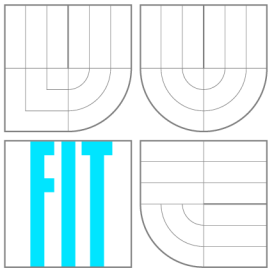


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

LÉKAŘSKÝ SOFTWARE PRO VYÚČTOVÁNÍ DOKLADŮ

MEDICAL SOFTWARE FOR ACCOUNTING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ POHL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MATĚJ GRÉGR

BRNO 2013

Abstrakt

Cílem práce je rozšíření lékařského softwaru OpenDositech. Rozšířením je implementace rozhraní pro elektronickou komunikaci s webovými portály zdravotních pojišťoven do programu OpenDositech. Nejprve je popsán dostupný lékařský software spolu s datovým rozhraním pro předávání lékařských dokladů zdravotním pojišťovnám. Práce se poté zabývá předáváním dokladů přes Portál zdravotních pojišťoven. V závěru práce je popsána implementace rozhraní do programu OpenDositech.

Abstract

The aim of this thesis is an extension of medical software OpenDositech. Implementation of interface for communication with portals of insurance companies is extension into OpenDositech program. In the first part available medical softwares are described together with API for sending. The thesis next deals with the Portal of insurance companies. At the end the thesis describes implementation of the API into OpenDositech program.

Klíčová slova

OpenDositech, pojišťovna, Qt framework, rozhraní, portál, certifikát, dávka, podpis

Keywords

OpenDositech, insurance company, Qt framework, interface, portal, certificate, batch, signature

Citace

Ondřej Pohl: Lékařský software pro vyúčtování dokladů, bakalářská práce, Brno, FIT VUT v Brně, 2013

Lékařský software pro vyúčtování dokladů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Matěje Grégra. Uvedl jsem všechny zdroje, ze kterých jsem čerpal.

.....

Ondřej Pohl
15. května 2013

Poděkování

Děkuji vedoucímu Ing. Matěji Grégrovi za cenné připomínky, rady a přínosné konzultace během řešení této práce. Chci také poděkovat panu Ing. Vladimíru Benešovi z Portálu ZP za vstřícnost při řešení problémů během testování aplikace.

© Ondřej Pohl, 2013.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	1
2 Lékařský software	2
2.1 Lékařský software společnosti Themis	2
2.2 Lékařský software společnosti CompuGroup Medical	3
2.3 Další lékařský software	4
3 Předávání dokladů zdravotní pojišťovně	5
3.1 Základní pojmy	5
3.2 Proces předávání dokladů	6
3.3 Datové rozhraní číselníků	7
3.4 Datové rozhraní dokladů	8
4 Portál zdravotních pojišťoven	9
4.1 Portál ZP	9
4.2 Registrace na Portál ZP	10
4.3 Služby Portálu ZP	11
5 Elektronické rozhraní Portálu ZP	13
5.1 Komunikační brána	13
5.2 Princip komunikace	13
5.3 Struktura XML zprávy	14
5.4 XML komunikace pro vyúčtování zdravotní péče	15
5.5 Demo aplikace knihovny ASSECOPZP	17
6 Implementace rozšíření do aplikace OpenDositech a testování	18
6.1 Základní program OpenDositech	18
6.2 Předávání dávek dokladů na Portál ZP	20
6.3 Další úpravy aplikace OpenDositech	29
6.4 Testování	31
7 Závěr	33
Literatura	33
A Obsah CD	35
B Ukázky datových rozhraní	36
C Portál ZP	37

Kapitola 1

Úvod

Využití výpočetní techniky ve zdravotnictví je dnes již samozřejmostí a nezbytností. Počítače postupně našly své místo od velkých nemocnic až po malé ordinace. Lékař v ordinaci používá počítač především k vedení lékařské dokumentace. K tomuto účelu využívá software, který mu usnadňuje potřebnou administraci. Lékař také software využívá pro komunikaci s pojišťovnou, které předkládá doklady k vyúčtování za lékařskou péči.

Cílem této práce je rozšíření programu OpenDositech, který umožňuje vytvářet vyúčtování pro zdravotní pojišťovny. Rozšíření realizuje předávání dokladů přes Portál zdravotních pojišťoven využívající elektronické rozhraní portálu.

Následující kapitola představuje vybraný lékařský software dostupný v současné době na českém trhu. Náplní kapitoly je charakteristika jednotlivých lékařských programů a zhodnocení možností jejich využití.

Tématem třetí kapitoly je vysvětlení základních principů komunikace lékaře a zdravotní pojišťovny při předávání dokladů. Zabývám se zde výkladem základních pojmů a datovým rozhraním.

Čtvrtá kapitola je věnována Portálu zdravotních pojišťoven. Zde jsem se zaměřil na popis portálu, možnosti přístupu a služby dostupné pro uživatele.

Pátá kapitola pojednává o elektronickém rozhraní Portálu zdravotních pojišťoven. Cílem kapitoly je popis způsobů přímé komunikace programu běžícího na počítači lékaře s portálem zdravotní pojišťovny.

Jádrem šesté kapitoly je popis implementace rozhraní pro předávání dokladů na portál do aplikace OpenDositech. Nejprve je představena původní verze OpenDositechu a poté se věnuji popisu všech úprav, které jsem v této práci implementoval. Testování je tématem závěru kapitoly.

Kapitola 2

Lékařský software

Tvorbou lékařského softwaru se v České republice zabývá mnoho firem. Lékař má na výběr množství programů, které se liší především cenou a funkcemi. V následující kapitole bude čtenář seznámen s produkty několika výrobců lékařských softwarů.

2.1 Lékařský software společnosti Themis

Společnost Themis¹ nabízí lékařům programy OBOLUS, DOSITECH 8 a Dech32 [6]. Podrobněji se zaměřím na software DOSITECH 8, jelikož je inspirací pro program Open-Dositech, kterému se budu věnovat v pozdější části práce.

Program DOSITECH 8² zajišťuje vytvoření dávky s doklady pro pojišťovnu a umožňuje vést i základní lékařskou dokumentaci. Dávku lze umístit na disketu, CD nebo připravit pro odeslání přes Portál zdravotních pojišťoven. DOSITECH 8 však není software, který přímo s Portálem zdravotních pojišťoven komunikuje. Na obrázku 2.1 je ukázka menu z programu DOSITECH 8.

Lékař si vystačí při ovládání programu pouze s klávesnicí. Přepsání papírových dokladů do elektronické podoby je takto velmi efektivní. Používání počítačové myši při přepisování údajů z papírových dokladů by bylo spíše na obtíž.

Mezi hlavní funkce programu DOSITECH 8 patří:

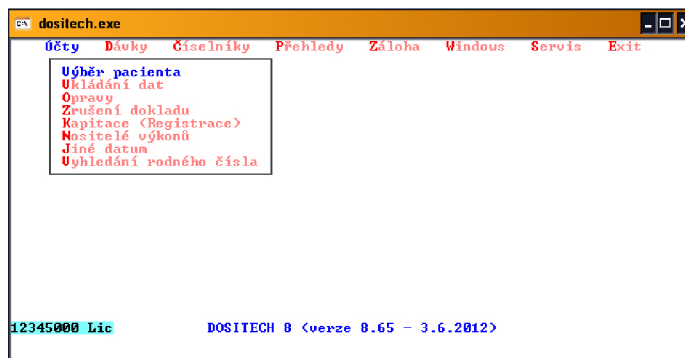
- Kartotéka pacientů
- Vkládání účtů
- Vytvoření dávky
- Číselníky
- Zálohování
- Přehledy a statistiky

DOSITECH 8 je určen původně pro DOS³, ale není problém jej spustit v prostředí Windows. Přímo pro Windows je určena novější podoba programu DOSITECH 8 s názvem Dech32.

¹<http://www.themis.cz>

²<http://www.themis.cz/dositech/dositech.htm>

³Disk Operating System



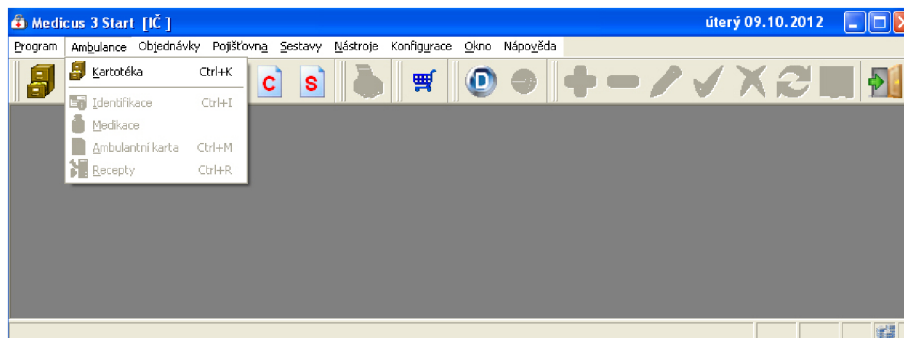
Obrázek 2.1: Menu programu DOSITECH 8 (barvy invertovány)

2.2 Lékařský software společnosti CompuGroup Medical

Programy MEDICUS 3 společnosti CompuGroup Medical⁴ označují řadu produktů určených do ordinací lékařů nebo velkých zdravotnických zařízení [11]:

- MEDICUS 3 – pro praktické i odborné lékaře
- MEDICUS 3 Stomatolog – pro stomatology
- MEDICUS 3 NIS – pro menší nemocnice
- MEDICUS 3 SPA – lázeňský IS⁵ (rozšíření programu MEDICUS 3 NIS)

Programy MEDICUS 3 nebo MEDICUS 3 Stomatolog lze získat ve třech verzích: Start, Professional a Komfort. Jednotlivé verze se liší dostupnými funkcemi a možnostmi připojení rozšiřujících modulů. Ukázku z programu MEDICUS 3 Start lze vidět na obrázku 2.2.



Obrázek 2.2: Ukázka menu programu MEDICUS 3 Start

Lékařský software MEDICUS 3 Start se vyznačuje podobnými funkcemi jako DOSITECH 8. Samozřejmostí je kartotéka, vkládání dokladů, vytváření dávek, přehledy a zálohování. MEDICUS 3 Start je možné rozšiřovat pomocí modulů.

Na rozdíl od programu DOSITECH 8 umožňuje MEDICUS 3 Start zasílat dávky portálům zdravotních pojišťoven elektronicky. Lékař má i přesto možnost využít uložení dávek na disketu. MEDICUS 3 Start je spustitelný v prostředí Windows XP a novějším.

⁴<http://www.medicus.cz/>

⁵Informační systém

2.3 Další lékařský software

Kromě již zmíněných programů DOSITECH 8 a MEDICUS 3 Start má lékař na výběr z mnoha jiných lékařských softwarů:

- J.H.Ambulance, J.H. Pediatr, J.H. Home Care⁶
- ORDIN REHAB, ORDIN HomeCare, ORDIN DOKTOR⁷
- DENTA, PRIVAT, OPTIK⁸

Ve většině lékařských programů je k dispozici kartotéka pacientů, vedení lékařské dokumentace a možnost vytvářet doklady a dávky. V základních funkcích se programy neliší. Lékařský software často doplňuje plno dalších funkcí, nástrojů a modulů podle toho, pro jakou specializaci je určen. Tím je ovlivněna i cena těchto produktů. Přehled vybraných lékařských programů s cenami a operačními systémy, na kterých je lze spustit, uvádí tabulka 2.1.

Tabulka 2.1: Přehled cen vybraných lékařských softwarů s podporovanými OS. Zdroj: [11, 6]

Lékařský software	Cena	OS
DOSITECH 8	1 800 Kč/rok	DOS, Windows 95/98/2000/XP/7
OBOLUS	3 200 Kč/rok	DOS, Windows 95/98/NT
MEDICUS 3 Start	7 920 Kč/program	Windows XP a novější
MEDICUS 3 Professional	15 696 Kč/program	Windows XP a novější

⁶<http://www.tomsussoftware.cz/>

⁷<http://www.msoft.uh.cz/>

⁸<http://www.tco.cz/software-pro-lekare.asp>

Kapitola 3

Předávání dokladů zdravotní pojišťovně

Lékař při poskytování zdravotní péče vykazuje provedené úkony ve formě dokladů. Doklady jsou pak v podobě dávek předávány pojišťovně. Pojišťovna následně provádí vyúčtování vykázané zdravotní péče. Takto zjednodušeně lze popsat proces předávání dokladů mezi lékaři a pojišťovnami. Pořizování, předávání dokladů a způsob vykazování zdravotní péče se řídí pravidly vydanými VZP¹. Zároveň pro předávání dat mezi lékařem a pojišťovnou na datových nosičích VZP určila datová rozhraní. Pravidla a datová rozhraní určená VZP jsou povinná pro všechny zdravotní pojišťovny [7].

3.1 Základní pojmy

Na úvod budou vysvětleny základní pojmy, jejichž výklad je nutný k pochopení problematiky předávání dokladů zdravotním pojišťovnám.

3.1.1 Doklad

Doklad slouží hlavně jako prostředek pro vyúčtování zdravotní péče při styku s pojišťovnami. Doklady se zároveň používají například k předepsání zdravotní péče nebo jako forma hlášení. Každý doklad je opatřen unikátním kódem a názvem. Lze se setkat s doklady [7]:

- v papírové formě – Jedná se o tiskopisy splňující formální a obsahové požadavky příslušného dokladu.
- v elektronické formě – Zde může jít o data obsažená na datovém nosiči (např. disketa, flash disk) splňující požadavky datového rozhraní, nebo lze místo datového nosiče využít předání dat přes elektronický portál.

3.1.2 Dávka

Dávka se skládá z jednoho nebo více dokladů. Doklady umístěné v dávce spadají do období, za které se provádí předání dávky k vyúčtování. Každá dávka je opatřena průvodním listem dávky. Průvodní list obsahuje informace pro kontrolu předávaných dokladů v dávce (identifikace dávky, počet dokladů v dávce, celková hodnota dávky). Identifikace dávky je určena rokem, měsícem a číslem dávky v rámci zdravotnického zařízení. V případě

¹Všeobecná zdravotní pojišťovna

předávání dávek elektronickou formou se dávky nachází v souboru KDAVKA. Pojišťovně lze předat nejen dávky určené k vyúčtování. V případě potřeby může lékař například předat dávku s registracemi pojištěnců nebo dávku se seznamem nositelů výkonů.

Podle charakteru dávky se rozlišuje:

- řádná dávka – Obsahuje pouze původní doklady.
- opravná dávka – Obsahuje opravené doklady, které byly dříve pojišťovnou odmítnuty (například proto, že doklad byl dříve nesprávně vyplněn).

Podle typu se může jednat o dávku:

- jednoduchou – V dávce se nacházejí doklady pouze jednoho druhu.
- smíšenou – Dávka obsahuje doklady více než jednoho druhu.

3.1.3 Číselník

Číselníky jsou soubory obsahující seznamy hodnot určitého druhu (např. číselník Typy lázeňské péče obsahuje seznam všech typů lázeňské péče). V dokladech se pak používají pouze hodnoty z číselníků (např. pokud je v dokladu nutné uvést kód výkonu, lze vyplnit pouze kód výkonu uvedený v číselníku Zdravotní výkony). Číselníky jsou vydávány a aktualizovány VZP.

3.1.4 Faktura

Pro vyúčtování se rozlišují dva druhy faktur:

- faktura za období – Zdravotnické zařízení může předat fakturu za měsíční nebo čtvrtletní období. Typicky se provádí předání faktur za každý měsíc.
 - faktura za měsíc – K faktuře musí být připojeny dávky uzavřené za daný měsíc.
 - faktury za čtvrtletí – Nejprve se předají faktury za poskytnutou péči v prvních dvou měsících čtvrtletí. Tyto faktury nemusí být dodány s dárkami dokladů za obě období. Na konci čtvrtletí zdravotnické zařízení předloží fakturu za poslední měsíc čtvrtletí. Zároveň s poslední fakturou již musí být dodány všechny dávky dokladů za všechny měsíce čtvrtletí.
- faktura za dávky – Musí být vždy připojeny požadované dávky dokladů. Tento způsob fakturace se užívá například při předávání dávek od lékáren nebo při předání opravných dávek.

Při předávání dokladů elektronicky je faktura uložena v souboru FDAVKA.

3.2 Proces předávání dokladů

V následujícím textu se budu zabývat předáním dokladů pojišťovně elektronickou formou, konkrétně přes datový nosič. Předání dokladů pojišťovně na datovém nosiči je v dnešní době nejpoužívanější. Z hlediska rychlosti následného zpracování je vhodnější než papírové předávání dokladů. Lékař nebo zdravotnické zařízení může datový nosič předat smluvní pojišťovně dvěma způsoby. Buď jej předá osobně nebo zašle disketu s označením IČZ² smluvní pojišťovně. Pojišťovna po nezbytné kontrole buď provede vyúčtování a nebo informuje lékaře o neúspěšném zpracování.

²Identifikační číslo zdravotnického zařízení

3.2.1 Odmítnutí datového nosiče

Při předání datového nosiče s dávkami dokladů pojišťovně je nejprve zkontrolováno, zdali datový nosič splňuje požadavky pro předání. Pojišťovna odmítne datový nosič přijmout v následujících případech, pokud [10]:

- obsahuje počítačový vir
- nelze přečíst (např. z důvodu poškození)
- IČZ na datovém nosiči nesouhlasí s IČZ uvedeným v průvodním listu
- chybí soubor KDAVKA
- není splněno datové rozhraní
- počet dávek nesouhlasí s počtem dávek v průvodním listu
- období uvedené na průvodním listu dávky nesouhlasí s obdobím na faktuře

3.2.2 Odmítnutí dávky

Během kontroly dávky se ověřuje, že počet dokladů v dávce odpovídá počtu uvedenému na průvodním listu dávky. Dále proběhne ověření, zda dávka se stejným číslem nebyla ještě v daném roce předána a také zda zdravotnické zařízení má s pojišťovnou platnou smlouvu. Pokud nejsou tyto požadavky splněny, je celá dávka odmítnuta.

3.2.3 Odmítnutí dokladu

Příčinou odmítnutí dokladu mohou být např. následující chyby:

- doklad s uvedeným číslem byl již zdravotnickým zařízením v kalendářním roce předán
- položka nebo položky dokladu nejsou uvedeny v číselníku
- doklad je určen jiné pojišťovně

Lékař nebo zdravotnické zařízení má možnost při neúspěšné kontrole dávku nebo doklady opravit. Ve smluveném termínu pak opravenou dávku opět předá pojišťovně k vyúčtování.

3.3 Datové rozhraní číselníků

VZP vydává číselníky pro zdravotnická zařízení na datovém nosiči (disketa). Číselníky lze získat buď na pobočce VZP nebo z portálu VZP.

Název souboru s číselníkem je ve tvaru: JMENO.XXX, kde [3]:

- JMENO – Název číselníku.
- XXX – Verze číselníku.

3.3.1 Verze číselníku

Číslo verze číselníku je v rozsahu 001 až 999. Verzování číselníků popisují na následujícím příkladě. V prvním čtvrtletí roku byl vydán např. číselník JMENO.290. Verze číselníku se pro každé následující čtvrtletí zvyšuje o hodnotu 10 (tzn. ve druhém čtvrtletí je vydán číselník JMENO.300). Hodnota verze na místě jednotek slouží k doplňujícím úpravám číselníku během čtvrtletí. Pokud počet verzí číselníku během čtvrtletí je vyšší než 10, pak se zvyšuje číslo verze u všech číselníků při změně verze o hodnotu 20 (místo o 10).

3.3.2 Obsah číselníku

Soubor s číselníkem obsahuje řádky (neboli věty). V každém řádku jsou odděleně uvedeny hodnoty atributů dle datového rozhraní konkrétního číselníku. Jako oddělovač hodnot atributů ve větě je určen znak ',' (čárka). Věty jsou od sebe odděleny novým řádkem (CRLF). Každý atribut je v datovém rozhraní popsán [3]:

- názvem (zkratkou)
- typem – Typ atributu může být znakový 'C' (obsahující jeden a více znaků), numerická hodnota 'N', datum 'D' (formát: DDMMRRRR) nebo měna '\$' (formát: X.Y).
- délkou ve znacích
- popisem atributu (popis obsahu atributu)

Příklad datového rozhraní číselníku Mezinárodní klasifikace nemocí je uveden v příloze B.

3.4 Datové rozhraní dokladů

Jak již bylo zmíněno, doklady se předávají na datovém nosiči v dávkách. Datový nosič musí být označen návěstím, na kterém se uvede IČZ. Na datovém nosiči musí být umístěn jeden ASCII soubor s názvem KDAVKA.XXX, kde XXX značí kód pojišťovny. Soubor KDAVKA.XXX může obsahovat i více než jednu dávku. Například soubor s názvem KDAVKA.111 označuje soubor dávek určený pro VZP.

Při předávání dávky dokladů elektronicky na portál zdravotní pojišťovny probíhá odeslání souborů KDAVKA a FDAVKA přímo z počítače uživatele. Ověření správnosti formátu zaslaných dat proti datovému rozhraní zajišťuje příslušný portál zdravotní pojišťovny a obratem informuje uživatele o výsledku zpracování.

Dávka obsažená v souboru KDAVKA.XXX je složena [4]:

- z úvodní věty dávky – Obsahuje informace o dávce (průvodní list dávky).
- z vět dokladů – Doklad musí začínat větou záhlaví dokladu a poté následují další věty specifikující obsah dokladu (např. několik vět výkonů). Příklad datového rozhraní pro záhlaví dokladu je uveden v příloze B.

Podobně jako u číselníků jsou jednotlivé věty umístěny na samostatných řádcích. Hodnoty atributů vět se však neoddělují. V datovém rozhraní dokladů jsou atributy vět popsány stejným způsobem jako v datovém rozhraní číselníků (tzn. názvem, typem, délkou, popisem). Některé druhy dokladů je umožněno zapsat i ve formě datového rozhraní XML. Soubor s dávkou se pak označuje jako XKDAVKA.XXX.

Obsah souboru s dávkou (i obsah souboru s fakturou) lékař přímo nevytváří, pouze zadává data přes uživatelské rozhraní lékařského softwaru. Lékařský software poté ze zadaných dat vytváří věty dle datového rozhraní a ukládá je do příslušných souborů. Podle typu lékařského softwaru může lékař vytvořené soubory vložit na datový nosič nebo přímo odeslat na portál zdravotní pojišťovny. Dalším způsobem je ruční vložení souborů na portál přes jeho webové rozhraní, pokud to portál umožňuje. Příklad obsahu souboru KDAVKA lze vidět v příloze B.

Kapitola 4

Portál zdravotních pojišťoven

V následující kapitole se nejprve budu zabývat základním popisem Portálu zdravotních pojišťoven. Dále se zaměřím na možnosti přístupu na portál a v závěru kapitoly se věnuji některým službám portálu.

4.1 Portál ZP

Portál zdravotních pojišťoven¹ (Portál ZP) nabízí možnost pohodlně komunikovat s několika zdravotními pojišťovnami. Portál mohou využívat zdravotnická zařízení, zaměstnavatelé i pojištěnci příslušné pojišťovny. Komunikovat lze celkem s pěti zdravotními pojišťovnami [8]:

- Česká průmyslová zdravotní pojišťovna
- Oborová zdravotní pojišťovna zaměstnanců bank, pojišťoven a stavebnictví
- Revírní bratrská pokladna-zdravotní pojišťovna
- Vojenská zdravotní pojišťovna ČR
- Zaměstnanecká pojišťovna Škoda

Uživatelé portálu nemusí provádět často zdlouhavou papírovou administrativu a posílat doklady či dokumenty poštou. Jedná se tedy o rychlý, pohodlný a snadný způsob komunikace s pojišťovnou. Portál ZP je webová aplikace a jeho služby jsou k dispozici uživatelům bezplatně. Pro každou z výše uvedených pojišťoven je vytvořeno jedno webové rozhraní odlišující se pouze grafickým stylem pojišťovny. V příloze C lze vidět příklad webového rozhraní portálu, konkrétně portálu Vojenské zdravotní pojišťovny ČR. Na portál konkrétní pojišťovny uživatelé vstupují přímo z oficiálních webových stránek nebo ze společné zóny Portálu ZP. Veškerá komunikace při používání portálu je zabezpečena.

VZP využívá k elektronické komunikaci se smluvními zdravotnickými zařízeními, pojištěnci a dalšími klienty vlastní portál - Portál VZP². Portál VZP nabízí podobné služby a funkce jako Portál ZP.

¹<http://www.portalzp.cz/>

²<http://www.vzp.cz/e-komunikace/portal-vzp>

4.2 Registrace na Portál ZP

Pro registraci na Portál ZP vyplňuje budoucí uživatel registrační formulář³. Před vyplněním registračního formuláře se uživatel musí rozhodnout, jaký typ autentizace chce v budoucnu využívat. Podle typu zvolené autentizace se pak liší postup při registraci na Portál ZP. Na výběr je buď autentizace pomocí osobního digitálního certifikátu a nebo autentizace pomocí SMS⁴. Uživatel Portálu ZP může mít aktivovány obě možnosti autentizace a v případě potřeby je kombinovat [5].

Po úspěšné registraci je vytvořeno uživateli portálové konto, jehož prostřednictvím bude komunikovat se zdravotními pojišťovnami. Registrací na Portál ZP získává uživatel možnost komunikovat se všemi pojišťovnami spolupracujícími s portálem. Stačí jen v registračním formuláři zvolit ty pojišťovny, ke kterým uživatel bude žádat přístup. Není tedy nutné žádat každou pojišťovnu o komunikaci v samostatném formuláři. Pro bezproblémovou práci s Portálem ZP je uživatelům doporučeno, bez ohledu na typ autentizace, nainstalovat do webového prohlížeče kořenové certifikáty I.CA⁵. Nyní se podrobněji zaměřím na již zmíněné možnosti autentizace na Portál ZP.

4.2.1 Přístup na Portál ZP přes osobní digitální certifikát

K přístupu na Portál ZP přes osobní certifikát je nutné před vyplněním registračního formuláře postupovat podle pokynů průvodce⁶. Uživatel musí mít před samotnou registrací k dispozici svůj osobní certifikát. Certifikát si pak importuje do svého webového prohlížeče, pokud jej již neimportoval dříve. Portál ZP uznává osobní certifikáty vydané jen některými certifikačními autoritami. K přístupu na portál lze použít osobní certifikáty [8]:

- I.CA
- Komerční banky
- České pošty
- eIdentity

Při komunikaci s portálem přes webový prohlížeč je využíván podpisový modul, který zajišťuje podpis dat osobním certifikátem uživatele. Webový prohlížeč MS Internet Explorer využívá podpisový modul založený na ActiveX⁷. Pro ostatní webové prohlížeče lze nastavit podpisový modul založený na platformě Java. Instalaci podpisového modulu nabídne prohlížeč buď sám, nebo uživatel provede jeho instalaci v průvodci. Až nyní může budoucí uživatel portálu přejít k vyplnění registračního formuláře. Na registračním formuláři musí uživatel povinně vyplnit své jméno, příjmení a email. Dále je nutné zvolit druh přístupu na portál, kde lékaři zvolí možnost Lékař nebo zdravotnické zařízení. Po výběru zdravotní pojišťovny, se kterou chce uživatel komunikovat, je nutné vyplnit, v případě lékaře, identifikační údaje zdravotnického zařízení, za které se budou předkládat doklady k vyúčtování. Jedná se o IČ⁸, číslo organizační jednotky a IČZ⁹. Po vyplnění registračního formuláře uživatel svou žádost ihned podepisuje svým osobním certifikátem a následně je jeho portálové

³<https://www.portalzp.cz/czspa01.phtml>

⁴Short message service

⁵První certifikační autorita, a.s.

⁶<https://www.portalzp.cz/czspa00.phtml>

⁷ActiveX je softwarový framework společnosti Microsoft

⁸Identifikační číslo osoby

⁹Identifikační číslo zdravotnického zařízení

konto aktivováno. Pro přihlašování ke svému portálovému kontu bude uživatel používat stejný osobní certifikát. Kromě přihlašování je osobní certifikát nutný i pro využití některých služeb portálu. Jedná se především o služby poskytované zdravotnickým zařízením, o kterých se zmíním v poslední kapitole o Portálu ZP.

Uživatel může později rozšířit svá přístupová oprávnění na portál. Pokud bude chtít aktivovat například přístup za zaměstnavatele nebo pojištěnce, provede na stejném formuláři jako při prvotní registraci výběr nového druhu přístupu. Následně podepíše žádost o rozšíření přístupových práv certifikátem, který má již na portálu zaveden a odešle.

4.2.2 Přístup na Portál ZP přes SMS

V případě, že uživatel zvolil možnost SMS autentizace, vyplní registrační formulář a odešle žádost. Následně se musí dostavit na pobočku libovolné pojišťovny spolupracující s portálem. S sebou musí mít mobilní telefon a průkaz totožnosti. Po uzavření smlouvy o SMS autentizaci mu bude poté aktivováno pracovníkem pojišťovny portálové konto. Přihlašování na Portál ZP pak uživatel provádí pokaždé zadáním jednorázového kódu z přijaté SMS zprávy. Výhodou SMS autentizace je možnost přistoupit na Portál ZP z jakéhokoli počítače. Naproti tomu při autentizaci osobním certifikátem je uživatel omezen možností přihlášení pouze z počítače, na němž je certifikát nainstalován.

4.3 Služby Portálu ZP

Hlavním účelem Portálu ZP je jednotná komunikace s pojišťovnami. Kromě toho Portál ZP usnadňuje vykonávání nutné administrativy spojené s komunikací mezi klientem a pojišťovnou. Portál ZP nabízí za tímto účelem mnoho funkcí pro každou ze skupin uživatelů portálu. Některé služby jsou společné pro všechny skupiny uživatelů. Především se budu nadále zabývat možnostmi Portálu ZP pro zdravotnická zařízení a lékaře.

4.3.1 Společné služby Portálu ZP

Všichni uživatelé mají po zaregistrování k dispozici Osobní schránku, funkci Ověření pojištěnce a Ověření zdravotnického zařízení [1]. Osobní schránka slouží pro příjem informací nebo souborů od pojišťovny. Přes schránku mohou uživatelé komunikovat nejen s pojišťovnou, ale i mezi sebou. Funkce Ověření pojištěnce umožňuje uživateli ověřit, u jaké pojišťovny je pojištěn. Stačí zadat číslo pojištěnce (rodné číslo) nebo číslo průkazu EHIC¹⁰. K ověření ZZ¹¹ je potřeba zadat IČZ nebo IČ spolu s číslem organizační jednotky. Poté je zobrazena informace s kontaktními údaji o ZZ a zdali je dané ZZ s pojišťovnou smluvní.

Užitečnou funkcí portálu je elektronická podatelna. Umožňuje zaslat pojišťovně zprávu, ať už se jedná o obecný dotaz, žádost nebo reklamaci spolu s přílohou. Elektronickou podatelnu a ověření pojištěnce nebo ZZ mohou využít i uživatelé, kteří na portálu nejsou registrováni.

4.3.2 Služby pro zdravotnická zařízení a lékaře

V kapitole o předávání dokladů zdravotní pojišťovně jsem se zmínil o předávání dokladů na datovém nosiči, což je značně nepohodlné. Portál ZP nabízí, jako jednu ze služeb pro

¹⁰European Health Insurance Card

¹¹Zdravotnické zařízení

lékaře, předat nezbytné soubory přes webové rozhraní portálu. Služba Vyučtování zdravotní péče poskytuje několik způsobů předání souborů pro vyúčtování pojišťovně, přitom v některých případech není nutné mít potřebné soubory předem vytvořeny.

V případě, že má lékař k dispozici soubory FDAVKA a KDAVKA, jednoduše oba dva soubory vloží přes webové rozhraní a odešle na portál. Nebo lze vložit jen soubor KDAVKA a soubor FDAVKA vytvořit pomocí webového rozhraní a následně provést odeslání. Vytvoření souboru FDAVKA přes webové rozhraní probíhá na základě dat vyplněných do formuláře. Další možností je vložit pouze soubor FDAVKA a ten odeslat. Poslední možnou variantou je opět odeslání souboru FDAVKA, avšak vytvořeného přes formulář.

Před odesláním souboru FDAVKA (včetně případu vytvoření přes formulář) a KDAVKA se provádí podepsání těchto souborů certifikátem. Uživatel má možnost provést výběr certifikátu pro podpis a pak odsouhlasit odeslání. Zároveň se před odesláním provádí kontrola předaných souborů a v případě nesrovnalostí je uživatel upozorněn a může příslušné soubory opravit.

Kapitola 5

Elektronické rozhraní Portálu ZP

Přestože je předávání dokladů pro pojišťovnu přes webové rozhraní pohodlné, lze pro tento účel využít také lékařský program. Většina lékařských programů umožňuje odeslat soubory s dávkami dokladů přímo na některou z pojišťoven. Portál ZP dává k dispozici dokumentaci elektronického rozhraní i demo aplikaci. Vývojáři softwaru tak mohou buď využít připravených knihoven pro implementaci komunikace s Portálem ZP a nebo provést vlastní implementaci komunikace dle dokumentace elektronického rozhraní.

5.1 Komunikační brána

S každou pojišťovnou na Portálu ZP probíhá elektronická komunikace přes komunikační bránu. Komunikační brány jednotlivých pojišťoven jsou dostupné na následujících adresách [9]:

- Společná brána pro všechny pojišťovny: https://www.portalzp.cz/kom_brana.phtml
- ČPZP – https://portalz.cpzp.cz/kom_brana.phtml
- OZP – https://portal.ozp.cz/kom_brana.phtml
- RBP – https://portal.rbp-zp.cz/kom_brana.phtml
- VoZP ČR – https://portal.vozp.cz/kom_brana.phtml
- ZPŠ – https://portal.zpskoda.cz/kom_brana.phtml

Výše uvedené adresy komunikačních bran slouží k plnohodnotnému zpracování požadavků zasílaného klientským programem. Nejsou vhodné pro testovací účely a ladění. Pro testování komunikace jsou vyhrazeny testovací komunikační brány:

- Společná brána: https://pilot-brn-pzp.asseco.cz/kom_brana.phtml
- https://pilot-pzp.asseco.cz/kom_brana.phtml

Požadavky zaslané na testovací brány se žádnou pojišťovnou nezpracovávají. Vývojáři softwaru brány využívají k ověřování správnosti implementace elektronického rozhraní Portálu ZP ve svých programech.

5.2 Princip komunikace

Elektronické rozhraní umožňuje komunikaci s Portálem ZP s použitím certifikátu a nebo s využitím SMS zpráv. Konkrétně se jedná o to, zdali se pro ověření identity odesílatele

ověřuje podpis dat nebo jméno, heslo a jednorázový SMS kód. V této práci se zabývám předáváním dokladů pro vyúčtování zdravotní péče na Portál ZP s využitím certifikátu.

Komunikace mezi programem a komunikační bránou probíhá přes protokol HTTPS. Uživatel svůj osobní certifikát pro navazování HTTPS spojení nepoužívá. Osobní certifikát uživatele slouží k podepisování odesílaných dat a komunikační brána certifikátem uživatele ověřuje tento podpis po přijetí dat. Software ze kterého proběhlo odeslání dat musí mít k dispozici certifikát příslušné brány k ověření podpisu dat v odpovědi zaslané bránou [2].

Certifikát každé brány je k dispozici na adrese: `https://[portal]/portal_sign.pem`. Kde [portal] značí adresu portálu pojišťovny. Například certifikát pro ověření podpisu dat zaslaných z portálu OZP¹ je dostupný na adrese `https://portal.ozp.cz/portal_sign.pem`. Jakmile je HTTPS spojení s komunikační bránou úspěšně navázáno, komunikující software musí sestavit zprávu pro odeslání dat. Zprávy zasílané mezi programem a bránou mají podobu XML dokumentu. Vytvořený XML dokument program zašle metodou HTTP POST komunikační bráně, která po jeho zpracování odpoví opět XML dokumentem. Zpráva zaslaná bránou jako odpověď informuje o úspěchu či neúspěchu zpracování daného požadavku. Elektronické rozhraní umožňuje několik způsobů odeslání XML dokumentu na komunikační bránu:

1. v těle HTTP POST požadavku se předá parametr `request`, jehož hodnotou je přímo XML dokument
2. hodnotou parametru `request` může být komprimovaný soubor (ZIP nebo GZIP) obsahující XML dokument
3. hodnotou parametru `request` jsou komprimovaná data (v kódování Base64²), následuje parametr `request_compression`, jehož hodnotou je typ komprese (zip nebo gz)

5.3 Struktura XML zprávy

Dokumentace elektronického rozhraní Portálu ZP specifikuje pravidla a formát pro tvorbu XML zpráv. Kořenovým elementem XML zprávy je element *Komunikace*. Element *Komunikace* obsahuje dále elementy *Data* a *Podpis*.

5.3.1 Element *Data*

V elementu *Data* se vyskytují tyto atributy:

- **Typ** – Určuje směr komunikace. Hodnota POZADAVEK - při předání požadavku komunikační bráně. Hodnota ODPOVED - při odpovědi komunikační brány na přijatý požadavek.
- **Ucel** – Udává účel předávaného požadavku. Pro předání vyúčtování zdravotní péče nabývá atribut hodnoty VYU. Dokumentace elektronického rozhraní definuje i další účely předání požadavku např. ORL - Předání registračních lístků, VERZZ - Ověření zdravotnického zařízení a další. Jinými účely komunikace s Portálem ZP se však nezabývám.
- **PZP_IdPodani** – Nachází se pouze v odpovědi portálu. Hodnotou je identifikační číslo přijatého požadavku.

¹Oborová zdravotní pojišťovna

²datový formát zobrazující binární data pomocí tisknutelných znaků ASCII

- **PZP_Chyba** – Pouze v odpovědi portálu. Hodnota 0 značí úspěšné přijetí požadavku a tento požadavek bude předán zdravotní pojišťovně. V opačném případě je hodnotou kód chyby a chybové hlášení je obsahem elementu *Soubor*.

Uvnitř elementu *Data* se dále musí vyskytovat elementy:

- *Prihlaseni* – Uvádí se pouze při odesílání požadavku na komunikační bránu. Obsahuje atribut **UnikatniInfo** k zajištění jedinečnosti zasílaného požadavku. Hodnotou atributu **UnikatniInfo** může být například pořadové číslo požadavku nebo nejlépe aktuální datum a čas odeslání požadavku. Element *Prihlaseni* předává přihlašovací text.
- *Soubor* – Uplatní se při obou směrech komunikace. Musí být obsažen alespoň jednou. Slouží pro předání vlastních dat (např. obsah souboru KDAVKA). Obsahuje atribut **Jmeno** udávající název souboru a atribut **Format** pro určení formátu dat (hodnota TEXT nebo BASE64).

5.3.2 Element *Podpis*

V elementu *Podpis* se nachází elektronický podpis obsahu elementu *Data*. Do podpisu se zahrnuje i počáteční a koncový tag elementu *Data*. Při komunikaci probíhá před odesláním XML zprávy vytvoření podpisu pomocí certifikátu. Při přijetí XML zprávy je provedeno ověření podpisu opět prostřednictvím certifikátu. Podpis je vytvářen na základě PKCS#7³.

5.4 XML komunikace pro vyúčtování zdravotní péče

V následujících podkapitolách budou podrobněji popsány XML zprávy zajišťující komunikaci při vyúčtování zdravotní péče.

5.4.1 XML zpráva požadavku

Na ukázce XML požadavku 5.1 lze vidět jeden ze způsobů předání vyúčtování zdravotní péče pojišťovně na Portálu ZP. Unikátnost požadavku je v tomto případě zajištěna pořadovým číslem komunikace (1234567). V přihlašovacím textu se doplní za [zkratka ZP] zkratka zdravotní pojišťovny, zde to bude ČPZP (205 - Česká průmyslová zdravotní pojišťovna). Při použití testovací brány se zadává zkratka PILOT. Za [navez služby] se dosadí text 'vyúčtování zdravotní péče'.

Příklad 5.1 ilustruje předání obsahu souboru faktury FDAVKA.205 spolu s obsahem souboru dávky KDAVKA.205. Soubor s fakturou se musí předat vždy. Obsahem elementu *Soubor* jsou věty z daných souborů.

³Public Key Cryptographic Standards #7 - Cryptographic Message Syntax Standard

```

<?xml version="1.0" encoding="ISO-8859-2"?>
<!DOCTYPE Komunikace>
<Komunikace>
  <Data Typ="POZADAVEK" Ucel="VYU">
    <Prihlaseni UnikatniInfo="1234567">
      Timto se prihlasuji k Portalu [zkratka ZP]. V ramci tohoto prihlaseni
      predavam podani [nazev sluzby] v definovanem datovem rozhrani.
      Po predani tohoto podani a prijmu odpovedi Portalu [zkratka ZP] se
      z Portalu [zkratka ZP] odhlasuji.
    </Prihlaseni>
    <Soubor Jmeno="FDAVKA.205" Format="TEXT">
      data faktury
    </Soubor>
    <Soubor Jmeno="KDAVKA.205" Format="TEXT">
      data davky
    </Soubor>
  </Data>
  <Podpis>podpis PKCS7</Podpis>
</Komunikace>

```

Ukázka kódu 5.1: Příklad XML výzvy

5.4.2 XML zpráva odpovědi - úspěšné zpracování

Po úspěšném zpracování XML požadavku Portálem ZP přijímá komunikující program XML odpověď ve tvaru 5.2. Kladnou odpověď portálu indikuje hodnota 0 atributu PZP_Chyba. Portál ZP vrací v odpovědi protokol o zpracování, v tomto případě ve formátu BASE64.

```

<?xml version="1.0" encoding="ISO-8859-2"?>
<!DOCTYPE Komunikace>
<Komunikace>
  <Data Typ="ODPOVED" Ucel="VYU" PZP_IdPodani="123" PZP_Chyba="0">
    <Soubor Jmeno="Protokol.txt" Format="BASE64">
      protokol o zpracovani
    </Soubor>
  </Data>
  <Podpis>podpis PKCS7</Podpis>
</Komunikace>

```

Ukázka kódu 5.2: Příklad XML kladné odpovědi

5.4.3 XML zpráva odpovědi - neúspěšné zpracování

Odpověď Portálu ZP při neúspěšném zpracování XML požadavku ukazuje příklad 5.3. Popis chyby Portál ZP předává v chybovém souboru Error.log. Při zpracování XML požadavku se lze setkat s chybami jako [2]:

- nevalidní podpis
- nezaregistrovaný certifikát
- chybné heslo při SMS přihlášení
- SMS přihlašování není aktivní
- další chyby např. při kontrole souborů FDAVKA a KDAVKA


```

<?xml version="1.0" encoding="ISO-8859-2"?>
<!DOCTYPE Komunikace>
<Komunikace>
  <Data Typ="ODPOVED" Ucel="VYU" PZP_IdPodani="123" PZP_Chyba="666">
    <Soubor Jmeno="Error.log" Format="BASE64">
      chybove hlaseeni
    </Soubor>
  </Data>
  <Podpis>podpis PKCS7</Podpis>
</Komunikace>

```

Ukázka kódu 5.3: Příklad XML záporné odpovědi

5.5 Demo aplikace knihovny ASSECOPZP

Portál ZP poskytuje vývojářům lékařského softwaru knihovnu ASSECOPZP. Knihovna ASSECOPZP je určena pro implementaci komunikace mezi lékařským softwarem a Portálem ZP. Tato knihovna je použitelná v jakémkoliv vývojovém nástroji, který podporuje technologii COM⁴ (např. Microsoft Visual C++, Active Server Pages, Delphi).

Společně s knihovnou ASSECOPZP distribuuje Portál ZP demo aplikaci demonstrující komunikaci knihovny ASSECOPZP s Portálem ZP. Demo aplikace, jejíž hlavní okno je znázorněno v příloze C umožňuje komunikovat s portály pojišťoven i testovacími portály.

Lze komunikovat všemi druhy účelů komunikace tak, jak jsou popsány v dokumentaci elektronického rozhraní. Nechybí ani možnost výběru způsobu ověření identity, buď osobním certifikátem, nebo SMS kódem. Knihovna ASSECOPZP využívá ActiveX komponentu pro podepisování a ověřování zasílaných dat mezi demo aplikací a Portálem ZP stejným způsobem jako webové rozhraní Portálu ZP.

⁴Component Object Model

Kapitola 6

Implementace rozšíření do aplikace OpenDositech a testování

V první podkapitole následujícího textu bude čtenář ve stručnosti seznámen s původní verzí programu OpenDositech. Ve zbylých podkapitolách se budu zabývat popisem implementace všech úprav, které jsem v rámci této práce do programu OpenDositech zakomponoval. Hlavní úpravou původní verze programu je rozšíření realizující zaslání dávek na Portál ZP. Dále byly provedeny úpravy některých stávajících funkcí programu, jejichž cílem bylo vytvořit přehlednější uživatelské rozhraní.

6.1 Základní program OpenDositech

V následující podkapitole se budu věnovat popisu funkcí nabízených základní verzí OpenDositech a některým třídám jeho základní implementace.

6.1.1 Práce v programu OpenDositech

Při prvním spuštění programu je uživatel vyzván k zadání IČZ. Všechny dávky vytvořené v programu jsou pak předkládány pojišťovně za zdravotnické zařízení, jehož IČZ uživatel zadal. Po zadání IČZ je zobrazeno hlavní okno programu, kde v dolní části je zobrazeno IČZ a v horní části se nachází následující menu:

- Účty
 - Vkládání dat – Umožňuje vytvořit novou dávku dokladů nebo přidat doklady do existující dávky. Nejprve je nutné zadat číslo dávky a kód pojišťovny. Poté je nabídnuta možnost vytvořit dva typy dávek: Dávka Ambulantní smíšená (typ 98) a Dávka Hospitalizační smíšená (typ 99). Následně uživatel vyplňuje potřebné doklady.
- Dávky
 - Vytvoření dávky – Připraví dříve vytvořenou dávku dokladů pro předložení pojišťovně k vyúčtování.
 - Přehled dávek – Zobrazí přehled vytvořených dávek, kde lze procházet obsah dávek a vytisknout průvodní list dávky.
 - Načtení dávky – Načte vybraný soubor s dávkou do programu.

- Nastavení
 - Změna data

Standardní postup uživatele při práci s programem je vytvoření dávky přes *Účty > Vkládání dat*. Po vytvoření dokladu a vyplnění jednotlivých výkonů je dávka zařazena do přehledu dávek v *Dávky > Přehled*. Jakmile uživatel potřebuje dávku předložit pojišťovně provede tak přes *Dávky > Vytvoření dávky*. Výsledná dávka (tzn. k obsahu dávky je přidán navíc průvodní list dávky) je jako jeden soubor uložen na zvolené místo.

6.1.2 Základní implementace

Program OpenDositech je napsán v programovacím jazyce C++ s využitím Qt frameworku. Qt framework umožňuje efektivní vývoj multiplatformních aplikací s grafickým uživatelským rozhraním. OpenDositech je vyvíjen pro operační systém Linux a Windows. K ukládání vytvořených dat uživatelem využívá databázi SQLite. Dále ke své činnosti potřebuje několik následujících číselníků VZP uložených ve složce *cis* programu OpenDositech:

- Zdravotní pojišťovny
- Územní pracoviště VZP
- Mezinárodní klasifikace nemocí
- Zdravotní výkony

Při spuštění programu je vytvořen adresář *archiv*, pokud zatím nebyl vytvořen. Adresář *archiv* slouží k uložení jen těch dávek, u kterých to uživatel požaduje. Pak se vytvoří nová nebo otevře již existující SQLite databáze. Při tvorbě nové SQLite databáze se vytvoří následující tabulky:

- *seznam_davek* – Obsahuje všechny dávky vytvořené programem.
- *doklady* – Obsahuje všechny vytvořené doklady.
- *vykony* – Obsahuje všechny zaznamenané výkony.

K databázi se poté připojí všechny číselníky ze složky *cis*. Program uloží IČZ zadané uživatelem při prvním spuštění do konfiguračního souboru *config.xml*. Při každém dalším spuštění programu se IČZ načítá z konfiguračního souboru. Jestliže soubor *config.xml* není nalezen, vytvoří se nový.

V následujícím seznamu bude čtenář seznámen s vybranými třídami základní implementace programu OpenDositech, mezi ně patří:

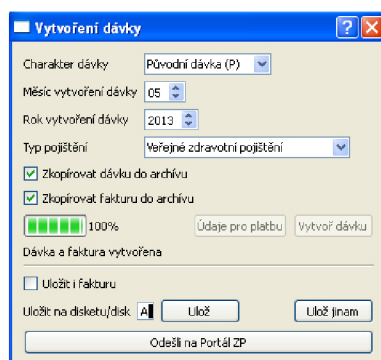
- *DB* – Vytváří a spravuje SQLite databázi. Obsahuje implementace metod pro přidání, aktualizaci, odstraňování záznamů o dávkách, dokladech a výkonech. Zajišťuje také zpětné získávání informací o uložených záznamech. Inicializace databáze provádí metoda *init()*. Připojení číselníků VZP ze složky *cis* realizuje metoda *attachDatabases()*.
- *XmlHandler* – Implementuje získání hodnot z konfiguračního souboru programu.
- *ICZdialog* – Obsahuje implementaci okna pro zadání IČZ, které se zobrazuje při prvotním spuštění programu nebo při nenalezení konfiguračního souboru.
- *MainWindow* – Implementuje hlavní okno programu, menu a rozhraní pro doklady. Menu programu je vytvořeno metodami *createMenu()* a *createActions()*. Implementaci uživatelského rozhraní pro zadávání dokladů zajišťuje metoda *createNewBatch()*.

- **InsertDialog** – Vytváří okno pro zadání čísla dávky a kódu pojišťovny při tvorbě nové dávky nebo tvorbě dávky pro pojišťovnu.
- **BatchInfo** – Implementuje widget pro zobrazení informací o dávce při zadávání dokladu.
- **RecordInfo** – Implementuje widget pro zobrazení informací o dokladu při jeho zadávání.
- **Record** – Implementuje widget pro zadávání výkonů v dokladu.
- **CreateBatchDialog** – Realizuje tvorbu okna pro vytvoření dávky pro pojišťovnu. Metoda *createBatch()* implementuje tvorbu dávky. Metody *createPruvodniListDavky()* provádí tvorbu věty průvodního listu dávky. Metody *createVetaZahlavi()* a *createVetaVykony()* pak zařídí tvorbu věty záhlaví dokladu a jedné nebo více vět výkonů v rámci dokladu.
- **ViewBatchDialog** – Implementuje okno s přehledem dávek uložených v databázi. Tisk průvodního listu dávky realizuje metoda *printPruvodniList()*. Zobrazení okna s detailními informacemi o obsahu dávky obstarává metoda *showBatchDetail()*.
- **LoadBatch** – Tvoří okno pro načtení dávky z libovolného umístění.
- **DateDialog** – Zobrazuje okno pro změnu data.
- **Message** – Obsahuje implementaci okna pro různé typy hlášení programu uživateli.

6.2 Předávání dávek dokladů na Portál ZP

V následujících podkapitolách se zaměřím na podrobný popis implementace rozšíření a dalších úprav, které jsem v rámci této bakalářské práce prováděl. Při úpravách základní implementace OpenDositechu jsem využíval svn¹, kde je k dispozici i nejnovější MSI instalace programu. Nejprve se budu věnovat popisu nejvýznamnější úpravy programu OpenDositech, a to tvorbě rozšíření pro předávání dokladů na Portál ZP.

Implementace rozhraní pro komunikaci s Portálem ZP byla hlavním cílem této práce. Po nastudování dokumentace elektronického rozhraní Portálu ZP přišlo na řadu rozhodnutí, kam do původního OpenDositechu zakomponovat rozšíření. K tomuto účelu jsem zvolil původní okno pro vytvoření dávky pro pojišťovnu (*Dávky > Vytvoření dávky*), kde jsem přidal možnost odeslání dávky na Portál ZP.



Obrázek 6.1: Upravené okno pro vytvoření dávky pro pojišťovnu

¹<http://code.google.com/p/opensitech/>

V základní verzi OpenDositechu nebyla implementována tvorba faktury za dávku. Do zmíněného okna byly proto přidány možnosti uložení faktury do archivu a zadání údajů platby pro fakturu. Fakturu lze stejně jako dávku uložit buď na disketu, nebo na jiné umístění. Výslednou podobu okna pro vytvoření dávky pojišťovně znázorňuje obrázek 6.1. Implementace tvorby faktury vyžadovala úpravu původní třídy `CreateBatchDialog`. Tvorbu faktury nově zajišťují metody `createVetaZahlaviVyuctovani()`, `createVetaZadostO-Vyuctovani()` a `createVetaDavkaProVyuctovani()`. V původní metodě `createBatch()` je pak získána kompletní faktura pro dávku voláním metody `createFaktura()`. Zde je také umístěna implementace uložení faktury do archivu. Pro uložení faktury na disketu a na jiné umístění byla provedena jednoduchá úprava metod `saveBatch()` a `saveDialog()`.

6.2.1 Metody `sendPortal()` a `createXmlRequest()`

Než přistoupím k popisu nových tříd programu OpenDositech, vysvětlím funkci zbývajících dvou metod, které jsem implementoval v původní třídě `CreateBatchDialog`.

Jakmile uživatel klikne na tlačítko Odešli na Portál ZP v okně pro vytvoření dávky 6.1, zavolá se metoda `sendPortal()`. Metoda `sendPortal()` otevře soubor s dávkou vytvořený dříve v metodě `createBatch()` a uloží jeho obsah do globální proměnné `m_batch_string`. Poté dojde k volání metody `createXmlRequest()`. Metoda `createXmlRequest()` sestavuje datovou část XML zprávy pro portál (tzn. obsah elementu `Data`), dle dokumentace elektronického rozhraní Portálu ZP. Sestavená datová část XML zprávy je uložena do globální proměnné `m_xmlRequest`. Nakonec je vytvořena instance třídy `SslClient`. Konstruktore této třídy je předána proměnná `m_xmlRequest` jako jediný parametr. Následně se zobrazí okno klienta.

Implementaci rozhraní pro komunikaci s Portálem ZP zajišťují třídy:

- `SslClient` – Implementuje klienta pro zajištění SSL² komunikace s portálem a další potřebná nastavení (výběr portálu, výběr certifikátu).
- `XmlSigner` – Slouží k tvorbě podpisu odesílaných dat klientem a ověřování podpisu přijatých dat z portálu.
- `CertInfo` – Spravuje certifikáty pro podpis. Využívá pomocnou třídu `CertInfoModel`.
- `XmlMessageParser` – Implementuje parser pro analýzu přijatých XML dat z portálu.
- `AccountDialog` – Nastavuje platební údaje při tvorbě faktury za dávku.

6.2.2 Třída `SslClient`

Základním problémem během implementace rozšíření bylo, jak realizovat zabezpečenou komunikaci s portálem. Jak již bylo řečeno v kapitole o elektronickém rozhraní Portálu ZP, je nutné vytvářet POST požadavek pro portál. Qt framework nabízí pro implementaci bezpečné síťové komunikace několik řešení v podobě tříd:

- `QSslSocket` – Umožňuje vytvořit SSL socket pro přenos a příjem šifrovaných dat. Do socketu se zapisují konkrétní data a je potřeba např. při tvorbě POST požadavku pro server přesně vědět, jakým způsobem tyto data sestavit, což může být někdy obtížné.
- `QHttp` – Již zastaralá třída `QHttp` poskytuje přímo metody pro tvorbu POST a GET požadavků. Pro zajištění HTTPS komunikace je potřeba navíc zvlášť nastavit SSL socket.

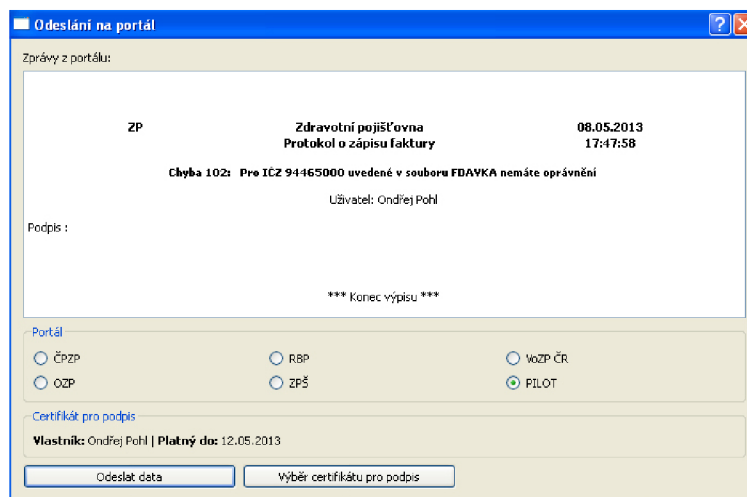
²Secure Sockets Layer

- **QNetworkAccessManager** – Poskytuje ideální řešení pro implementaci zabezpečené komunikace a tuto třídu využívám ve svém rozšíření. Umožňuje aplikaci odesílat požadavky po síti a přijímat odpovědi na tyto požadavky.

Nastavení klienta před odesláním dat

V konstruktoru třídy `SslClient` je vytvořen objekt třídy `QNetworkAccessManager` do globální proměnné `m_netMgr`. Zpracování všech přijatých dat je následně přiřazeno metodě `receiveData()`. Do globální proměnné `m_xml` je uložena datová část XML zprávy pro portál. Nakonec je provedena inicializace okna zobrazujícího uživatelské rozhraní klienta pro komunikaci s Portálem ZP.

V okně klienta znázorněného obrázkem 6.2 provádí uživatel odeslání dat na Portál ZP. Na obrázku 6.2 lze vidět konkrétní případ komunikace OpenDositechu s portálem. Jedná se o komunikaci s testovacím portálem PILOT, kde portál hlásí ve zprávě uživateli, že nemá na portálu nastaveno oprávnění za zdravotnické zařízení s daným IČZ. Klient zobrazuje textové pole, které je určené pro přijaté zprávy z portálu. Pod ním se nachází výběr portálu ke komunikaci, informace o použitém certifikátu, možnost odeslání dat a možnost výběru certifikátu k podpisu. Uživatel nemusí vybírat certifikát k podpisu, pokud má nastavený výchozí certifikát v souboru `cert/user-cert-list`. Výchozí certifikát, pokud existuje, se načte v klientovi automaticky při jeho zobrazení. Více o výchozích certifikátech a obecně o výběru certifikátu pro podpis pojednává podkapitola popisující třídu `CertInfo`. Před odesláním dat musí uživatel vybrat portál, se kterým chce komunikovat a certifikát, kterým bude data podepisovat. Metoda `setPortal()` zajistí uložení URL adresy po výběru portálu do globální proměnné `m_portalToConnect`. Po kliknutí na tlačítko Odeslat data je zavolána metoda `sendData()`.



Obrázek 6.2: Okno klienta pro komunikaci s Portálem ZP

Odeslání dat na Portál ZP

Metoda `sendData()` nejprve zobrazí okno vyzívající uživatele pro zadání hesla k privátnímu klíči certifikátu. Proměnná `signer` představující podepisovač dat, uchovává instanci třídy `XmlSigner`. Zkonstruování podpisu dat zajistí metoda podepisovače `signXML()`, které

je předána datová část XML zprávy, cesta k certifikátu pro podpis a zadané heslo k privátnímu klíči. Získaný podpis je pak v proměnné `signature`. Sestavení výsledné XML zprávy pro portál ilustruje úryvek kódu 6.1.

```
// data zasilana na portal
QByteArray data;

// vytvoreni obsahu POST parametru "request"
// sestaveni XML zpravy dle pozadavku v dokumentaci portálu
data.clear();
data.append("<?xml version=\"1.0\" encoding=\"ISO-8859-2\"?>\n"
           "<!DOCTYPE Komunikace>\n"
           "<Komunikace>\n"
           + m_xml +
           "\n<Podpis>\n"
           + signature +
           "</Podpis>\n"
           "</Komunikace>");
```

Ukázka kódu 6.1: Sestavení XML zprávy pro portál

Před zasláním POST požadavku síťovým manažerem `m_netMgr` je vytvořen tzv. síťový požadavek `QNetworkRequest` reprezentovaný proměnnou `dataRequest`. Síťovému požadavku se nastaví SSL konfigurace s příznakem `QSSLocket::VerifyNone`. Tím je zajištěno, že při vytváření SSL spojení s portálem nebude vyžadován certifikát jak je popsáno v dokumentaci elektronického rozhraní Portálu ZP. Dokumentace stanovuje, že autentizace probíhá pouze na základě dohledání certifikátu použitého pro podpis. Dále je požadavku přiřazena URL adresa portálu. Nakonec se pro síťový požadavek nastaví hodnota parametru Content-Type na MIME³ typ `application/x-www-form-urlencoded`. Tímto bylo provedeno nastavení parametru Content-Type přímo v hlavičce HTTP POST požadavku. Pro odeslání dat je zavolána metoda `post()` síťového manažera `m_netMgr`. Metodě `post()` je v prvním parametru předán síťový požadavek. V druhém parametru se předávají samotná data. Dle dokumentace elektronického rozhraní Portálu ZP je předáván parametr s názvem `request`, jehož hodnotou je sestavená XML zpráva.

K ověření podpisu dat, zasláných portálem jako odpověď, je potřeba mít k dispozici certifikát konkrétního portálu. K jeho získání používám opět síťového manažera, který zajistí zaslání HTTP GET požadavku na příslušnou URL adresu, kde se certifikát nachází. Síťový manažer poskytuje pro tento účel metodu `get()`. Podobně jako při zaslání POST požadavku, je i zde vytvořen síťový požadavek. Síťovému požadavku `portalCertRequest` třídy `QNetworkRequest` je nastavena totožná SSL konfigurace jako požadavku `dataRequest` a dále přiřazena URL adresa, z které bude stažen certifikát portálu. Metoda `get()` požaduje ve svém jediném parametru předat síťový požadavek. Ihned co síťový manažer odešle požadavky a obdrží odpovědi z portálu, provádí se jejich jednotlivé zpracování metodou `receiveData()`.

Přijetí dat z Portálu ZP

V metodě `receiveData()` uchovává přijatá data proměnná `response`. Jelikož byly síťovým manažerem zaslány dva požadavky, data pro portál a požadavek na získání certifikátu portálu, je nutné příchozí data rozlišit. Data proměnné `response` se předají metodě `fromPEM()` třídy `QCA::Certificate` z `QCA` knihovny. Podrobnější popis `QCA` knihovny lze

³Multipurpose Internet Mail Extensions

nalézt v následující podkapitole o třídě `XmlSigner`. Metoda `fromPEM()` vrátí hodnotu `NULL`, pokud zjistí, že data jí předaná nereprezentují strukturu certifikátu dle standardu X.509. Tím je rozlišeno přijetí dat se zprávou uživateli nebo s certifikátem portálu. Certifikát je uložen do souboru umístěném ve složce `cert/portal` OpenDositechu. Data se zprávou pro uživatele pokračují do dalšího zpracování.

Parser třídy `XmlMessageParser` prezentovaný proměnnou `xmlParser` slouží pro analýzu přijatých dat určených uživateli. Parseru jsou předána přijatá XML data a obratem získán obsah elementu `Data` a elementu `Podpis`. Podpis přijatých dat je předán k ověření metodě `verifyXML()` podepisovače `signer`. Zpráva pro uživatele je po převedení z kódování Base64 vhodným způsobem zobrazena v klientovi.

6.2.3 Třída `XmlSigner`

V páté kapitole o elektronickém rozhraní Portálu ZP byla přiblížena problematika elektronického podpisu dat určených portálu. Podpis dat, jak je uvedeno v dokumentaci, musí být proveden formou PKCS#7. Qt framework nenabízí prostředky, které by umožňovaly provádět digitální podpis dat. Pro Qt framework však existuje externí knihovna QCA⁴, která tyto prostředky nabízí. Třída `XmlSigner` využívá funkcí knihovny QCA pro tvorbu podpisu a ověřování podpisu dat zasílaných mezi klientem a portálem.

Knihovna QCA (Qt Cryptographic Architecture)

Jelikož je knihovna QCA pro spolupráci s Qt přímo určena, její používání se nijak neliší od práce s běžnými třídami poskytovanými Qt frameworkem. Možnosti základní knihovny QCA lze podle potřeby rozšířit použitím pluginů. Pro realizaci podpisu splňujícího PKCS#7 používám plugin QCA OSSL. Pro správnou funkci pluginu je vyžadována knihovna OpenSSL. Před použitím jakékoliv třídy knihovny QCA vyžadující podporu některého pluginu je nutné provést inicializaci knihovny. Inicializace knihovny QCA se provádí vytvořením objektu `QCA::Initializer`. Zároveň je potřeba zajistit vytvoření objektu `QCA::Initializer` tak, aby byla zaručena jeho existence po celou dobu využívání knihovny QCA programem.

Podpis a ověřování podpisu dat

Třída `XmlSigner` obsahuje dvě metody: `signXML()` pro tvorbu podpisu a `verifyXML()` pro ověření podpisu. V konstruktoru třídy `XmlSigner` je vytvářen objekt třídy `QCA::CMS` reprezentovaný globální proměnnou `cms`. `CMS`⁵ je určena kromě jiných účelů k tvorbě digitálního podpisu.

Metoda `signXML()` přijímá tři parametry, a to data k podpisu, cestu k certifikátu a heslo k privátnímu klíči certifikátu. Nejprve je uložen certifikát uživatele prostřednictvím metody `fromPEMFile()` třídy `QCA::Certificate` do proměnné `cert`. Poté je provedeno vložení načteného certifikátu do globální proměnné `chain`, představující tzv. řetěz certifikátů třídy `QCA::CertificateChain`. Do proměnné `privateKey` třídy `QCA::PrivateKey` je poté načten obecně použitelný privátní klíč metodou `fromPEMFile()` této třídy. Pro práci s CMS zprávami je nutné vytvořit speciální klíč třídy `QCA::SecureMessageKey` a objekt třídy `QCA::SecureMessage` prezentující zabezpečenou zprávu. Klíč pro CMS je uložen v proměnné `signer`. Pomocí metod `setX509CertificateChain()` a `setX509PrivateKey()` je

⁴Qt Cryptographic Architecture

⁵Cryptographic Message Syntax

klíči pro CMS nastavena veřejná a soukromá část klíče. Proměnná `msg` uchovávající objekt třídy `QCA::SecureMessage` reprezentuje rozhraní pro práci s CMS zprávami. Část kódu metody `signXML()` implementující podpis dat ukazuje kód na obrázku 6.2. Tvorba podpisu probíhá v několika krocích voláním následujících metod třídy `QCA::SecureMessage`:

- `msg.setSigner()` – Nastaví privátní klíč pro podpis.
- `msg.setFormat()` – Nastaví formát výsledného podpisu. Nastaví se na PEM⁶.
- `msg.startSign()` – Spustí operaci podepisování. Nastaví se, aby podpis byl po dokončení operace oddělen od podepisovaných dat.
- `msg.update()` – Předá data pro podepsání.
- `msg.end()` – Dokončí nastavení operace podepisování.
- `msg.waitForFinished()` – Nastaví čas pro provedení operace podpisu.
- `msg.success()` – Zjistí, zdali operace podepisování proběhla v pořádku.
- `msg.signature()` – Vrátí výsledný podpis dat předaných metodou `update()`.

```
// inicializace rozhrani pro zabezpecene zpravy
QCA::SecureMessage *msg = new QCA::SecureMessage(cms);

msg->setSigner(signer);
msg->setFormat(QCA::SecureMessage::Ascii);
msg->startSign(QCA::SecureMessage::Detached);
msg->update(data);
msg->end();
msg->waitForFinished(2000);

if (!msg->success()) {
    ...
}

return msg->signature();
```

Ukázka kódu 6.2: Tvorba digitálního podpisu pomocí knihovny QCA

Metoda `verifyXML()` vyžaduje na svém vstupu dva parametry: data, jejichž podpis má být ověřen, a podpis těchto dat. Do proměnné `portalCert` je uložen certifikát portálu. Certifikát portálu je poté přidán do kolekce certifikátů `QCA::CertificateCollection` v proměnné `cerCollection`. Prostřednictvím metody `setTrustedCertificates()` je vytvořenému objektu `cms` třídy `QCA::CMS()` nastavena zmíněná kolekce. Jinými slovy je zajištěno, že podpis bude ověřován důvěryhodným certifikátem, kterým je certifikát stažený z portálu konkrétní pojišťovny, s níž aplikace komunikuje. V metodě `verifyXML()` dále probíhá vytvoření `QCA::SecureMessage` objektu do proměnné `msg`, stejně jako tomu bylo v metodě `signXML()`. Ověření podpisu pak probíhá v pěti následujících krocích:

- `msg.startVerify()` – Spustí operaci ověření podpisu předaného metodě.
- `msg.update()` – Předá data, jejichž podpis má být ověřen.
- `msg.end()` – Dokončí nastavení operace ověřování podpisu.
- `msg.waitForFinished()` – Nastaví čas pro provedení operace ověření podpisu.
- `msg.success()` – Zjistí, zdali operace ověření podpisu proběhla v pořádku.

⁶Privacy Enhanced Mail

Během implementace podepisování pomocí knihovny QCA bylo nutné vyřešit jeden základní problém, kterým je připojení knihovny QCA a pluginu QCA OSSL k OpenDositechu. Na operačním systému Linux stačí provést instalaci balíčku libqca2 a knihovna je připravena k použití. Na Windows bylo nutné umístit *.dll* soubory knihoven na správná umístění a také přidat potřebné konfigurační soubory pro správnou funkci knihovny QCA ke stávající instalaci Qt frameworku.

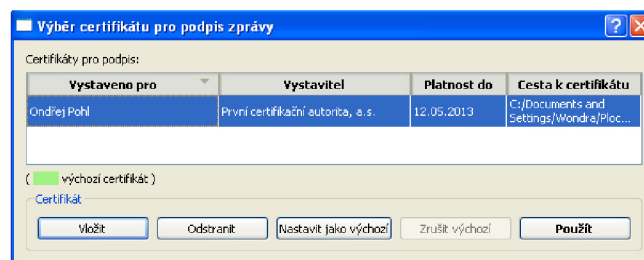
6.2.4 Třída CertInfo

Výběr certifikátu pro podpis dat zasílaných Portálu ZP provádí uživatel po stisknutí tlačítka Výběr certifikátu pro podpis v okně klienta. Okno s výběrem certifikátu ilustruje obrázek 6.3

Konstruktor třídy `CertInfo` přijímá tři parametry: proměnné pro uložení cesty vybraného certifikátu, jména vlastníka certifikátu a uložení data konce platnosti certifikátu. Tabulka seznamu certifikátů prezentovaná globální proměnnou `m_tableView` je objektem třídy `QTableView`. V tabulce se zobrazují tyto informace:

- vlastník certifikátu
- vydavatel certifikátu
- datum konce platnosti certifikátu
- cesta k certifikátu

Pro operace s tabulkou pak slouží model implementovaný třídou `CertInfoModel` zděděnou od třídy `QAbstractTableModel`. Na model se odkazuje globální proměnná `m_table`. Implementace třídy `CertInfoModel` se nachází ve stejném zdrojovém souboru jako implementace třídy `CertInfo`. Po zobrazení okna je přečten soubor `cert/user-cert-list`, ve kterém jsou uloženy cesty certifikátů používaných v OpenDositechu. Certifikáty jsou pak zobrazeny v tabulce. V případě, že soubor `cert/user-cert-list` neexistuje, je vytvořen nový. Poté probíhá načtení certifikátů z úložiště certifikátů OpenDositechu umístěného ve složce `cert/user`. Všechny nalezené certifikáty jsou opět přidány do tabulky.



Obrázek 6.3: Okno s výběrem certifikátu pro podpis

Vložení certifikátu

Pokud uživatel potřebuje vložit do seznamu certifikátů nový certifikát, může to udělat několika způsoby. Nahraje soubor s certifikátem do složky programu `cert/user` a tento certifikát bude při příštím spuštění aplikace automaticky načten. Nebo vloží certifikát z libovolného umístění na disku po stisku tlačítka Vložit. V dialogovém okně pak vyhledá

soubor s certifikátem a vloží jej do seznamu. Vložení certifikátu do seznamu zajišťuje metoda *certInsert()*. Metoda pomocí přijatých parametrů, mezi nimiž je i cesta k certifikátu, rozlišuje, zdali je vkládán certifikát do seznamu ze složky *cert/user*, souboru *cert/user-cert-list* nebo z místa zadaného uživatelem. Při vkládání certifikátu ze složky *cert/user* nebo libovolného umístění probíhá volání metody *isInCertList()*. Tato metoda zabrání tomu, aby byl vložen do seznamu duplicitní certifikát. Z certifikátu jsou poté získány metodami třídy *QCA::Certificate* informace o vlastníkovi (*subjectInfo()*), vydavateli (*issuerInfo()*) a konci platnosti (*notValidAfter()*). Metoda *insertRows()* třídy *CertInfoModel* zajistí vložení nového řádku do tabulky *m_tableView*. Poté jsou postupně do tabulky vloženy informace o certifikátu. Metoda *index()* třídy *CertInfoModel* získá index položky v příslušném sloupci a metoda *setData()* vloží na index danou hodnotu. Je-li metoda *certInsert()* volaná s příznakem vložení výchozího certifikátu, vloží se navíc do posledního skrytého sloupce tabulky příznak 'D' (default). Nakonec je cesta k certifikátu zapsána do souboru *cert/user-cert-list*.

Výběr certifikátu

Vybírá-li uživatel řádek v tabulce s certifikáty, volá se metoda *certSelect()*. Proměnná *selectionModel* uchovává model výběru tabulky *m_tableView*. Jedná se o objekt třídy *QItemSelectionModel*. Metodou *selectedRows()* modelu je získán index vybraného řádku. Podobně jako při vkládání je metodou *index()* zjištěno umístění konkrétní položky v řádku. Metodou *data()* třídy *CertInfoModel* jsou pak postupně získány informace o certifikátu a uloženy. V proměnných *m_certUserSelect*, *m_certUserOwner* a *m_certUserValidity* je pak uložena cesta, vlastník a platnost certifikátu. Údaje o vlastníkovi a platnosti jsou pak zobrazeny v klientovi.

Odstranění certifikátu

Při odstraňování certifikátu ze seznamu certifikátů vybere uživatel příslušný certifikát a stiskne tlačítko Odstranit. Odstranění certifikátu implementuje metoda *certRemove()*. Uživateli je nabídnuta možnost odstranit i samotný soubor s certifikátem, na což je upozorněn zprávou. V každém případě bude certifikát odstraněn ze seznamu a také ze souboru *cert/user-cert-list*.

Cesta k certifikátu není ze souboru *cert/user-cert-list* odstraňována přímo. Nejprve je obsah souboru načten do pomocného seznamu *userCertSet* představujícího seznam *QList* hodnot *QString*. Odstranění cesty certifikátu probíhá nejprve ve zmíněném pomocném seznamu a pak v seznamu s certifikáty. Obsah pomocného seznamu *userCertSet* je na konci metody zapsán do souboru *cert/user-cert-list*.

Výchozí certifikát

Uživatel má možnost zvolit ze seznamu certifikátů jeden certifikát jako výchozí. Nastavení certifikátu jako výchozího provede stiskem tlačítka Nastavit jako výchozí. Řádek s nastaveným výchozím certifikátem je barevně odlišen. V souboru *cert/user-cert-list* je u cesty k tomuto certifikátu přidán příznak 'D'. Při opětovném spuštění *OpenDositechu* uživatel nemusí při odeslání dat vybírat znovu certifikát k podpisu. Výchozí certifikát je automaticky načten do klienta i s informacemi o vlastníkovi a platnosti certifikátu. Uživatel tak může provést okamžité odeslání dat. Zrušení nastavení výchozího certifikátu provede uživatel stiskem tlačítka Zrušit jako výchozí.

Operaci nastavení nebo zrušení certifikátu jako výchozího provádí metoda *certDefault()*. Po načtení cest certifikátů ze souboru *cert/user-cert-list* do pomocného seznamu je hledána cesta s výchozím certifikátem. Pokud uživatel provádí operaci nastavení výchozího certifikátu a výchozí certifikát již existuje, změní se nově zvolený certifikát na výchozí. Nastavit lze vždy pouze jeden certifikát jako výchozí. Ruší-li uživatel nastavení výchozího certifikátu, odstraní se příznak 'D' z cesty příslušného certifikátu. Obsah pomocného seznamu *userCertSet* je v závěru provádění metody zapsán do souboru *cert/user-cert-list*.

Použití certifikátu pro podpis

Stiskem tlačítka Použit se vyvolá metoda *certUse()* a nastaví proměnnou *m_certUser* na hodnotu proměnné *m_certUserSelect*, čímž předá klientovi cestu k vybranému certifikátu pro podpis. Poté je okno s výběrem certifikátu uzavřeno a uživatel se vrací zpět do klienta.

6.2.5 Třída XmlMessageParser

Parser pro analýzu přijatých dat nesoucích XML zprávu z portálu je vytvořen v metodě *receiveData()* třídy *SslClient*. Při tvorbě instance parseru je jeho konstruktoru předán jediný parametr, a tím jsou přijatá data z portálu. V konstruktoru třídy *XmlMessageParser* je vytvořena instance třídy *QXmlStreamReader* s názvem *m_xmlReader* tvořící jádro parseru. Třída *QXmlStreamReader* poskytuje parser pro čtení dat ve formátu XML. XML dokument zpracovává *QXmlStreamReader* jako proud tzv. tokenů různých typů. Například pokud parser narazí při čtení XML na token typu *QXmlStreamReader::StartElement* znamená to, že parser načel počáteční tag určitého elementu.

Parser implementovaný třídou *XmlMessageParser* slouží pro získání obsahu elementu *Data* nebo elementu *Podpis* z přijaté XML zprávy od portálu. Při potřebě získat obsah posledně jmenovaných elementů je vždy volána metoda *parseXML()*. Parametr předaný této metodě obsahuje název elementu, jehož obsah je požadován.

Čtení XML provádí v cyklu *m_xmlReader* tak dlouho, dokud není dosaženo přečtení kompletního XML dokumentu. Dosažení konce XML dokumentu testuje metoda *atEnd()* třídy *QXmlStreamReader* vždy na začátku každého průchodu cyklem. Vždy je přečten jeden token metodou *readNext()* a zjištěn typ tokenu. Je-li nalezen počáteční tag nějakého elementu, zjistí se název tohoto tagu. V případě nalezení počátečních tagů elementu *Data* nebo elementu *Podpis* se volají příslušné metody *parseDataElement()* nebo *parsePodpisElement()*. V metodě *parseDataElement()* probíhá postupným voláním metody *readNext()* přes *m_xmlReader* posun v XML dokumentu až na obsah elementu Soubor. Dle dokumentace elektronického rozhraní Portálu ZP obsahuje element *Soubor* odpověď portálu a ta je vrácena jako výsledek metody. Při volání metody *parsePodpisElement()* je postup podobný jako v předchozím případě. Po nalezení elementu *Podpis* je metodou vrácen jeho obsah.

6.2.6 Třída AccountDialog

Při tvorbě dávky pro pojišťovnu v okně *Dávky > Vytvoření dávky* je spolu s dávkou vytvářena i faktura dávky. Nastavení platebních údajů na faktuře pro pojišťovnu provádí uživatel po stisku tlačítka Údaje pro platbu ve zmíněném okně. Okno pro zadání platebních údajů je implementováno v metodě *setupUI()* třídy *AccountDialog*. Nastavení údajů pro platbu probíhá v metodě *setAccountInfo()*. Uživatel zadává v okně pro platební údaje

znázorněném obrázku 6.4 předčíslí účtu, číslo účtu, kód banky a specifický symbol. Zadané údaje jsou po stisku tlačítka OK zapisovány do souboru *account-info* a automaticky načítány při opětovném spuštění aplikace. V souboru *account-info* jsou jednotlivé platební údaje odděleny znakem ',' (čárka) a uvedeny na jednom řádku. Platební údaje jsou uloženy v souboru v tomto pořadí: kód banky, specifický symbol, předčíslí účtu a číslo účtu. Bez nastavených platebních údajů nelze provést vytvoření dávky.

Obrázek 6.4: Okno pro zadání platebních údajů na fakturu

6.3 Další úpravy aplikace OpenDositech

Nyní se budu věnovat několika úpravám původního programu OpenDositech. Tyto úpravy měly za cíl vylepšení uživatelského rozhraní aplikace OpenDositech.

6.3.1 Zobrazení přehledu dávek ve stromu

Uživatel má v základní implementaci OpenDositech v okně *Dávky > Přehled* zobrazený přehled dávek ve formě tabulky, jak je vidět z obrázku 6.5. Nevýhodou původního zobrazení přehledu v tabulce je značná nepřehlednost při větším počtu vytvořených dávek. Každá dávka je předložena pojišťovně za určitý rok a měsíc. Tabulka sice obsahuje skryté sloupce s těmito údaji, ale při zobrazení přehledu tyto parametry nezohledňuje. Během řešení této práce jsem proto provedl převedení obsahu tabulky do stromu. Qt framework nabízí pro tyto účely třídu *QTreeWidgetItem*. Nový přehled dávek lze spatřit na obrázku 6.6.

	IČZ	číslo dávky	Pojišťovna	Počet dokladů	Počet bodů
1	94465000	450002	209	1	121
2	94465000	422222	217	2	242
3	94465000	666666	205	1	121
4	94465000	333334	205	1	121
5	94465000	781111	201	1	242
6	94465000	149882	207	1	121

Obrázek 6.5: Původní přehled dávek OpenDositech

Implementaci stromu pro přehled dávek jsem provedl v původní třídě *ViewBatchDialog* v metodě *setUpUI()*. Původní tabulku prezentovanou proměnnou *m_view* třídy *QTableView* jsem zachoval. Tabulka zůstala napojena na databázi aplikace. Napojení na databázi je

provedeno přes globální proměnnou `m_model` třídy `QSqlTableModel`. Provedl jsem skrytí původní tabulky a na její místo umístil objekt `QTreeWidget` zobrazující strom s přehledem dávek. Stávající tabulku `m_view` dále využívám jako zdroj dat. Strom je prezentován proměnnou `m_treeWidget`. Do stromu je poté přidán řádek s hlavičkou stejnou jako v tabulce. Navíc je přidán sloupec pro zobrazení roku a měsíce a skrytý sloupec pro ukládání indexu řádků zdrojové tabulky `m_view`. Jednotlivé řádky stromu představují objekty třídy `QTreeWidgetItem`. Nejdříve jsou ve stromu vytvořeny nové prázdné řádky pro roky a měsíce. Nakonec se strom naplní daty ze zdrojové tabulky.

Rok -> Měsíc	IČZ	Číslo dávky	Pojišťovna	Počet dokladů	Počet bodů
2013-5	94465000	422222	217	2	242
2013	94465000	333334	205	1	121
2012-5	94465000	666666	205	1	121
2012-10	94465000	450002	209	1	121
2011-Nevydané	94465000	149852	207	1	121

Obrázek 6.6: Nový přehled dávek OpenDositechu

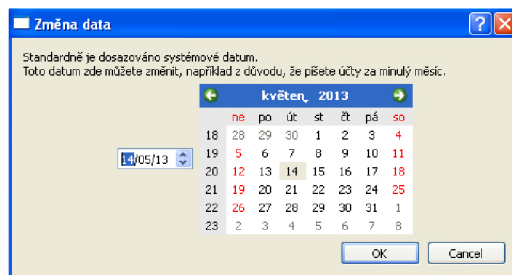
Nejprve je prováděn průchod zdrojovou tabulkou `m_view` a uloženy nalezené roky do seznamu `years`. Není-li v daném řádku zdrojové tabulky uveden rok, znamená to, že dávka byla vytvořena, ale nebyla ještě předána pojišťovně. Jednotlivé roky jsou pak vloženy do stromu a ke každému roku je pak přidáno dvanáct řádků pro jednotlivé měsíce. Všechny řádky s měsíci jsou skryty a odkrývají se až v případě, když je pro příslušný měsíc nalezena dávka ze zdrojové tabulky. Na stejnou úroveň s roky je do stromu vložen speciální uzel s názvem `Nevydané` v případě, že zdrojová tabulka obsahuje i dávky zatím nepředložené pojišťovně k vyúčtování.

Plnění stromu daty probíhá průchodem řádky zdrojové tabulky a vkládáním těchto řádků do stromu. Po zjištění hodnoty roku a měsíce z daného řádku tabulky se všechny hodnoty z tohoto řádku vloží na odpovídající místo ve stromu. Je-li nalezena dávka bez časových údajů, vkládá se automaticky pod uzel `Nevydané`. Do skrytého sloupce stromu s názvem `IndexSQL` se pro každý vložený řádek zapíše hodnota indexu řádku zdrojové tabulky. Výběr řádků zpracovává metoda `treeViewSelectedItem()`. Při výběru řádků ze stromu probíhá zároveň výběr odpovídajícího řádku ve zdrojové tabulce. V ostatních původních metodách třídy `ViewBatchDialog` jsem neprováděl žádné změny. I nadále tyto metody využívají původní tabulku s přehledem dávek. Uživatel však pracuje se stromem, jenž používá původní tabulku pouze jako zdroj dat. Při zobrazení okna s přehledem dávek je automaticky označena nejnovější dávka.

6.3.2 Nastavení data a IČO

V okně *Nastavení > Změna data* lze změnit datum v případě, že uživatel potřebuje z nějakého důvodu vypsát účty za předešlé měsíce. Původní verze OpenDositechu implementuje výběr data přes objekt třídy `QDateEdit` prezentovaný proměnnou `dateEdit`. Objekt `QDateEdit` umožňuje přes metodu `setCalendarWidget()` nastavit kalendář pro výběr data, jak si lze všimnout na obrázku 6.7. Uživatel tak již nemusí pracně volit požadované datum, jako tomu bylo dříve.

V nové verzi OpenDositechu je přidána fakturace dávek. Soubor `FDAVKA` předávaný Portálu ZP musí dle datového rozhraní obsahovat kromě IČZ i IČO zdravotnického zařízení.



Obrázek 6.7: Výběr data v nové verzi OpenDositechu

Z toho důvodu bylo upraveno okno pro zadání IČZ, jehož implementaci provádí původní třída ICZdialog. Zadané IČO se ukládá do konfiguračního souboru *config.xml*. Novou podobu okna pro zadání identifikačních údajů zdravotnického zařízení ilustruje obrázek 6.8.



Obrázek 6.8: Nové okno pro zadání identifikačních údajů ZZ

6.4 Testování

K ověření správnosti implementace rozhraní pro komunikaci s Portálem ZP jsem využil testovací komunikační bránu Portálu ZP. Testovací brána je dostupná na adrese https://pilot-pzp.assec.co.cz/kom_brana.phtml. Testovací komunikační brána se od ostatních komunikačních bran jednotlivých pojišťoven odlišuje v několika bodech:

- Data přijatá bránou nejsou předány žádné pojišťovně ke zpracování.
- Brána přijímá fakturu a dávku s doklady pro pojišťovnu s testovacím kódem 999.
- Zasláná data se podepisují testovacím certifikátem. Testovací certifikát je zaveden k vytvořenému účtu uživatele v testovacím prostředí Portálu ZP.
- Dávky a faktury za zdravotnické zařízení jsou přijímány jen po rozšíření oprávnění účtu uživatele. V testovacím prostředí lze zaregistrovat libovolné IČZ. Administrátor Portálu ZP pak přiřadí v testovacím prostředí oprávnění k IČZ, za které uživatel požaduje přístup.

Testování v ostrém provozu na komunikačních branách konkrétních pojišťoven Portál ZP nedoporučuje. Portál ZP určil pro testování komunikace testovací prostředí zahrnující testovací bránu. Základní rozdíl mezi ostrým provozem a testovacím provozem je, že požadavky zpracovávané testovacím prostředím jsou implicitně ignorovány. Testovací prostředí je využíváno Portálem ZP také při nasazování novinek.

Testovací certifikát jsem si nechal vystavit u I.CA⁷. Testovací certifikát má platnost 30 dní. Žádost o testovací certifikát jsem prováděl na operačním systému Windows. Poté jsem

⁷První certifikační autorita, a.s. - <http://www.ica.cz/>

provedl vyexportování certifikátu i s privátním klíčem do souboru *.pfx*. Nakonec pomocí utility OpenSSL jsem příkazem `openssl pkcs12 -in mycert.pfx -out mycert.pem` provedl převod certifikátu s privátním klíčem do formátu *.pem*. Zařídil jsem si registraci uživatelského účtu v testovacím prostředí Portálu ZP a ve spolupráci s administrátorem rozšířil oprávnění pro mnou zadané IČZ.

Během řešení této práce a testování jsem využíval také demo aplikaci nabízenou Portálem ZP. Při zasílání testovacích dat jsem porovnával výstupy z demo aplikace s výstupy OpenDositechu. Testovací data jsem vytvořil v OpenDositechu jako dávku s několika doklady a spolu s fakturou zaslal na testovací bránu. Odpověď testovací brány na data zasláná z OpenDositechu přes implementované rozhraní lze vidět na obrázku 6.9. Brána při úspěšném zpracování dat k vyúčtování odpovídá protokolem o přijetí faktury a dávky.

ZP	Zdravotní pojišťovna	10.05.2013																								
	Protokol o zápisu faktury	20:10:38																								
IČZ																										
	94465000																									
Faktura č. 123/2013 byla založena pod ref. číslem 62440																										
Faktury obsažené ve vyúčtování		Položky obsažené ve faktuře																								
Číslo faktury	123	Char. Typ IČZ Rok Měsíc Číslo Počet Body Kč/mat.																								
Měsíc/Rok	5/2013	dávky dokl.																								
Datum předání	10.05.2013	P 98 94465000 2013 5 422222 2 242 0.00																								
Fakturovaná částka	0.00																									
Počet dávek	1																									
Poznámka																										
<p>Varování 212: V souboru KDAVKA předání tito pojištěnci, kteří v daném období nebyli pojištěni u ZP :</p> <table border="1"> <thead> <tr> <th>IČZ</th> <th>Dávka</th> <th>Doklad</th> <th>Řádek dokladu</th> <th>Rok</th> <th>Číslo pojištěnce</th> <th>Datum</th> <th>Pojištěn u</th> </tr> </thead> <tbody> <tr> <td>94465000</td> <td>422222</td> <td>50001</td> <td></td> <td>2 2013</td> <td>9001225761</td> <td>15.04.2013</td> <td></td> </tr> <tr> <td>94465000</td> <td>422222</td> <td>50002</td> <td></td> <td>4 2013</td> <td>1111111111</td> <td>15.04.2013</td> <td></td> </tr> </tbody> </table>			IČZ	Dávka	Doklad	Řádek dokladu	Rok	Číslo pojištěnce	Datum	Pojištěn u	94465000	422222	50001		2 2013	9001225761	15.04.2013		94465000	422222	50002		4 2013	1111111111	15.04.2013	
IČZ	Dávka	Doklad	Řádek dokladu	Rok	Číslo pojištěnce	Datum	Pojištěn u																			
94465000	422222	50001		2 2013	9001225761	15.04.2013																				
94465000	422222	50002		4 2013	1111111111	15.04.2013																				

Uživatel: Ondřej Pohl

Obrázek 6.9: Protokol o úspěšném přijetí požadavku na vyúčtování

Kapitola 7

Závěr

Cílem této práce byla implementace rozhraní pro komunikaci s pojišťovnami Portálu ZP do aplikace OpenDositech. Úspěšné splnění cíle práce bylo ověřeno na testovací bráně Portálu ZP.

V úvodní části práce byly představeny vybrané lékařské programy a jejich možnosti použití. Následovalo vysvětlení principů předávání dokladů zdravotním pojišťovnám a popis datového rozhraní, s jehož využitím předávání dokladů probíhá. Čtenář byl seznámen s Portálem ZP a elektronickým rozhraním portálu. V kapitole o implementaci byla na úvod stručně popsána základní implementace programu OpenDositech. Další podkapitoly se věnovaly výhradně implementaci rozhraní a jiným úpravám původní verze programu.

Nová verze OpenDositechu nyní obsahuje následující změny:

- přidáno rozhraní pro komunikaci s Portálem ZP
- přidaná tvorba faktur za dávky
- nový přehled dávek zobrazovaný jako strom
- nový způsob nastavení data

K dosažení cíle této bakalářské práce bylo potřeba nastudovat problematiku předávání dokladů zdravotním pojišťovnám. Podstatné bylo seznámení s původním programem OpenDositech a jiným lékařským softwarem. Dále bylo nutné pochopit princip fungování elektronického rozhraní Portálu ZP, k čemuž mi dopomohla dostupná dokumentace. Pro úspěšnou implementaci řešení bylo nezbytné naučit se práci s knihovnou QCA a tuto knihovnu pak v OpenDositechu vhodně použít.

Z pohledu dalšího vývoje OpenDositechu lze předpokládat další úpravy a rozšíření. Mezi budoucí úpravy bezesporu patří možnost zadávat účty pro zbývající typy dávek definovaných VZP. Další úpravou nutnou pro využití OpenDositechu v praxi je implementace výpočtu hodnoty faktury v Kč. Pro další vylepšování programu OpenDositech se lze inspirovat v lékařských softwarech dostupných na trhu. Jedním z těchto vylepšení může být kartotéka pacientů s vedením jednoduché lékařské dokumentace. OpenDositech s navrhovanými budoucími úpravami by se mohl bez nadsázky stát reálnou konkurencí některých lékařských softwarů. OpenDositech navíc disponuje jednou významnou konkurenční výhodou - je dostupný i pro lékaře využívající ve svých ordinacích operační systém Linux.

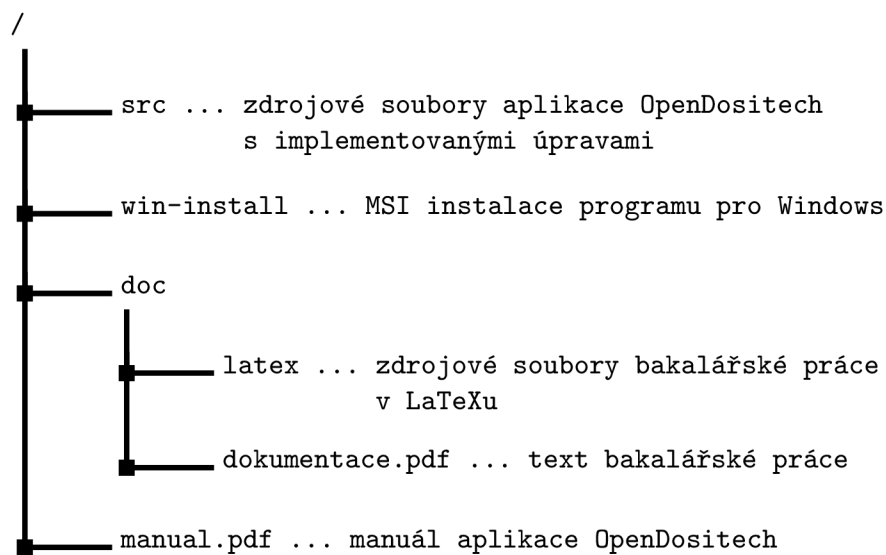
Literatura

- [1] Asseco Central Europe. Uživatelský manuál. *Portál zdravotních pojišťoven* [online]. 2012 [cit. 15. 11. 2012].
Dostupné z: <<http://www.portalzp.cz/download/PriruckaPortaluZP.pdf>>
- [2] BENEŠ, Vladimír. Popis komunikace s portálem (klient). *Portál zdravotních pojišťoven* [online]. 2012 [cit. 15. 12. 2012].
Dostupné z: <<http://goo.gl/09mai>>
- [3] Datové rozhraní VZP ČR. *Všeobecná zdravotní pojišťovna České republiky* [online]. 2012 [cit. 29. 11. 2012].
Dostupné z: <<http://www.vzp.cz/uploads/document/dr1-v6221-pdf.pdf>>
- [4] Datové rozhraní VZP ČR – individuální doklady. *Všeobecná zdravotní pojišťovna České republiky* [online]. 2012 [cit. 29. 11. 2012].
Dostupné z: <<http://www.vzp.cz/uploads/document/dr2-v6221.pdf>>
- [5] Jak se stát uživatelem portálu ZP. *Portál zdravotních pojišťoven* [online]. 2012 [cit. 14. 11. 2012].
Dostupné z: <<http://www.portalzp.cz/jak-se-stat-uzivatelem-portalu-zp>>
- [6] Lékařský software. *Lékařský software autorů společnosti Themis* [online]. 2012 [cit. 12. 10. 2012].
Dostupné z: <<http://www.themis.cz/>>
- [7] Metodika pro pořizování a předávání dokladů VZP ČR. *Všeobecná zdravotní pojišťovna České republiky* [online]. 2012 [cit. 29. 11. 2012].
Dostupné z: <<http://goo.gl/Q2TDA>>
- [8] Portál ZP. *Portál zdravotních pojišťoven* [online]. 2012 [cit. 14. 11. 2012].
Dostupné z: <<http://www.portalzp.cz/>>
- [9] Portál ZP - Komunikační brána pro klienty. *Portál zdravotních pojišťoven* [online]. 2012 [cit. 14. 12. 2012].
Dostupné z: <<http://www.portalzp.cz/distribuce.ext/AktualniStav.html>>
- [10] Pravidla pro vyhodnocování dokladů ve VZP ČR. *Všeobecná zdravotní pojišťovna České republiky* [online]. 2012 [cit. 28. 11. 2012].
Dostupné z: <<http://goo.gl/iLzPL>>
- [11] Programy řady MEDICUS 3. *CompuGroup Medical Česká republika s.r.o* [online]. 2012 [cit. 12. 10. 2012].
Dostupné z: <<http://www.medicus.cz/produkty/programy/>>

Příloha A

Obsah CD

Obsah přiloženého CD má následující strukturu:



Příloha B

Ukázky datových rozhraní

Příklad datového rozhraní číselníku Mezinárodní klasifikace nemocí:

Název	Typ	Délka	Popis
KOD	C	5	Kód diagnózy
ZNA	C	1	Znak pro hvězdičkové diagnózy
POH	C	1	Pohlaví
VOD	C	2	Věk od
VDO	C	2	Věk do
NAZ	C	70	Název diagnózy

Příklad datového rozhraní záhlaví dokladu Vyúčtování výkonů v ambulanci péči:

Název	Typ	Délka	Začátek	Popis
TYP	C	1	0	Typ věty „A“ – záhlaví
HCID	N	7	1	Číslo dokladu
HSTR	N	1	8	Pořadové číslo listu dokladu
HPOC	N	1	9	Celkový počet listů dokladu
HPOR	N	3	10	Pořadové číslo listu v dávce. Povolený rozsah od 1 do 999.
HCPO	C	3	13	Číslo pojišťovny
HPPP	C	1	16	Typ připojištění
HICO	C	8	17	Identifikační číslo pracoviště (IČP)
HVAR	C	6	25	Variabilní symbol. Nepovinné, nemusí se vyplnit.
HODB	C	3	31	Smluvní odbornost pracoviště
HROD	C	10	34	Číslo pojištěnce
HZDG	C	5	44	Číslo základní diagnózy, pro kterou byl pacient v tomto období léčen
HKON	C	1	49	Údaj se nevyplňuje
HICZ	C	8	50	IČP žadatele – nevyplňuje se
HCDZ	N	7	58	Číslo dokladu žadatele – nevyplňuje se
HREZ	C	10	65	Rezerva – nevyplňuje se
HCCEL	\$	10.2	75	Cena celkem – nepovinný údaj
HCBOD	N	7	85	Body celkem – nepovinný údaj
DTYP	C	1	92	Doplněk typu věty záhlaví. Rezerva, nevyplňuje se.
Celkem: délka věty 93				

Příklad obsahu souboru KDAVKA:

```
DP981234567899902008020003360010000000026500000000000000000.001
A000196400001999112345678      00162551001262108      0      0.00      0
V26022008010221      Z108      265
```

Příloha C

Portál ZP

Ukázka portálu Vojenské zdravotní pojišťovny ČR:

The screenshot shows the login page of the Vojenská zdravotní pojišťovna ČR (VozP). The header includes the logo, the name of the organization, and the user's policy number (201). A navigation menu on the left lists various services. The main content area is titled 'Přihlášení do systému - Vstup pro registrovaného uživatele' and contains three main options: logging in with a certificate, logging in via SMS, and registering for a new account. There are also links for account settings and a user profile section at the bottom left.

Hlavní okno demo aplikace knihovny ASSECOPZP:

The screenshot shows the main window of the ASSECOPZP demo application. The title bar reads 'Demo aplikace pro knihovnu ASSECOPZP.dll'. The window title is 'Portál ZP' and the subtitle is 'Demo aplikace pro knihovnu ASSECOPZP'. The interface is divided into several sections: 1. Authentication: Radio buttons for 'Certifikátem' (checked) and 'SMS kódem'. 2. Selection: Radio buttons to choose the insurance company (VozP ČR is selected) and the service (VYU is selected). 3. Parameters: A section for 'Nepovinné parametry' with a timeout of 300 seconds and a certificate ID. 4. Files: A list of files to be sent, including two PDF invoices. 5. Buttons: 'Zažádat o SMS kód', 'Odeslat podání', and 'Konec'.