod)*12)+(mesdo-mesod)+1
for i=1:1:pocetcyklu {
set mz(i)=1 }
</script>
<SCRIPT LANGUAGE="SQL" NAME="diagnozRZ">
SELECT PIPRIJEM,PPPpropusteni, SUBSTRING(PIPRIJEM,9,4) AS Iprok, SUBSTRING(PIPRIJEM,13,2) AS Ipmes,
SUBSTRING(PIPRIJEM,15,2) AS Ipdn, SUBSTRING(PIPRIJEM,17,2) AS Iphod, SUBSTRING(PIPRIJEM,19,2) AS Ipmi,
SUBSTRING(PPPpropusteni,9,4) AS Oprok, SUBSTRING(PPPpropusteni,13,2) AS Opmes, SUBSTRING(PPPpropusteni,15,2)
AS Opden, SUBSTRING(PPPpropusteni,17,2) AS Ophod, SUBSTRING(PPPpropusteni,19,2) AS Opmin,
PPEpizoda, PIEpizoda,PIPokoj,PIStanice,PITarif,
PAPrijmeni,PACisloRodne,PAJmeno
FROM SQLUser.VPrijem
LEFT JOIN SQLUser.Vpropusteni ON PIEpizoda=PPEpizoda
LEFT JOIN SQLUser.VPacient ON PAPacient=SUBSTRING(PIEpizoda,1,3)
WHERE (PACisloRodne IS NOT NULL) AND (PPPpropusteni IS NULL OR
(SUBSTRING(PPPpropusteni,9,6)>={fn CONCAT("#(rokod)#,#(mesod)#}))) AND
(SUBSTRING(PIPRIJEM,9,6)<={fn CONCAT("#(rokdo)#,#(mesdo)#})))
Group by PACisloRodne ORDER by SUBSTRING(PIPrijem,9,12) ASC
</SCRIPT>
<CSP:WHILE CONDITION=diagnozRZ.Next()>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set pmp=((diagnozRZ.Data("Iprok"))-rokod)*12+((diagnozRZ.Data("Ipmes"))-mesod)+1 // pomocna promena vypocet od jake
matice se ma zaznamenavat
if pmp<1 {set mp=1}
else {set mp=pmp}
if ((diagnozRZ.Data("Oprok"))="") {set mk=pocetcyklu}
else {set pmk=((diagnozRZ.Data("Oprok"))-rokod)*12+((diagnozRZ.Data("Opmes"))-mesod)+1 // pomocna pr. vypocet az do jake
matice se ma zaznamenavat
if pmk>pocetcyklu {set mk=pocetcyklu}
else {set mk=pmk}
}
for i=1:1:(pocetcyklu) {
if (mp<=i) & (mk>=i) {
set M(i,mz(i),1)=(diagnozRZ.Data("PAPrijmeni"))
set M(i,mz(i),2)=(diagnozRZ.Data("PAJmeno"))
set M(i,mz(i),3)=(diagnozRZ.Data("PACisloRodne"))
set M(i,mz(i),4)=(diagnozRZ.Data("Ipdn"))
set M(i,mz(i),5)=(diagnozRZ.Data("Ipmes"))
set M(i,mz(i),6)=(diagnozRZ.Data("Iprok"))
set M(i,mz(i),7)=(diagnozRZ.Data("Iphod"))
set M(i,mz(i),8)=(diagnozRZ.Data("Ipmi"))
set M(i,mz(i),9)=(diagnozRZ.Data("PIPokoj"))
set M(i,mz(i),10)=(diagnozRZ.Data("PIStanice"))
set M(i,mz(i),11)=(diagnozRZ.Data("PITarif"))

set mz(i)=(mz(i))+1
} }
</script>
</CSP:WHILE>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set vmesod=mesod
set vrokod=rokod
for i=1:1:pocetcyklu { // nebo vypis obracen i=pocetcyklu:-1:1b ++ uprava pocitani mesice a roku

write "<fieldset style=width:700px;><legend style=color:green;> Registr: ",registrNR," - ",vmesod,"/",vrokod,"
</legend><br>"
write "<table class=tb1><td><table class=tb2 align=center><tr style=background-
color:silver;><td><b>P&#345;ijmeni</b></td><td><b>Jmeno</b></td><td align=center><b>Rodne &#269;islo</b></td><td
align=center><b>P&#345;ijat</b></td><td align=center><b>Pokoj</b></td><td align=center><b>Stanice</b></td><td
align=center><b>Tarif</b></td></tr>"
for z=1:1:(mz(i)-1) {
write "<tr><td><div id=noprint><a title=",M(i,z,1),"
href=",namef,"?jmeno=",(%request.Data("jmeno",1)),"&heslo=",(%request.Data("heslo",1)),"&obsah=h&infopac=1&brokod=",rokod,"&br
okdo=",rokdo,"&bmesod=",mesod,"&bmesdo=",mesdo,"&bregistrNR=",registrNR,"&hledatpacrc=",M(i,z,3),"&nbsp; ",M(i,z,1)," -
dne-v
write "<td align=center> ",M(i,z,2)," </td>" //Jmeno
write "<td align=center> ",M(i,z,3)," </td>" //RC
write "<td align=center> ",M(i,z,4)," ",M(i,z,5)," ",M(i,z,6)," v ",M(i,z,7)," ",M(i,z,8)," </td>" // Prijat
write "<td align=center> ",M(i,z,9)," </td>" //PIPokoj
write "<td align=center> ",M(i,z,10)," </td>" //PIStanice
write "<td align=center> ",M(i,z,11)," </td></tr>" //PITarif
}
}
}

```



```
<CSP:COMMENT> Volne info.. vytvoril.. posledni uprava </CSP:COMMENT>
<font size="-1" color="#003399">
<!-- Jste nav&#353;t&#283;vnik &#269;islo <b>#($Increment(^NavstevnikCislo))#</b> na této stránce.<br>
Vytvo&#345;il v rámci magisterské práce: Martin Holub<br></font> -->
<font size="-2">Copyright &copy; Martin Holub a VUT FEKT Brno 2006-2010</font><br>
<font size="-2">Poslední úprava provedena: ##($ZDATETIME($H,4))##
</font>
<CSP:COMMENT> div konce a layoutu </CSP:COMMENT>
</div></div>
<CSP:COMMENT> elseif přihlasovaciho if (kontrola hesla) (dodatek pokud admin : ) </CSP:COMMENT>
<CSP:ELSEIF CONDITION='$G(%request.Data("heslo",1))="admin"'>
<div id="inout">
<CSP:IF CONDITION='$G(%request.Data("pridat",1))["P&#345;idat"]>
Přidávám u&#345;ivatele: Jméno: #($G(%request.Data("login",1)))#<br> Heslo:#($G(%request.Data("password",1)))#<br>
<SCRIPT LANGUAGE="SQL" NAME="uzivatele">
  INSERT INTO SQLUser.uzivatele VALUES ('5','aaa','dasdasd')
</SCRIPT>
</CSP:IF>
<br><br><h2>Vítejte v administraci u&#345;ivatel&#367; </h2> p&#345;ihlá&#353;en:
<a href="#"(namef)#?jmeno=&heslo=&odhlasit=Odhlasit">#(%request.Data("heslo",1))#&nbsp;&nbsp;&nbsp;<b>ODHLASIT</b></a>
<br><br>
<CSP:IF CONDITION='$G(%request.Data("edit",1))["Ulo"]>
  Ulo&#382;it zm&#277;ny v ID2:#($G(%request.Data("ids",1)))# <br>Jméno: #($G(%request.Data("login",1)))#<br>
  Heslo:#($G(%request.Data("password",1)))#<br>
  <SCRIPT LANGUAGE="SQL" NAME="update">
    UPDATE SQLUser.uzivatele SET login="ccc",password="cccc" WHERE ids=($G(%request.Data("ids",1)))
  </SCRIPT>
<CSP:ELSEIF CONDITION='$G(%request.Data("edit",1))["Smaz"]>
  Smazat ID2:#($G(%request.Data("ids",1)))# <br>Jméno: #($G(%request.Data("login",1)))#<br>
  Heslo:#($G(%request.Data("password",1)))#<br>
  <SCRIPT LANGUAGE="SQL" NAME="drop">
    DELETE FROM SQLUser.uzivatele WHERE ids=($G(%request.Data("ids",1)))
  </SCRIPT>
</CSP:IF>
<SCRIPT LANGUAGE="SQL" NAME="uzivatele">
  SELECT * From SQLUser.uzivatele Order by ID
</SCRIPT>
<fieldset style="width: 650px;"><br>
<CSP:WHILE CONDITION=uzivatele.Next()>
<form name="formse" method="post" action="#"(namef)#">
  #(.InsertHiddenField("#(namef)#","jmeno",(%request.Data("jmeno",1)))#
  #(.InsertHiddenField("#(namef)#","heslo",(%request.Data("heslo",1)))#
  ID2: #(uzivatele.Data("ids"))# &nbsp;&nbsp;&nbsp;
  <input type="hidden" name="ids" value="#"(uzivatele.Data("ids"))#>
  Jméno: <input type="text" style="color: blue;" name="login" value="#"(uzivatele.Data("login"))# > &nbsp;&nbsp;&nbsp; &nbsp;&nbsp;&nbsp;
  Heslo: <input type="text" style="color: red;" name="password" value="#"(uzivatele.Data("password"))# >
  <input type="submit" name="edit" value="Ulo&#382;it">
  <input type="submit" name="edit" value="Smazat">
</form>
</CSP:WHILE>
</fieldset> <br><br>
<h3>P&#345;idat nového u&#382;ivatele:</h3>
<form name="form" method="post" action="#"(namef)#">
<fieldset style="width: 650px;">
  #(.InsertHiddenField("#(namef)#","jmeno",(%request.Data("jmeno",1)))#
  #(.InsertHiddenField("#(namef)#","heslo",(%request.Data("heslo",1)))#
  Jméno: <input type="text" style="color: blue;" name="login" value=" login" onfocus="this.value="">
  Heslo: <input type="text" style="color: red;" name="password" value=" password" onfocus="this.value="">
  <input type="submit" name="pridat" value="P&#345;idat u&#382;ivatele">
</fieldset> <br><br></form> </div>
<CSP:COMMENT> else přihlasovaciho if (kontrola hesla) </CSP:COMMENT>
<CSP:ELSE>
<div id="inout">
<h1><br>Vítejte ve webové aplikaci ur&#269;ené k p&#345;ístupu<br></h1><h2> do nemocni&#269;ního informa&#269;ního systému
CLINICOM<br></h2><h3>na ÚBEMI, FEKT, VUT v Br&#283;<br></h1><br><br><br>
<CSP:COMMENT> Pokud je přihlasit=přihlasit a heslo nesouhlasí </CSP:COMMENT>
<CSP:IF CONDITION='$G(%request.Data("prihlas",1))["P"]>
<h3><font color="red">&#352;patn&#283; zadané jméno nebo heslo, zkuste to znovu:</font></h3><br><br>
</CSP:IF>
<CSP:COMMENT> Pokud je odhlasit=odhlasit </CSP:COMMENT>
<CSP:IF CONDITION='$G(%request.Data("odhlasit",1))["O"]>
<h3><font color="green">Prob&#283;hlo úspě&#283;&#353;né odhlá&#353;ení u&#382;ivatele.</font></h3><br><br>
</CSP:IF>
<CSP:COMMENT> pokud se nejedna o špatně zadane heslo ani o odhlášení provede PRIHLASENI </CSP:COMMENT>
<form name="form2" method="post" action="#"(namef)#">
<input type="text" name="jmeno" value="Zadejte Jméno" onfocus="this.value="">
```

```

<input type="text" name="heslo" value="Zadejte Heslo" onfocus="this.value="">
<input type="submit" name="prihlas" value="P&#345;ihlásit">
</form>
</div>
<font size="-2">Copyright &copy; Martin Holub a VUT FEKT Brno 2006-2010</font><br>
</CSP:IF>
</center>
</body>
</html>

```

Soubor ds.csp:

```

<HTML>
<HEAD>
<TITLE> Vygenerovaný vzorový kód v DS 3 - Diplomová práce, Martin Holub a VUT Brno</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250" />
<style type="text/css">
<!--
table { border-style: none ; font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-style: normal; }
input,select { border: 1px #000033 solid; background-color: #DAF0ED; font-family: Arial, Helvetica, sans-serif; font-size: 12px; font-weight: bold; color: #003399; text-decoration: none; }
BODY { SCROLLBAR-FACE-COLOR: #DAF0ED; SCROLLBAR-HIGHLIGHT-COLOR: #DAF0ED; SCROLLBAR-SHADOW-COLOR: #003399; SCROLLBAR-3DLIGHT-COLOR: #ffffff; SCROLLBAR-ARROW-COLOR: #003399; SCROLLBAR-TRACK-COLOR: #ffffff; SCROLLBAR-DARKSHADOW-COLOR: #003399; }
-->
</style>
</HEAD>

<BODY>
<center>
<script LANGUAGE="CACHE" RUNAT="SERVER">
set overeni=((($ZDATE($HOROLOG,8))*1.35)*$EXTRACT($ZTIME($PIECE($HOROLOG,"",2),1,9),1,2))
if $G(%request.Data("ove",1))=(overeni) write "<font color=green>Autorizace: ..P&#345;istup povolen.</font><br>"
else write "<font color=red><br><b>P&#345;istup zamítnut! <br>(pravd&#277;podobn&#277; vyprchal &#269;asovy
limit)</b></font><br>"
set rokod=$G(%request.Data("rokod",1))
set mesod=$G(%request.Data("mesod",1))
set rokdo=$G(%request.Data("rokdo",1))
set mesdo=$G(%request.Data("mesdo",1))

set pacopak=0
set opak=0
set zakoncitip="ne"
</script>
<CSP:IF CONDITION=$G(%request.Data("ove",1))=overeni>
<br><u>Nalezená data pro zvolený registr/interval/diagnózu, zobrazená ve formátu DS 3:</u><br>
<table bordercolor="#003399" border="1" width="65%">
<tr><td align="center"><b>V<br>z<br>o<br>r<br>o<br>v<br>y<br> &nbsp;<br>v<br>y<br>g<br>e<br>n<br>e<br>r<br>o<br>v<br>
a<br>n<br>y<br> &nbsp;<br>k<br>ó<br>d<br></b></td>
<td align="center">&lt;?xml version='1.0' encoding='Windows-1250' standalone='no' ?&gt;
<br>&nbsp;<br>&lt;DOCTYPE dasta SYSTEM "ds030701.dtd"&gt;
<br>&nbsp;<br>&lt;dasta
id_soubor="VUT_UBMI_00_00000_#($ZDATE($Horolog,3))#T#($ZTIME($PIECE($HOROLOG,"",2),1))#" verze_ds="03.01.01"
ur="T" potvrzeni="N"&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;zdroj_is kod_firmy="VUT_UBMI" kod_prog="CSP" verze_prog="xx" liccis_prog="xx" /&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;pm ico="12345678" &gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;as typ="E" &gt; <!-- blok adresa-spojenci cil -->
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;obsah&gt;adresat@vutbr.cz&lt;/obsah&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;/as&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;a typ="X" &gt; <!-- blok adresa X-jina -->
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;jmeno&gt;Jmeno Prijmeni (firma) &lt;/jmeno&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;psc&gt;630 00&lt;/psc&gt; <!-- max 9 znaku -->
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;mesto&gt;Brno&lt;/mesto&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;stat&gt;CZ&lt;/stat&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;a&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;/a&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;pm&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;/pm&gt;
<br>&nbsp;&nbsp;&nbsp;&nbsp;&lt;icz="12345678"&gt; <!-- odesilatel -->
<!-- Sloučení diagnóz tentokrát bez group by, sloučení diag. s pacienty, vyber pacientu podle diag -->
<SCRIPT LANGUAGE="SQL" NAME="diagnozDS">
SELECT SUBSTRING(DGDIAGNOZA,1,3) AS DGZ, DGSkupinaDiagnoz, DGPoprve, DGICD10,SUBSTRING(DGICD10,2,5)
AS CD10, ICICD10, ICPopis, Aenderungsdatum, geaendert_durch,
SUBSTRING(Aenderungsdatum,1,4) AS dsdrok, SUBSTRING(Aenderungsdatum,5,2) AS dsdmes,
SUBSTRING(Aenderungsdatum,7,2) AS dsdden,
SUBSTRING(Aenderungsdatum,9,2) AS dhod, SUBSTRING(Aenderungsdatum,11,2) AS dmin

```