

Absolventská práce

Využití webových služeb SOAP pro přizpůsobení funkcí VoIP telefonní ústředny IPEX pro hotelové služby.

Autor:

Antonín Ečer

Vedoucí práce: Vladimír Klimeš

Vzdělávací program: 26 – 47 – N / 01: Informační technologie

Obor vzdělání : 26 – 47 – N / 01: Informační technologie

Září 2011

Poděkování

Za pomoc, trpělivost a směřování mé kreativity při tvorbě absolventské práce chci poděkovat především svému třídnímu učiteli a zároveň vedoucímu práce panu Ing. Vladimíru Klimešovi.

Prohlášení

Prohlašuji, že jsem svou absolventskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona 151/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 16.12.2010

.....

Abstrakt

Využití webových služeb SOAP pro přizpůsobení funkcí VoIP telefonní ústředny IPEX pro hotelové služby.

Hlavním výsledkem mé práce je vytvoření jednoduchého a spolehlivého interface pro VoIP telefonní ústředny IPEX a jejich přizpůsobení pro použití v hotelovém provozu. Telefonní ústředny IPEX podporují užití webových služeb (SOAP), které by měly plně postačovat k aplikaci základních hotelových služeb, jako jsou Odemknutí/Uzamknutí linky v hotelovém pokoji a dále buzení hosta telefonem s kontrolou vzbuzení a předání tarifikačních údajů hotelovým interfacům.

Toto téma jsem si zvolil především proto, že jsem byl osloven zaměstnavatelem, zdali bych byl schopen vytvořit interface pro VoIP telefonní ústředny IPEX. Do budoucna mám v plánu interface dále rozšiřovat dle nově identifikovaných provozních potřeb. V rámci absolventské práce jsem vytvořil aplikaci, která poslouží jako základní verze interface pracovně pojmenovanou HIFAS 1.0. Jedná se o zkratku slov Hotelový Interface Asterisk.

Abstract

Usage of the SOAP web services for an adaption of IPEX VoIP PBX's functions for hotel services.

The main result of my work is to creating a simple, reliable interface for IPEX VoIP PBX and adapt them for hotel operation usage. IPEX telephone exchange supports use of the web services (SOAP), which should fully comply with basic hotel service requirements, such as unlock / lock the phone line in a hotel room. Next functions are wake-up control and transfer of tarification informations to the PMS interface.

I chose this topic because I was asked by my employer to create an interface for the IPEX VoIP telephone exchange. I plan to develop the interface according to future business needs. As a part of this thesis, I created an application serving as a basic version of the interface software named HIFAS 1.0. Its name is an abbreviation of Hotel Asterisk Interface.

1 Úvod	7
1.1 Přínosy webových služeb	7
1.2 Historie webových služeb	8
1.3 SOAP v současnosti	9
1.4 Pojednání o možném využití webových služeb v budoucnosti	9
2 Popis problému, specifikace cíle	11
2.1 Popis problému	11
2.2 Požadavky na implementovaný systém	11
3 Analýza a návrh řešení	13
3.1 Analýza používaných protokolů	13
3.2 Zjištění používaných funkcí v hotelové praxi	13
3.3 Volba vhodného databázového stroje	15
3.4 Analýza komunikace stávajícího systému Caracas-Link	15
3.5 Analýza komunikace přes SOAP	16
4 Návrh	17
4.1 Návrh tabulek databáze	17
4.2 Návrh funkcí programu HIFAS	20
4.2.1 Konfigurace spojení s SQL a telefonní ústřednou IPEX	20
4.2.2 Komunikace s hotelovým telefonním rozhraním ústřednou IPEX	23
4.2.3 Konfigurace vlastností pro sestavení SOAP zpráv	25
4.2.4 Automatické načtení linek z ústředny	25
4.2.5 Manuální vyzískání tarifikačních údajů	25
4.2.6 Manuální nastavení odchozích oprávnění na jednotlivých linkách	25
4.2.7 Možnost vynucení volání z linky na jiné číslo	25
4.2.8 Výpis nastavených buzení	25
4.2.9 Odmazání vybraných záznamů buzení z databáze	26
4.2.10 Načtení přenosového souboru	26
4.2.11 Zapnutí a vypnutí automatického provozu programu	26
4.2.12 Ukončení programu	26
4.3 UML diagram tříd a jejich popis	27
4.4 Typický cyklus práce programu	37
5 Vývoj	39
5.1 Výběr vývojového prostředí	39
5.2 Testovací prostředí	39
5.2 Vyvstálé problémy	39
5.3 Zprovoznění nového testovacího prostředí	42
5.3.1 Postup otevření projektu v Netbeans	42
6 Odzkoušení funkčnosti s PMS systémy	49
7 Ochranný mechanismus	51
8 Prezentace programu dodavateli telefonních ústředn IPEX	53
9 Závěr	55
9.1 Možná rozšíření do budoucna	55
10 Seznam použitých zkratk	57
11 Seznam použitých zdrojů	59
12 Seznam příloh	63

1 Úvod

Webová služba je softwarový systém umožňující interakci dvou strojů na síti. S webovou službou ostatní stroje komunikují způsobem předepsaným v jejím popisu pomocí protokolu [SOAP](#) [1] přepravené pomocí jiných, již zavedených protokolů, tzv. Tunelování firewallu. Důvodem pro „zapouzdření“ SOAP zprávy je nedostatek důvěry ze strany [firewallů](#) [2] již zavedených systémů.

SOAP (celým názvem Simple Object Access Protocol) je protokolem pro výměnu zpráv založených na XML přes síť, hlavně pomocí HTTP. Formát SOAP tvoří základní vrstvu komunikace mezi webovými službami a poskytuje prostředí pro tvorbu složitější komunikace. SOAP poskytuje jednoduchý návod jak uvést do praxe distribuované komponenty. Stručně řečeno, radí: vezměte svoje dosavadní služby a zajistěte, aby byly schopny komunikovat přes velice jednoduchý protokol přenášený přes HTTP. Tím s minimálními náklady získáte možnost začlenit vaše služby do široké komunity Internetu.[3] Existuje několik různých druhů šablon pro komunikaci na protokolu SOAP. Nejznámější z nich je RPC(Remote procedure call) šablona, kde jeden z účastníků komunikace je klient a na druhé straně je server. Server ihned odpovídá na požadavky klienta.

SOAP je nástupce XML-RPC, ačkoliv si zapůjčuje jeho způsob přenosu dat a další vlastnosti.

Původně ho navrhl Dave Winer, Don Box, Bob Atkinson a Mohsen Al-Ghosein v roce 1998 za podpory firmy Microsoft (kde tou dobou Atkinson a Al-Ghosein pracovali).[1]

Dnes je SOAP specifikace držena XML skupinou tvořící internetové protokoly z W3C konsorcia.

1.1 Přínosy webových služeb

Hlavním přínosem webových služeb je přijetí jejich protokolů jako standardů mezi systémové komunikace. Vytvářejí tak vhodný nástroj komunikací mezi firemními systémy a tím i zefektivnění B2B ([Business-to-Business](#))[4] operací.

Jako vedlejší přínos, ale stále velmi významný, se jeví přijetí protokolu [SOAP](#) jako standardu pro přenášení zpráv mezi programy nebo systémy. Mnoho programů tak nabízí rozhraní pro využívání jeho nástrojů prostřednictvím zpráv [SOAP](#). Jednu z možností využití SOAP protokolu v této práci představím na konkrétním příkladu mého programu HIFAS vytvořeného v programovacím jazyce Java a využívajícího pro uložení dat Microsoft SQL Serveru 2005 Express edici.

1.2 Historie webových služeb

Ačkoliv webové služby představují novou technologii, můžeme hodně pochopit, pokud známe jejich nedávnou historii. V devadesátých letech minulého století se objevila technologie komponent, která umožnila budovat aplikace jejich sestavováním. Technologie založená na komponentech je skutečným rozšířením objektově orientovaných principů za hranice libovolného jazyka, takže se stává jádrem infrastruktury, kterou může kdokoli použít. Zatímco objektově orientovaný jazyk umožňoval vývojářům opakovaně používat objekty ve svých aplikacích, technologie založená na komponentách jim umožňovala snadno sdílet zkompilované objekty mezi aplikacemi. Vznikly dvě dominantní technologie založené na komponentách – COM (Component Object Model) a CORBA (Common Object Request Broker Architecture). Od té doby se objevily další technologie založené na komponentách (jako například JavaBeans a .NET), nicméně jsou navrženy jako proprietární řešení pro konkrétní programovací prostředí.

Krátce poté, co byly COM a CORBA vytvořeny, byly tyto standardy použity u distribuovaných komponent tak, aby aplikace mohla pracovat s objekty umístěnými v různých počítačích zapojených do sítě. Ačkoli COM i CORBA jsou po technické stránce velice sofistikované, je často velmi obtížné je nastavit a podporovat v prostředí lokální sítě, nehledě na fakt, že nemohou fungovat společně. S objevením Internetu se pak logicky objevil dramatický nárůst dalších problémů. I přesto všechno je vývojáři začali používat při vytváření distribuovaných aplikací, tzn. WAN (Wide Area Networks) aplikací, které se ovšem velmi pomalu rozšiřovaly a které také byly méně spolehlivé. Protože technologie COM i CORBA jsou velice komplexní a současně proprietární, ani jedna z nich se v tomto prostředí neprosadila natolik, aby se to dalo považovat za úspěch.

Webové služby jsou novou technologií, která má za cíl vyřešit tyto problémy rozšířením technologie objektů a komponent do webového prostředí. Webová služba je v zásadě jednotkou aplikační logiky (komponentou), která může být vzdáleně zavolána přes Internet. Očekávání vkládaná do webových služeb se v mnohém shodují s očekáváními vkládanými do technologie komponent. Jejich cílem je usnadnit sestavování aplikací pomocí hotové aplikační logiky, sdílet funkcionalitu mezi partnery a vytvářet mnohem modulárnější aplikace. Na rozdíl od dřívějších technologií založených na komponentách jsou webové služby navrženy výhradně pro tento účel.[5]

1.3 SOAP v současnosti

V současnosti se webové služby využívají především v B2B komunikaci, kde probíhá velký objem předávaných dat a obchod je vázán časovými limity, jako jsou uzávěrky obchodování. Vzniká nový trend komunikace A2A (aplikace komunikuje přímo s jinou aplikací).

Uvedu zde několik příkladů společností masivně využívající webové služby. Jako první uvedu společnost OTE a.s., která používá SOAP pro následující úkony mezi subjekty obchodu : zadání nabídky, anulace nabídky, akceptace nabídky, zadání objednávky a anulace objednávky, viz seznam použitých zdrojů [6]

Jako další příklad zde uvedu Ministerstvo financí ČR, které používá webové služby SOAP pro komunikaci všech účetních jednotek s centrálním systémem účetních informací státu (typické pro sběr účetních záznamů, či stažení číselníku, stavové zprávy, požadavky na data vydané CSÚIS) .

1.4 Pojednání o možném využití webových služeb v budoucnosti

Webové služby dovolují spojit různé aplikace bez ohledu na počítačovou platformu, použité programovací jazyky a síťové protokoly. Díky jejich použití eliminujeme možnost vzniku chyb způsobených lidským faktorem při přepisu dat, například při vyřizování emailové objednávky. Dynamický rozvoj a využití webových služeb v posledních letech naznačuje masový nástup aplikací využívajících webové služby. Příkladem bych uvedl oblast mobilních telefonů. Zde jako průkopníka vyzdvihuji firmu Nokia s jejími Nokia web aplikacemi. Dále obor měření a regulace, kde zmíním firmu IBM a její projekt „chytřejší města“, který umožňuje maximální zefektivnění jak rozvodů elektrické energie které již dávno nejsou pouze jednosměrné, tak chytřejší dodávky vody s čidly průtoku a filtrací, v dopravě propojení systémů městské hromadné dopravy a v neposlední řadě též čidla a spotřebiče v domácnostech, které nám umožní větší úsporu energie, kupříkladu naplánováním doby praní. Pro bližší informace o projektu IBM „Chytřejší město“ navštivte následující stránky: <http://www-05.ibm.com/innovation/cz/thesmartercity/> , kde se spustí opravdu poutavá Flash presentace.

Osobně se těším na dobu, kdy díky masově nasazeným webovým službám dostane lidstvo přístup ke skoro každé myslitelné informaci, či službě. Zkuste si představit, že vyndáte z ledničky mléko a po vrácení na místo zjistí lednička sama jeho zbylé množství (všechny produkty již mají čárový kód a porovnáním s přednastaveným seznamem dojde k zahájení interakce) a tak Vás ihned upozorní, že daný produkt dochází a požádá Vás o objednávku. Zde začíná kouzlo SOAP - stačí nastavit na Vašem spotřebiči jen URL oblíbeného

obchodu, jméno, heslo a mít podle dohody zboží doručené, či připravené k vyzvednutí, ať již jste kdekoli.

Tedy orientace malých a středních podniků na prezentaci svých produktů v rámci svého teritoria a užití webových služeb je nasnadě. Proto se domnívám, že vývoj webových aplikací komunikujících přes SOAP tyto podniky jednoduše propojí a tím dojde k urychlení a zjednodušení běžných i ne úplně všedních úkonů, například rezervací a podobně.

2 Popis problému, specifikace cíle

Jak jsem již nastínil v úvodu, budu se v této práci zabývat užitím SOAP protokolu k přizpůsobení služeb telefonní ústředny IPEX (jde o nadstavbu telefonního systému ASTERISK) pro užití v hotelovém prostředí. Pro implementaci byl použit programovací jazyk Java a databázový systém Microsoft SQL Server 2005 edice Express.

2.1 Popis problému

V návaznosti na potřeby nalezení zdroje úspor bylo v naší mateřské organizaci CIMEX INVEST s.r.o. (firma zabývající se mimo jiné pronájmem nemovitostí, bytů a nebytových prostor) implementováno VoIP telefonní řešení firmy IPEX a to v několika administrativních budovách včetně našeho úseku společnosti Aiteco. Při jednání s dodavateli bylo zmíněno, že velmi významnou částí koncernu Cimex je hotelový řetězec Orea, který vlastní několik desítek ubytovacích zařízení, kde by se úspory na hovorném daly využít. VoIP telefonní ústředny jsou ideální pro užití v kancelářských budovách vzhledem k nákladům hovorného proti klasickým telefonním operátorům. Nedokáží již však poskytnout ani základní hotelové služby, které jsou standardně hotelem poskytovány hostům v rámci dané kvality ubytovacích služeb, například buzení telefonem a okamžitý tarifikační výstup.

2.2 Požadavky na implementovaný systém

Mezi požadované funkce, které jsou od ústředny použitelných v hotelovém provozu očekávány, patří následující vyjmenované funkce:

- Check in/out (zakázání / povolení odchozích volání z hotelového pokoje)
- Nastavení buzení pokoje a to i více než jednou
- Poskytování tarifikačních informací v reálném čase
- Poskytování informací o výsledku buzení pokojů

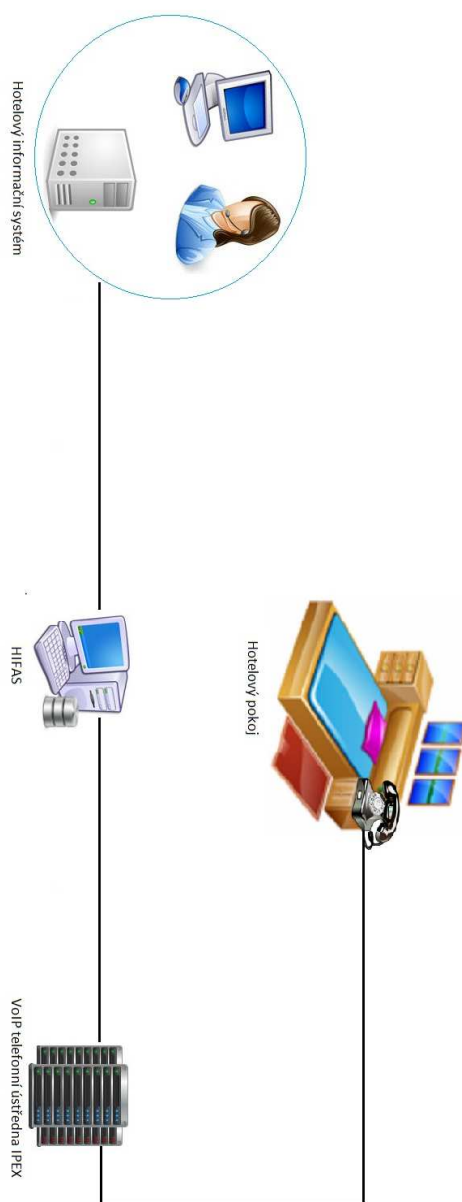
Obchodní zástupci firmy IPEX nabídli jako možnou alternativu užití webových služeb - SOAP protokolu a plnou podporu při vývoji našeho rozhraní ze strany jejich programátorů.

Byl jsem tedy ředitelem firmy dotázán, zdali bych byl schopen vývoje hotelového interface s touto telefonní ústřednou.

Tuto výzvu jsem přijal, protože jsem již několik telefonních můstků naprogramoval, či byl účasten vedení vývoje telefonních interfaců pro řetězce Orco, Genea Hotels a OREA v rámci implementací hotelového software Epitome jako IT manažer pro střední a východní Evropu.

V následujících řádcích Vás seznámím s procesem rozvahy, návrhu, vývoje,

nastalými problémy a jejich řešením a konečně s testováním telefonního interface, pracovně pojmenovaném HIFAS (hotelový interface asterisk), s dvěma různými hotelovými programy (jmenovitě Hotel2000 a Epitome).



Obrázek 1 – Blokové schéma umístění můstku HIFAS

3 Analýza a návrh řešení

3.1 Analýza používaných protokolů

Prvním bodem ve fázi plánování nového interface bylo shrnutí již používaných hotelových funkcí interfaců k telefonním ústřednám v hotelech OREA a vydefinování komunikačních protokolů. Na většině hotelů je nasazena telefonní ústředna Siemens Hicom 300 s hotelovou nadstavbou [Caracas Link](#), což je inteligentní komunikační řešení podporující hotelové služby pro telefonní ústředny Hicom, který je dle předchozích zjištění podporován převážnou většinou v současné době využívaných hotelových programů. Komunikace hotelového interface s telefonním subsystémem probíhá na bázi chráněné souborové výměny a již definovaného protokolu Caracas Link. (1)

3.2 Zjištění používaných funkcí v hotelové praxi

Druhým bodem bylo zjištění výčtu použitelných funkcí v hotelové praxi. Při srovnání nabízených funkcí dostupných přes SOAP komunikaci a původních požadavků jsme došli k rozumnému kompromisu a ustálili jsme se na výčtu následujících funkcí:

- Načtení nastavení telefonních linek do databáze dle aktuálního nastavení ústředny.
- Vyzískání informací o voláních za definovatelné období se zamezením duplicity dat.
- Vynucené volání z definované linky na libovolné číslo (v závislosti na omezeních dané linky)
- Možnost interaktivně měnit oprávnění jednotlivých linek
- Nastavení buzení s dosledováním výsledku buzení.
- Možnost zrušení nastavených buzení při příjezdu nového hosta (volitelné).
- Možnost uzamknutí/odemknutí linek bez zrušení nastavených buzení.

Po několika konzultacích s hotely jsme zjistili, že výše vydefinované funkce pokrývají požadavky středně velkých a menších hotelů bez výhrad.

Funkce servisních úkonů, například umožnění pokojové službě jednoduchým úkonem (vyvoláním aplikace ústředny zavoláním na určenou linku) dát vědět hotelovému systému o uklizení daného pokoje, potřebují kooperaci programátorů dodavatelské firmy IPEX. Podobné funkce v základní verzi programu nebudou nabízeny.

Tabulka SOAP funkcí poskytovaných telefonními ústřednami IPEX, kde označení * za názvem funkce znamená, že tato funkce je naším programem používána:

SOAP funkce pro management linek.	
createExten	Vytvoří linku.
extenInfo *	Vrací nastavení linky a SIP kanálu.
getAllExten *	Nastavení všech linek. získání pole informací o linkách.
deleteExten	Zrušení linky.
getExtenCalls	Zjištění stavu linky.
getAllExtenCalls *	Zjištění stavu všech linek.
updateExten *	Změna parametru linky.
redirectExten	Nastaví okamžité přesměrování linky lineNumber na targetNumber.
cancelRedirectionExten	Zruší přesměrování linky lineNumber.
SOAP funkce pro management volání	
originateExten *	Vyvolá hovor z fromNumber linky na lineNumber(na jakékoliv číslo).
getActualCallList	Získá seznam aktuálních hovorů, které probíhají. Pro každý hovor je nutný především unikátní kód hovoru call_id. Jde o to získat call_id pro vyvolání příposlechu. Příposlech je možné vyvolat i v případě, že neprobíhá žádný hovor. Pak je slyšet ticho do té doby, než nějaký hovor na dané lince začne. Teoreticky toto ID nebudete ani potřebovat. Stačí se omezit na číslo linky i v případě zavěšení hovoru.
actionTransfer	Předání příchozího hovoru na jiné číslo.
isRingingExten	Vrátí číslo, zvonící na lince lineNumber, když nikdo nevolá, vrátí false.
getCall	Získá informace o aktuálním hovoru call_id.
monitorExten	Nastavit příposlech. Proměnnou call_id rozumíme unikátní identifikátor každého hovoru.
SOAP funkce pro práci s výpisy	
getCDRList *	Získá výpis hovorů (pole, kde budou obsaženy všechny informace z CDR výpisu) omezený úsekem zadaných dat.
VoipexVypisHovoru *	Získá výpis uskutečněných hovorů omezený úsekem zadaných dat.
SOAP funkce pro práci s nahráváním	
setRecordingExtenOn	Nastaví, že linka bude nahrávána.
setRecordingExtenOff	Nastaví, že linka nebude nahrávána.
SOAP funkce pro práci s kontakty	
createContact	Vytvoří kontakt.
getContact	Získá informace o kontaktu.
getAllContacts	Získá informace o všech kontaktech.
deleteContact	Odstraní kontakt z adresáře.
SOAP funkce pro práci s ústřednou	
reloadConfiguration *	Znovunačtení konfigurace ústředny.

3.3 Volba vhodného databázového stroje

Třetím bodem bylo vydefinování vhodného databázového stroje pro tento program.

Po konzultaci s dodavatelem konkurenčního produktu Hotel2000 (díky dlouhodobým obchodně-osobním vztahům se společností Alto s.r.o.) jsme se shodli na faktu, že hotely používají pro své systémy serverové operační systémy Microsoft a buď již na nich SQL server běží, či je možné z hlediska licenční politiky firmy Microsoft bezplatné užití Microsoft SQL serveru 2005 edice Express. Tento program bude pracovat s Microsoft SQL serverem 2000 a 2005.

3.4 Analýza komunikace stávajícího systému Caracas-Link

Dále bylo nutné analyzovat potřebné příkazy vyměňované mezi hotelovým systémem a systémem Caracas, což nebylo složité, protože jsem měl k dispozici již funkční hotelový interface a také popis protokolu Caracas-link. Komunikace mezi oběma systémy probíhá na bázi souborové výměny. Pro komunikaci směrem od hotelového systému k ústředně se používá soubor caracas.ftp (zapisuje jej hotelový systém a čte jej Caracas-Link) a pro komunikaci od ústředny do hotelového systému se používá soubor host.ftp (zapisuje jej Caracas), tyto soubory jsou většinou umístěny na hotelovém PMS serveru, či na počítači, kde běží Caracas-Link.

Komunikace směrem do ústředny:

- nastavení buzení na linku 1202 7.9.2010 11:55 jednorázové:
031202 0000000000070910115510
- otevři linku 1101 jméno a zruší buzení na tuto linku:
011101 000000000000101BRANDT, GEORG
- otevři linku 1101 jméno bez zrušení buzení:
091101 000000000000901BRANDT, GEOR
- uzamkni linku 1804:
011804 000000000000201GYNA, TOMAS

Komunikace směrem do Hotelového interface:

- linka 1202 7.9.2010 11:55 byla vzbuzena:
171202 000000000007091011551
- poklizen - nebude se implementovat !:
131354 000000000001
- tarifikační věta linka 1835 24.10.2009 15:30:02 do 15:30:51 na číslo 481548111:
111835 000000000021100910241530021530510010835 0
00000481548111 00 00000003500

Detailní popis všech Caracas příkazů najdete v příloze (1) strany 541 až 563.

3.5 Analýza komunikace přes SOAP

Pro SOAP webové služby existuje pro Javu několik frameworků, ve škole jsme se učili o frameworku Axis2. Axis je open source frameworkem určeným pro jednoduchou implementaci SOAP zpráv v XML podobě na základě WSDL souboru.

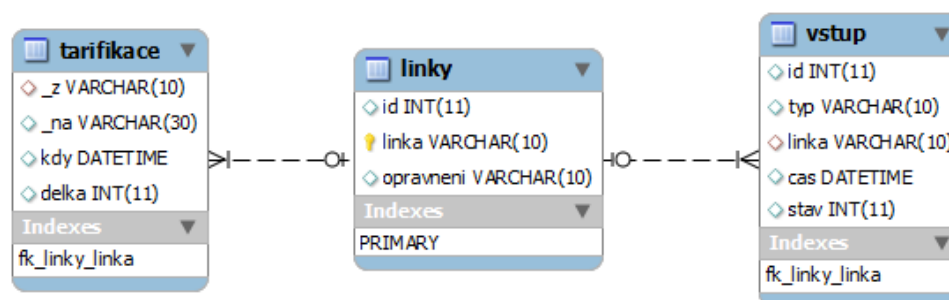
Při prvních pokusech o komunikaci jsme s panem Klimešem zjistili, že WSDL soubor obsahuje chyby, tento fakt jsme konzultovali s dodavatelem telefonní ústředny. Obdržel jsem od dodavatele ústředny potvrzení, že chyba je na jejich straně a že používají NON-WSDL řešení. Ze strany dodavatele telefonní ústředny byla slíbena brzká náprava. Bohužel po odzkoušení s upraveným WSDL souborem byl opět problém. Fakt, že WSDL soubor nekoresponduje s používanými SOAP zprávami jasně vymezil, že používání frameworku Axis2 je tímto vyloučeno.

Nemohu se spolehnout, že v další verzi ústředny bude korespondující WSDL soubor se SOAP funkcemi. Všechny vstupně-výstupní SOAP zprávy si tedy ve svém programu připravuji nezávisle na jakémkoliv frameworku pro SOAP komunikaci. Pro tuto přípravu jsem vytvořil třídu SoapFce.java.

Ačkoliv IPEX programátor přislíbil, že již WSDL soubor bude korespondovat se SOAP komunikačními větami, již jsem měl plně funkční mechanismus skládání XML vět a snad díky převládající zvědavosti jsem zvolil tu náročnější cestu. Při této volbě musím ovšem počítat s relativně častými úpravami při změně SOAP zpráv.

4 Návrh

Pro výsledný program bylo nutné vytvořit databázi, kde budou udržovány informace o telefonních linkách, telefonních hovorech, příkazech a stavu jejich řešení a definici interních příkazů. Tak vzniklo prvotní konceptuální datové schéma, které obsahuje 3 základní tabulky. Tabulka všech linek s číslem linky a odchozím oprávněním. Tabulka s informacemi o telefonních voláních tarifkace s informacemi o volajících číslech, volaných linek, délce a čase volání. Tabulka pro vstupní příkazy pro ústřednu s id požadavku, typu požadavku, linky, času zadání a stavu požadavku.



Obrázek 2 – Konceptuální datové schéma

Toto schéma však doznalo značných změn, hlavně díky zjištěním při analýze komunikace přes SOAP, přibýly tabulky, musel jsem zrušit cizí klíče a změnit strukturu tabulek.

4.1 Návrh tabulek databáze

Vznikly tak následující tabulky(entity):

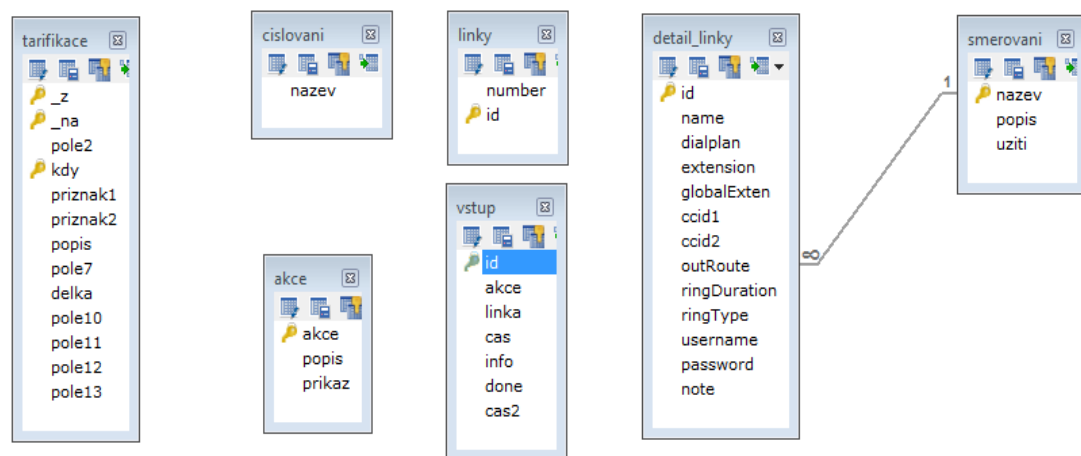
- **linky** – tato tabulka je číselníkem identifikujícím telefonní číslo linky a ID linky, používaného pro identifikaci při SOAP příkazech.
- **detail_linky** – tato tabulka obsahuje dostupné detailní informace o telefonních linkách nastavených v telefonní ústředně. Tabulka je pracovním obrazem databáze linek v telefonní ústředně. Obsahuje ID linky, číslo linky, odchozí směrování, nastavenou dobu zvonění a heslo linky. Ačkoli by se zdálo, že tato tabulka má mít vztah s tabulkou linky 1:1 a má mít nasazeno integritní omezení cizího klíče na ID linky či číslo linky, tak nebylo použito žádného vztahu, protože tabulka linky není v programu použita na nic jiného, než na prvotní iniciaci programu. Je tomu proto, že ústředna při dotazu na všechny linky umí dát výstup jen ID linky = číslo linky. Program použije tedy tabulku linky k tomu, aby pro každou linku vyslal do ústředny požadavek na vyzískání dat o jejich detailech, v dalším programu již je používána pouze tabulka detail_linky.

- **smerovani** – tato tabulka obsahuje list dostupných odchozích směrování ústředny. Zde se používá integritní omezení 1:N s tabulkou detail_linky, kde každá linka může mít jen jedno platné odchozí směrování. Tuto tabulku je nutné programem vyplnit dle nastavení telefonní ústředny před spuštěním iniciace databáze.
- **cislovani** – tato tabulka obsahuje jediný záznam a to název číslovacího plánu ústředny, který je nutný pro sestavení správné hlavičky XML dokumentu pro SOAP komunikaci s danou ústřednou. Tuto tabulku je nutné programem vyplnit dle nastavení telefonní ústředny před spuštěním iniciace databáze.
- **tarifikace** – tato tabulka obsahuje informace o odchozích telefonních hovorech s informacemi, ze které linky, kdy a kam bylo voláno. Zde nebylo použito žádných entitních vztahů, protože v případné dodatečného přidání, či ubrání linky během provozu by mohlo dojít k nekonzistenci databáze.
- **akce** – tato tabulka obsahuje interní číselník dostupných funkcí programu. Jmenovitě jde o tyto funkce:

0101	-	Check-in	(povolení odchozích volání na lince)
0102	-	Check-out	(zamezení odchozích volání na lince)
0301	-	Buzení telefonem	

Tato tabulka je využita pouze pro interní potřeby programátora, jsou v ní pouze informativní data, jaké typy akcí umí program načíst z tabulky vstup.

- **vstup** – tato tabulka je vlastně samotným jádrem programu. Zde se shromažďují veškeré příkazy určené pro obsluhu hotelových funkcí ústředny. Je zde zaznamenána veškerá komunikace s ústřednou. Obsahuje sloupce ID (ID záznamu), akce (co se má vykonat), linka (pro kterou linku), čas (hlavně pro buzení), info (informace o dané akci), done (příznak, zdali je příkaz aktivní, či je již vykonán 0/nic=čeká se na vykonání , 1=příkaz úspěšně vykonán, 2=neúspěšné vykonání příkazu, 3=probíhající buzení), čas2 (původní čas buzení). Pro posílání příkazů směrem k ústředně bude program používat právě tuto tabulku a v rozšíření pro hotelový provoz se nad touto tabulkou může napsat jednoduchý PHP skript pro ovládání buzení, případně pro uzamykání-odemykání jednotlivých linek.



Obrázek 3 – ERD diagram databáze IPEX

Na první pohled se zdá být tato databáze bez potřebných entitních vztahů, ale není tomu tak. Při iniciaci databáze, kterou může obsluha provést po přidání nových linek, či úpravě a mazání existujících linek, dochází k odmazání obsahu tabulek linky a detail_linky a údaje o historii činnosti na linkách zůstávají v tabulkách tarifikace a vstup, stejně tak i obsah tabulek číslování a směřování není nikterak ovlivněn.

4.2 Návrh funkcí programu HIFAS

4.2.1 Konfigurace spojení s SQL a telefonní ústřednou IPEX

Nejprve je nutné mít možnost nakonfigurovat spojení s SQL databází, spojení s telefonní ústřednou a další specifické vlastnosti a chování programu HIFAS. Tato nastavení jsou realizována pomocí konfiguračního souboru *soap.properties*. Jedná se o textový soubor se strukturou klíč=hodnota v UTF8 formátování, proto v něm nepoužívám diakritiku.

Soubor *soap.properties* obsahuje následující klíče a hodnoty:

Název programu, který se bude zobrazovat v popisné části programu.

napis na menu

program_name=HifAs Hotelový Interface pro Asterisk

Má-li se spustit automatické načítání ihned po spuštění programu, po dokončení ladících kroků doporučuji mít nasaveno na yes.

automaticke spusteni nactani po spusteni programu nacti_start=yes

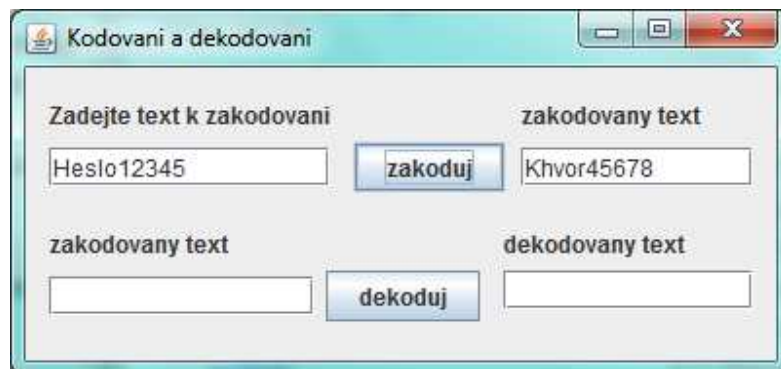
nacti_start=yes

Interval komunikace s ústřednou se nastavuje v milisekundách, osvědčilo se použití dvacetivteřinového intervalu, tedy 20000.

nastaveni intervalu v milisekundach

program_interval=20000

Důležitou částí je definice napojení na SQL databázi, bez které by program vůbec nefungoval. Zde se zastavím nad údajem `db_password`. Tento údaj je viditelný běžnému uživateli. Po krátké debatě s nadřízeným bylo ujasněno, že takovéto snadno dostupné informace musí být v zakódované podobě. Zde přicházelo v úvahu hned několik druhů šifrování. V průběhu testování jsem odzkoušel několik šifrovacích algoritmů, některé potřebovaly import dalších knihoven, proto jsem nakonec zvolil cestu nejjednodušší a vytvořil jsem si sám velice jednoduché kódování zvané Caesarova šifra. V této šifře jde o prosté posunutí písmen, kde písmeno „a“ je nahrazeno o 3 znaky následným písmenem „d“. Pro uživatele přívětivé prostředí pro toto šifrování a dešifrování byl vytvořen program *CodeDecode.jar*, jehož použití je jasně intuitivní.



Obrázek 4 – Program *CodeDecode.jar*

```
# napojeni se do databaze interface
db_connect_string=jdbc:jtds:sqlserver://ECER_NTB:1433/ipex1
db_userid=sa
db_password=ruhd
```

Definice, které směřování bude program používat pro povolení a zákaz odchozích volání. Toto nastavení musí korespondovat s nastavením telefonní ústředny.

```
# zakaz/povoleni odchoziho volani
volani_povol=MZN
volani_zakaz=INT
```

Pro definici SOAP komunikace s ústřednou je třeba nastavit následující klíče. U klíče `soap_ustredna` je možné použít jak IP adres, tak jmenného překladu adresy.

U klíče `soap_password` je použita Caesarova šifra viz `db_password`.

```
# napojeni na ustrednu - soap komunikace
soap_ustredna=192.168.224.106
soap_login=soap_app
soap_password=Khvor45678
#soap_plan=Základní davam do DB a ne do properties
```

Pro buzení je třeba mít na ústředně nastavenou aplikaci (telefonní linku), která po jejím vyvolání přehraje budící audio nahrávku.

```
# buzeni
# linka kde je nastavena hlaska pro buzeni
buzeni_linka=899
```

Pro kontrolu buzení je třeba mít na ústředně nastavenou aplikaci (telefonní linku), která po jejím vyvolání přehraje audio nahrávku s upozorněním, že nedošlo ke vzbuzení jedné či více linek.

```
# linka kde je nastavena hlaska pro oznameni neuspesneho buzeni
buzeni_upozorneni=898
```

Pro kontrolu buzení je třeba mít na ústředně nastavenou telefonní linku, kam bude

telefonicky oznámeno, že nedošlo ke vzbuzení jedné či více linek - je to obvykle linka recepce.

```
# linka kam se bude oznamovat neuspesne buzeni
```

```
buzeni_oznam=105
```

Pro ukončení pokusu o buzení dané linky se nastaví klíč buzeni_odmaz, který je v minutách. Jde o mechanismus, který zamezí cyklickému buzení linky. Vzhledem k praxi v hotelovém řetězci OREA je tento klíč nastaven na hodnotu 12 minut.

```
# odmazani requestu na buzeni po x minutach
```

```
buzeni_odmaz=12
```

Pokud nebyl host vzbuzen (není zvednuto sluchátko), ukončí se buzení standardně po uplynutí nadefinované doby v ústředně a program se pokusí vyvolat další buzení po nadefinovaném intervalu v klíči buzeni_opakovani, které je rovněž v minutách.

```
# po kolika minutach se ma znovu budit
```

```
buzeni_opakovani=3
```

Pokud chcete odmazat buzení na linkách při novém nájezdu hosta, nechte klíč buzeni_zrus_pri_checkin nasaven na yes, toto nastavení doporučuji.

```
# zda-li se ma odmazat buzeni na checkinovane lince buzeni_zrus_pri_checkin=yes
```

```
buzeni_zrus_pri_checkin=yes
```

Program má možnost náhledu, případně odmazání probíhajících a nastavených buzení. Jelikož jsme se nemohli dohodnout jaká data má program ukazovat, je toto řešeno použitím klíče buzeni_vypis_sql. Doporučuji ponechat daný sql select, který vypíše aktuálně nastavená a probíhající buzení v rozmezí 20-ti minut. Jako další příklad je zapoznámkován select, který ukazuje všechny stavy buzení v rozmezí 60 minut.

```
# sql veta pro vypis buzeni
```

```
buzeni_vypis_sql=select linka,isnull(done,'0'),id,cas2,isnull(info,'') from vstup where akce='0301' and isnull(done,'0') in('0','3') and( datediff(mi,cas2,getdate())< 20) order by cas2
```

```
#select case(isnull(done,'0')) when '1' then ' OK ' when '2' then 'ZRUSENO ' when '3' then 'PROBIHA ' when '0' then 'V CEKANI ' end as stav, linka , cas2 , cas , isnull(info,'') from vstup where akce='0301' and isnull(done,'0') in('0','2','3','1') and( datediff(mi,cas2,getdate())< 60) order by cas2 desc
```

Pro souborovou komunikaci s hotelovým telefonním interfacem je třeba nastavit přesné názvy přenosových souborů i s jejich cestami.

```
# nastaveni cesty k prenosovym souborum .ftp
```

```
hifas_file_in=C:\data\Hotel\IF-DATA\caracas.ftp
```

```
hifas_file_out=C:\data\Hotel\IF-DATA\host.ftp
```

```
hifas_file_out_safe=C:\data\Hotel\IF-DATA\host.ft#
```

Pro přeposílání kontroly každého výsledku buzení hotelovému telefonnímu interfacu je použit klíč hifas_wakeup_false_word: tento klíč je zde pro to, že některé interfacy mají nastavenou logiku: pokud nedostanou například do 15 minut potvrzení o vzbuzení, považují pokoj za nevzbuzený a sami mají funkcionalitu upozornění obsluhy, aby šla hosta vzbudit přímo na pokoj.

```
# nastaveni odpovedi pri neuspesnem buzeni (po buzeni_odmaz zapis do host.ftp)
# pro epitome musi byt =no, jinak ma byt =yes
hifas_wakeup_false_word=no
```

4.2.2 Komunikace s hotelovým telefonním rozhraním ústřednou IPEX

Pro výměnu zpráv mezi hotelovým telefonním interfacem a programem HIFAS slouží souborová výměna informací, založená na systému Caracas-Link. V našem případě simulují systém Caracas určený pro telefonní ústředny Siemens Hicom a přebírám aktuální nastavení tohoto systému, jak je používána na hotelech OREA.

Tato komunikace je založena na výměně dvou souborů: směrem od HIFAS k hotelovému systému jde o soubor *host.ftp* a směrem od hotelového systému směrem k HIFAS jde o soubor *caracas.ftp*.

Pro zajištění správné funkcionality je použit následující algoritmus:

Program při zápisu informací pro hotelový systém (jde především o informace o telefonních hovorech a výsledcích buzení) kontroluje, zda-li již neexistuje soubor *host.ftp*.

Pokud tento soubor existuje, hotelový interface ještě nenačetl a nezpracoval připravené informace: pak se nové informace zapíše na konec souboru *host.ft#*, který při neexistenci vytvoří.

Pokud ovšem soubor *host.ftp* neexistuje, program zkontroluje existenci souboru *host.ft#*: pokud takový soubor neexistuje, vytvoří se nový soubor *host.ftp*: pokud ovšem soubor *host.ft#* existuje, pak jej přejmenuje na *host.ftp* a založí nový soubor *host.ft#* s novými informacemi pro hotelový interface.

Stejný algoritmus pro souborovou výměnu používá i hotelový interface, který jen místo názvu bez přípony „host“ používá název „caracas“.

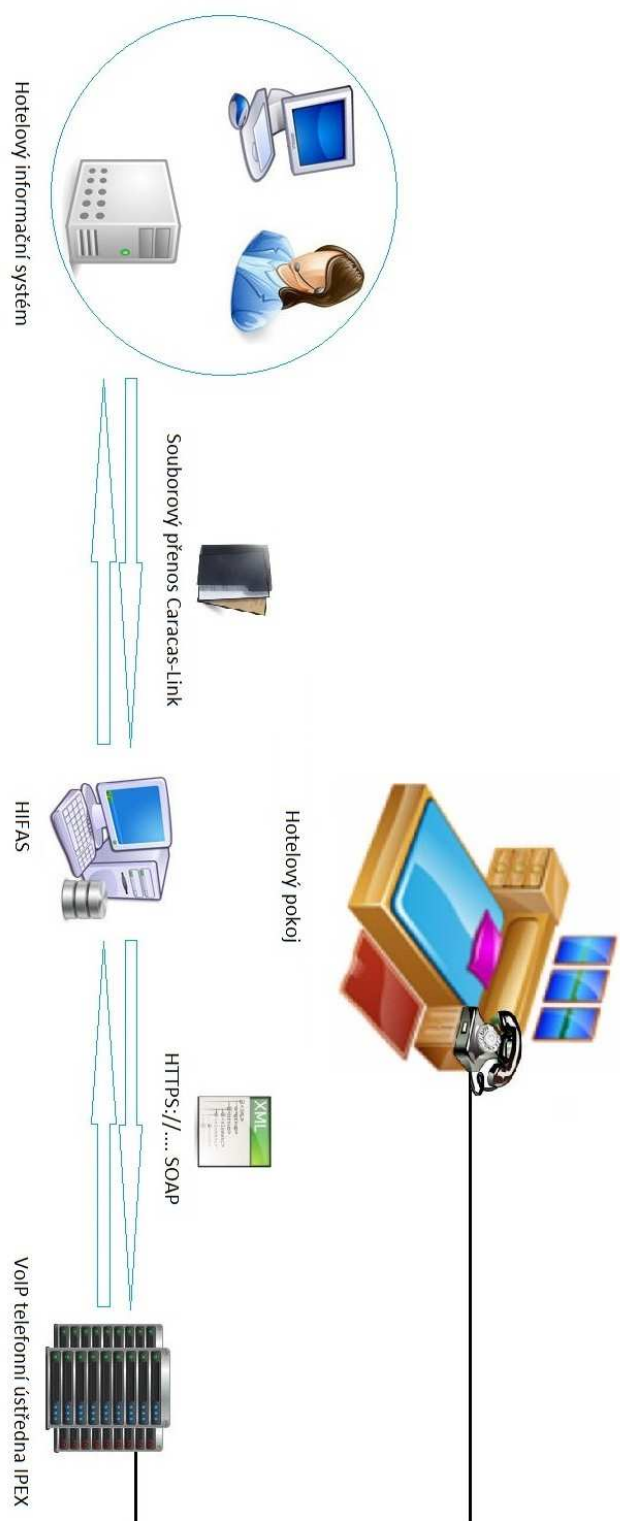
Tuto komunikaci zajišťuje třída *Caracas.java*

Pro komunikaci telefonní ústředny a programu HIFAS je používáno webových služeb přes SOAP.

Komunikace je zajištěna formou dotazu a následné odpovědi webové služby běžící na ústředně IPEX. Odpovědi i dotazy jsou XML dokumenty.

Tuto komunikaci zajišťuje třída *SoapSpust.java*

Synchronizaci v aplikaci zajišťuje překlápění hodnot logické proměnné pojmenované kývačka a použití časovače, který po každém spuštění vyhodnocuje tuto proměnnou a spustí jednu či druhou komunikaci. Konkrétně jde buď o komunikaci s hotelovým telefonním rozhraním či komunikaci s telefonní ústřednou.



Obrázek 5 – HIFAS zobrazení komunikace s použitými protokoly

4.2.3 Konfigurace vlastností pro sestavení SOAP zpráv

Při použití nové ústředny bylo zjištěno, že musí být vyvinut jednoduchý konfigurační nástroj, který umožní jednoduché napojení na již běžící telefonní ústřednu. Ne každá ústředna má nastavené stejný jmenný prostor, který je důležitou součástí SOAP komunikace. Stejně tak i odchozí směrování, která jsou nastavena v ústředně.

4.2.4 Automatické načtení linek z ústředny

Ke správné funkci programu je zapotřebí mít k dispozici seznam všech linek nadefinovaných v ústředně. Proto je třeba mít možnost automatického načtení telefonních linek a jejich vlastností do databáze programu. Tato funkcionality je určena pouze pro technickou obsluhu. Jde o jednorázovou akci, kdy si program odmaže nastavené linky z databáze a dotáže se ústředny jaké má nastavené telefonní linky a každou zvlášť načte se všemi detaily do databáze.

4.2.5 Manuální vyzískání tarifikačních údajů

Pro manuální načtení tarifikačních údajů z ústředny je třeba manuálního zadání datum - od kdy do kdy chceme získat tarifikační údaje. Program musí zabezpečit, že nedojde k duplicitním záznamům, kontroluje si tedy linku, čas volání a telefonní číslo volaného: pokud daný záznam v tabulce neexistuje, zapíše jej do ní.

4.2.6 Manuální nastavení odchozích oprávnění na jednotlivých linkách.

Tato funkcionality byla navržena pro správce telefonní ústředny k usnadnění nastavení oprávnění jednotlivých linek, případně k opravě nesprávně nastaveného směrování na telefonních linkách.

4.2.7 Možnost vynucení volání z linky na jiné číslo

Možnost vynucení volání z linky na jiné číslo v rámci oprávnění (nastaveného směrování).

Tato funkce bude použita pro nastavení buzení po telefonu či informaci recepce o nevzbuzeném hostovi a pro další, zatím jen diskutované možnosti, jako je například napojení na EPS systémy.

4.2.8 Výpis nastavených buzení

Výpis nastavených buzení - k tomuto tématu jsme diskutovali s kolegy a nakonec jsme dospěli ke kompromisu a SQL skript pro výpis nastavených buzení bude

v souboru *soap.properties*. Viz bod 4.2.1 Konfigurace spojení s SQL a telefonní ústřednou IPEX – popis souboru *soap.properties*, konkrétně jde o klíč *buzeni_vypis_sql*.

4.2.9 Odmazání vybraných záznamů buzení z databáze

Možnost odmazání vybraných záznamů buzení z databáze, tato funkcionality je určena pouze pro ladící účely při nastavování systému, bez nutnosti přímého zásahu do SQL databáze přes „Query Analyzer“.

4.2.10 Načtení přenosového souboru

Možnost vynucení jednorázového načtení přenosového souboru. Tato funkce je pouze pro servisní účely a ladění chyb.

4.2.11 Zapnutí a vypnutí automatického provozu programu

Možnost zapnutí a vypnutí automatického provozu programu. Tato funkcionality je pouze pro servisní účely a ladění chyb.

4.2.12 Ukončení programu

Při ukončení programu dojde k přednastavení vstupní tabulky u aktuálně buzených telefonních linek na příznak „čekám na zpracování“.

ShowProperty.java

Tato třída slouží pro získání informací o PC, na kterém program běží. Jde konkrétně o název PC a MAC adresu síťové karty.

DB.java

Třída určená pro komunikaci programu s SQL databází. Pro tuto komunikaci používá následující metody:

public void update(String sql) – provede sql příkaz update/delete uvedený v parametru nad připojenou databází.

public ArrayList<String> nactiId() – vrátí seznam id linek z databáze. Jde o přípravu na další práci s linkami, konkrétně na zaplnění tabulky detail_linky.

public ArrayList<Linka> nactiOpraveniLinek() – vrátí seznam všech linek z databáze - jen id, extension a outroute.

public ArrayList<String> nactiRouty() – vrátí seznam nastavených odchozích směrování z databáze.

public String nactiRouty(String id) – vrátí nastavené směrování u linky s id. Ústředny IPEX mají identifikaci všech linek a služeb přes jejich id, nikoliv přes jejich čísla.

public ArrayList<Akce> nactiVstup() – vrátí seznam nových akcí z tabulky vstup pro zpracování.

public String getdatum() – vrací aktuální čas ve formátu yyyy-MM-dd HH:mm

public int interval(int id) – vrací počet uběhlých minut od nastaveného buzení akce s id z parametru, jde o id dane akce, která je vykonávána v samostatném vlákně, tedy svazuji je přes toto id z tabulky akci vstup.

public String buzeno(int id) - navrácení řetězce pro odeslání informace o nevzbuzení - akce z tabulky vstup s id z parametru.

public boolean existujeTarifikace(String _z, String _na, String kdy) – vrací logický stav pravda v případě, že záznam v tarifikační tabulce neexistuje a může tedy do ní být zapsaná tarifikační věta: jinak vrací false a k duplicitnímu záznamu nedojde.

public ArrayList<String> kontrolaBuzeni() - vrátí seznam aktuálních buzení z tabulky vstup, které právě mají být vzbuzeny.

public ArrayList<String> select(String s) - provede sql příkaz select uvedený v parametru.

public String cislovani() – vrací z databáze název číslovacího plánu (nutné pro složení XML hlavičky SOAP zprávy)

IniciaceDB.java

Třída určená pro odmazání nastavených linek a následné načtení všech linek z ústředny IPEX do databáze, konkrétně jde o tabulky linky a detail_linky.

NactiVstup.java

Jde o výkonnou třídu, která metodou NactiVstup načte data ze vstupní tabulky SQL databáze a dostane seznam aktuálních Akcí pro telefonní ústřednu. Tyto akce vykonává metodou delej() ve vláknech aplikačního poolu programu..Po ukončení se spustí dotaz na tarifikační údaje z ústředny za posledních x minut, viz nastavení soap.properties

Buzení.java

Tato třída zajišťuje obsluhu buzení a návratu kontroly vzbuzení. Obsahuje 2 přetížené konstruktory a metodu run pro práci ve vláknech.

public Buzeni(int i, String line) – konstruktor, který očekává parametr id z tabulky vstup a kterou linku má vzbudit. Linku pro buzení bere z nastavení v soap.properties

public Buzeni(String line, String target) – konstruktor, který očekává linku, z které se bude volat a linku na kterou se bude volat, kde cílová linka může být libovolné telefonní číslo.

public void run() – metoda, která poběží v novém vláknu. Vytočí linku pro buzení a za minutu vrátí zprávu o výsledku, či přenastaví čas příštího buzení o x minut dle soap.properties souboru.

Dále zde uvedu třídy určené pro komunikaci s telefonní ústřednou IPEX.

MySSLSocketFactory.java

Jde o staženou třídu z webu Apache, díky níž je navázána SSL zabezpečená komunikace s telefonní ústřednou IPEX. Pro tuto komunikaci musíme mít od výrobce poskytnutý certifikát v souboru jssecacerts umístěném v rootu projektu.

SoapFce.java

Tato třída skládá XML dotazy pro SOAP komunikaci s telefonní ústřednou IPEX. Obsahuje následující metody:

public String xlDatum(String d) – vrací textovou reprezentaci datumu zadaného v parametru (YYYYMMDD) ve formátu, který je očekáván pro SOAP komunikaci.

public String vypisHovoru() – vrací XML dotaz, který zajistí vypsání tarifikačních údajů z ústředny IPEX za posledních 5 minut. Je nutné dbát na to, aby jak PC, na kterém běží program HIFAS, tak ústředna i hotelové počítače měly nastaven stejný NTP server, aby zde nedocházelo k chybám vzniklých časovým posunem mezi systémy.

public String vypisHovoru(String linka, String _od, String _do) – vrací XML dotaz, který zajistí vypsání tarifikačních údajů pro danou linku, nebo vložíme-li prázdný řetězec jako linku, tak pro všechny linky za určené období _od _do, kde datumy očekává ve formátu YYYYMMDD.

public String vytocLinku(String _z, String _na) - vrací XML dotaz, který zajistí vyvolání telefonního hovoru z jedné linky na druhou. Typicky používané pro buzení.

public String vypisLinek() - vrací XML dotaz, který zajistí výpis všech nastavených linek v ústředně .(id linky a číslo linky)

public String vypisNastaveniLinky(String id) – vrací XML dotaz, který zajistí výpis detailů jedné telefonní linky dle jejího id předaného v parametru.

public String updateLinky(String id, String plan, String popis) – vrací XML dotaz, který zajistí úpravu odchozího oprávnění a popisu dané linky v ústředně. Zde se opět používá id linky a ne číslo linky.

public String updateLinky(String linka, String plan) – vrací XML dotaz, který zajistí úpravu odchozího oprávnění dané linky.

public String vsechnyVolani() – vrací XML dotaz, který zajistí výpis všech aktuálně probíhajících volání na ústředně IPEX. Tato metoda je připravena k dalším funkcím, v tomto programu není použita.

public String getExtenInfo(String linka) – vrací XML dotaz pro výpis nastavení jedné linky. Tentokrát používá číslo linky.

public String getCDRlist(String line) – vrací XML dotaz, který zajistí informaci, byla-li daná linka vzbuzena či nikoliv.

public String potvrd() – vrací XML dotaz, který zajistí aktivace změn odeslaných do ústředny.

public String readProperties() – vrací název PC a MAC adresu počítače v desítkovém tvaru.

SoapSpust.java

Třída, jež odesílá SOAP zprávy telefonní ústředně. Tato třída má 2 konstruktory jeden bezparametrický a druhý jednoparametrický, který rovnou očekává XML dotaz. Další metody:

public void setID(String s) – nastaví id záznamu

public void soapKomunikace(String request) – spustí komunikaci s ústřednou a předá jí dotaz, který dostane v parametru, odpověď zpracovává třída LinkyHandler

LinkyHandler.java

Jedná se o SAX parser, který zpracovává odpovědi od telefonní ústředny a zaplní s nimi proměnné.

SoapSpust1.java

Třída, jež odesílá SOAP zprávy telefonní ústředně výhradně jen dotazy, zdali byla daná linka vzbuzena, jelikož se názvy elementů v odpovědi ústředny mixují, nelze použít jeden SAX XML parser, odpověď zpracovává třída LinkyHandler1.

LinkyHandler1.java

Jedná se o SAX parser, který zpracovává odpovědi od telefonní ústředny a zaplní s nimi proměnné: tento konkrétně obsluhuje pouze odpovědi o vzbuzení, či nevzbuzení.

Dále tu máme třídu pro komunikaci s hotelovým telefonním interfacem.

Caracas.java

Tato třída zabezpečuje souborovou komunikaci mezi hotelovým telefonním interfacem a programem HIFAS na bázi Caracas-Link protokolu. Tato třída obsahuje následující metody:

public String vzbuzeno(String linka, String cas, boolean jak) – vrací Caracas větu s výsledkem buzení linky v daný čas s výsledkem buzení, podle zadaných parametrů metody.

public String tatifikace(String linka, String cil, String cas, int delka, String trunk) – vrací Caracas tarifikační větu pro hotelový interface, která linka volala kam, kdy, jak dlouho a přes jaký trunk, podle zadaných parametrů metody.

public void zapisHost(String s) – tato metoda obsluhuje zabezpečený souborový zápis směrem k hotelovému telefonnímu interface.

public void kontroluj(String nacteno) – tato metoda podle načtené věty ze souboru přicházejícího od hotelového telefonního rozhraní zjistí co se má vykonat a provede zápis do SQL tabulky vstup.

public void nacti() – metoda pro načtení souboru Caracas.ftp , kde jednotlivé řádky předá metodě kontroluj. Po načtení tento soubor odmaže.

Dále zde uvedu třídy zabezpečující komunikaci s operátorem – instalátorem.

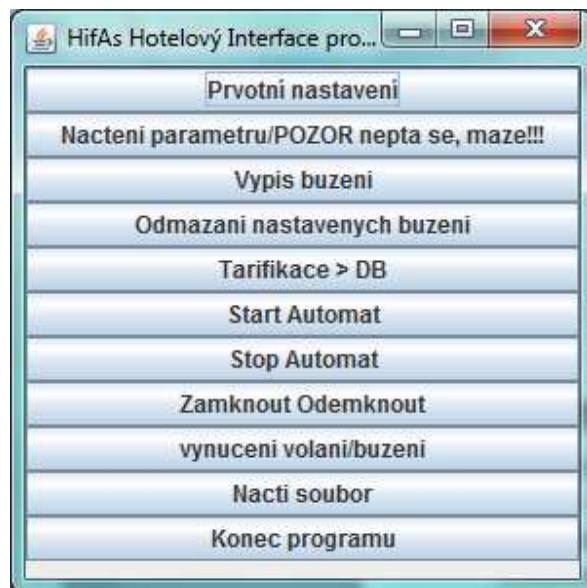
Menu.java

Jedná se o hlavní třídu programu, která načte nastavení programu do svých proměnných ze souboru soap.properties a podle nich se bude program chovat. Je to jedno z několika z grafických rozhraní programu. Tato konkrétně umožní operátorovi provést operace vydefinované v bodech 4.2.4 až 4.2.12

Založí jedno vlákno pro FIFO frontu na komunikaci s hotelovým interfacem určenou pro souborový zápis. Dále založí pool spustitelných vláken pro spouštění několika úloh v jednom okamžiku, zde například jde o využití spuštění buzení více linek najednou, kde si každé vlákno hlídá svou komunikaci s telefonní ústřednou.

public void nacti() - metoda která spustí načtení čekajících příkazů pro telefonní ústřednu a jejich vykonání.

Public static void main(String Args) – Nejprve zkontroluje název a MAC adresu počítače, kterou pomocí Caesar šifry zakóduje a výsledný řetězec porovná se souborem serial.ser a pokud se tyto shodují, vytvoří jednoduchou tlačítkovou lištu určenou pro servisní účely například jako je manuální nastavení oprávnění linek či vypnutí / zapnutí automatu.



Obrázek 7 – Hlavní nabídka programu HIFAS

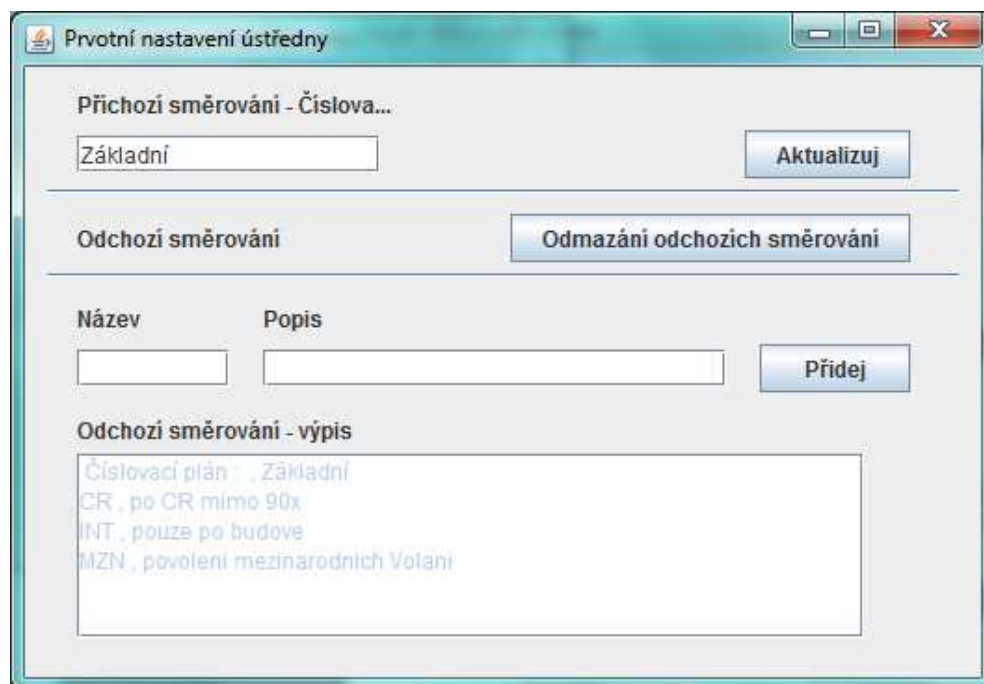
NewPBX.java (Prvotní nastavení)

Jak již bylo zmíněno, nelze informace o názvu číslovacího plánu pro SOAP obálky a v ústředně nastavených odchozích směrování získat automaticky. Pro toto ruční zadání byla sestavena tato grafická třída jako jedno z grafických rozhraní. Tato konkrétně umožní instalátorovi zadat údaje, bez kterých nemůže programu navázat úspěšnou SOAP komunikaci s ústřednou.

Zčásti je sestavena NetBeans grafickým editorem a do posluchačů tlačítek byl vložen vlastní kód pro editaci záznamů sql tabulek číslování a směrování.

public void delej() - zaplní textové pole této komponenty nastavením z SQL tabulek číslování a směrování.

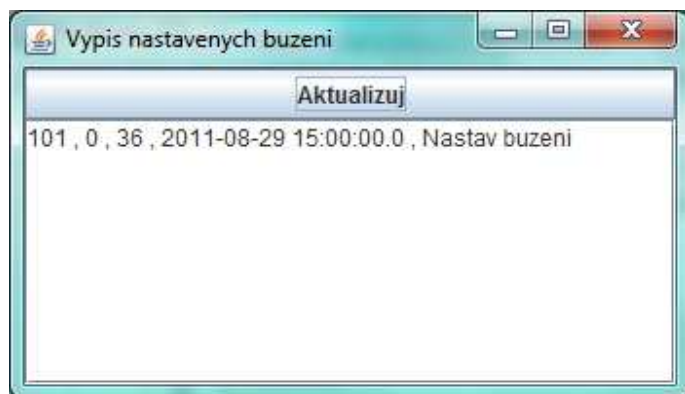
public void vykresli() – načte data a aktualizuje hodnoty grafických komponent.



Obrázek 8 Grafická komponenta NewPBX

Fronta.java (Výpis buzení)

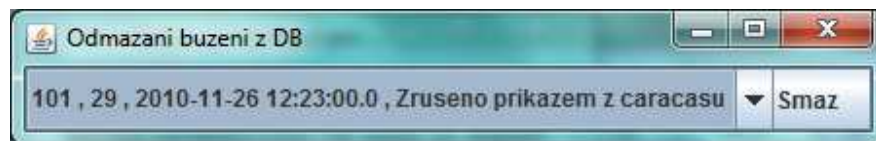
Tato grafická třída je další z grafických rozhraní. Konkrétně zobrazuje výpis aktuálně nastavených buzení.



Obrázek 9 – Výpis nastavených buzení

MazBuzeni.java (Odmazání nastavených buzení)

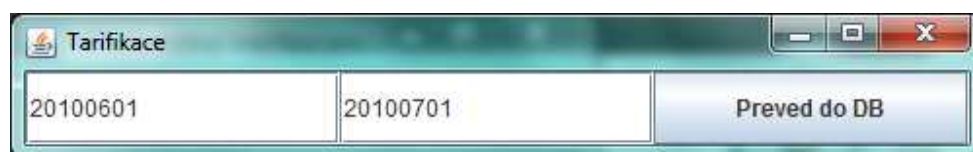
Tato grafická třída je další z grafických rozhraní. Konkrétně umožní operátorovi odmazat nastavená buzení. Tato třída je určena primárně pro servisní účely.



Obrázek 10 – Odmazání nastavených buzení

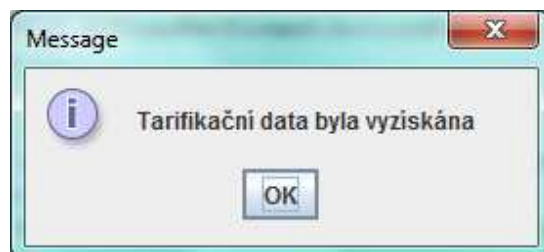
Tarifikace.java (Tarifikace > DB)

Tato grafická třída je další z grafických rozhraní. Konkrétně umožní operátorovi vynutit tarifikační výstup z IPEX ústředny do SQL databáze za zadaný interval dnů. Formát datumů se zadává ve formátu YYYYMMDD či jiném , kterému rozumí MS SQL stroj.



Obrázek 11 – Vynucení tarifikačních vět za dané období

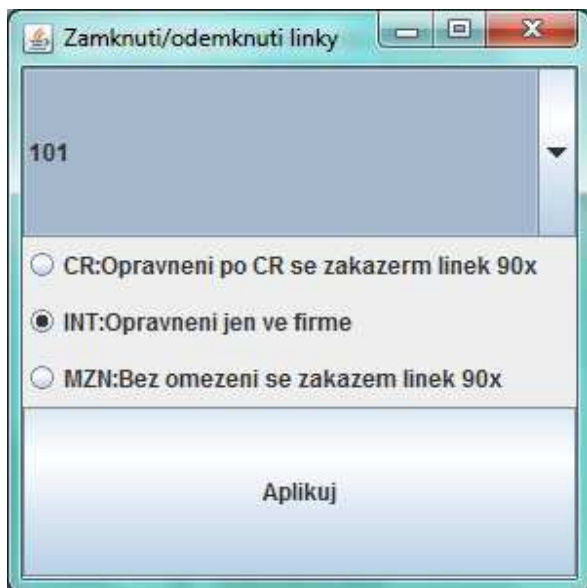
Po stisku tlačítka „Preved do DB“ dojde ke komunikaci s ústřednou a ta odešle vyžádaná data do SQL databáze. Po skončení se zobrazí zpráva:



Obrázek 12 – hlášení o ukončení načítání tarifikačních vět

OutRouteGui.java (Zamkni Odemkni)

Tato grafická třída je další z grafických rozhraní. Konkrétně načte všechny linky z databáze a umožní operátorovi nastavit odchozí routy jednotlivým linkám ústředny, tedy oprávnění k volání z dané linky.



Obrázek 13 – Změna odchozích práv na linkách

Zde si vyberete linku a objeví se její aktuální nastavení, které může operátor změnit.

Volani.java (Vynucení volání/buzení)

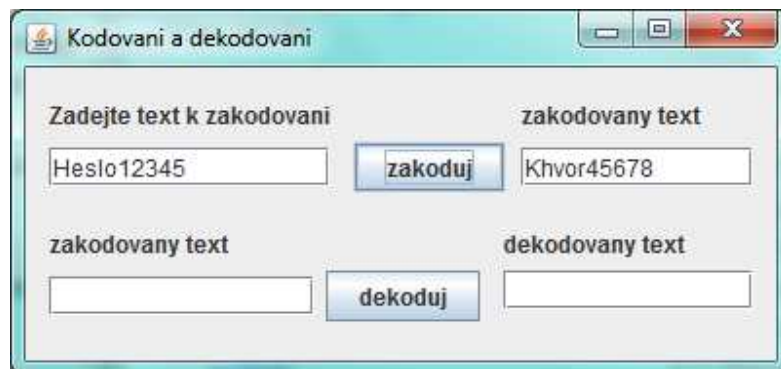
Tato grafická třída je další z grafických rozhraní. Konkrétně dovolí vynutit volání z jedné linky na jinou linku, či na telefonní číslo v rámci oprávnění dané linky. Linku, z které se bude volat si může obsluha vybrat z nabídky.



Obrázek 14 – Vynucení volání z linky na jinou linku

SifraGui.java

Tato grafická třída je další z grafických rozhraní. Konkrétně umožní instalátorovi jednoduše zašifrovat a dešifrovat text. Zčásti je sestavena NetBeans grafickým editorem a do posluchačů tlačítek byl vložen kód pro volání metod třídy Caesar.



Obrázek 15 – Kódování a dekódování textu

4.4 Typický cyklus práce programu

Prvním krokem je zprovoznění potřebných nastavení na straně telefonní ústředny a nastavení parametrů programu: toto nastavení je popsáno v příloze *PostupZprovozneniNoveUstredny.doc(6)*: toto nastavení se provede pouze na poprvé.

Program při správném nastavení pracuje samostatně a není prvotně určen pro uživatelskou obsluhu. Nicméně obsahuje nezbytné grafické uživatelské rozhraní nutné k řešení většiny známých potíží.

Po spuštění programu se automaticky v nastavených intervalech snaží shromažďovat vstupy od hotelového interface, ty zapisuje do tabulky vstup. Při následujícím průchodu načítá informace z tabulky vstup a vykonává SOAP příkazy na ústředně dle zadání a dle odpovědi od ústředny upravuje výsledky jednotlivých operací ve vstupní tabulce.

Tyto dva stavy se neustále opakují, dokud obsluha nepřeruší tento automatický cyklus tlačítkem „Stop automat“. Při řešení případných problémů doporučuji toto udělat jako první věc, dále stiskem tlačítka „Nacti soubor“ dojde k načtení dat ze souboru od hotelového systému, které zapíše do vstupní tabulky, kde si data lze prohlédnout v závislosti na typu dat - buď přímo v SQL analyzáru, nebo v programu stiskem tlačítka „Vypis buzení“. Po zjištění a odstranění závady opětovně spustíte automatickou komunikaci mezi hotelovým systémem a ústřednou stiskem tlačítka „Start automat“.

5 Vývoj

5.1 Výběr vývojového prostředí

Jako vývojové prostředí pro program HIFAS jsem použil NetBeans 6.x a programovací jazyk Java.

Pro tento program bylo třeba stáhnout několik knihoven, konkrétně sqldrivers pro MS SQL server, dále httpcomponents-client-4.0.1 pro http komunikaci a pro logování log4j.

5.2 Testovací prostředí

Pro testování jsem dostal IP telefonní přístroj AASTRA 6731i, kterými je vybavena jedna naše administrativní budova v ulici Vyskočilova na Praze 4 a byla pro mne založena telefonní linka 997. Při dalších testech jsem užíval také softwarové telefonní klienty 3CX a X-Lite a pro testování v pozdější fázi nám byla zapůjčena telefonní ústředna IPEX, která je umístěna v racku naší firemní serverové místnosti a je dosažitelná jen přes VPN a softwarové klienty.

5.2 Vystalé problémy

Prvotním problémem byl fakt, že autentizace s telefonní ústřednou probíhá na bázi http basic autentizace. Komunikace je šifrovaná na protokolu SSL. Musel jsem tedy pátrat na Internetu, než jsem našel postup, aby program považoval každý certifikát za platný. Tento krok jsem konzultoval s panem Klimešem. Nakonec používám standardní řešení pro SSL komunikaci, kde je využíván soubor *jssecacerts* umístěný přímo v pracovním adresáři projektu programu.

Jak jsem již popsal v bodu 3.5, jedním z hlavních mezníků byl fakt, že ústředna používá non-WSDL řešení.

Tedy fakt, že WSDL soubor nekoresponduje s používanými SOAP zprávami vedl k závěru, že používání frameworku Axis2 je tímto vyloučeno - nemohu se spolehnout, že v dalším upgrade verze ústředny bude korespondující WSDL soubor se SOAP funkcemi. Všechny vstupně-výstupní SOAP zprávy rozděluji a připravuji nezávisle na jakémkoliv frameworku pro SOAP komunikaci.

Ačkoliv IPEX programátor přislíbil, že již WSDL soubor bude korespondovat se SOAP komunikačními větami, již jsem měl plně funkční mechanismus skládání XML vět.

Jedním z klíčových problémů bylo to, že ústředna sice dostala příkaz ke změně odchozího směřování, ale tyto změny musely být manuálně provedeny přes webové rozhraní ústředny. S programátory jsme se domluvili, že pro aplikaci změn v ústředně přibude další SOAP příkaz a tato nová verze software byla nasazena koncem července 2010. Ihned po nasazení upgrade software ústředny,

když byly přidány požadované nové funkce pro reload ústředny, se vyskytly potíže, například tarifkace již nemá 11 polí, nýbrž 13 a podobně.

Při odladování HIFAS interface jsem zjistil, že neustálou SOAP komunikací zahlcují paměť IPEX telefonní ústředny.

Systémové informace

Hostname:	vyskocilova.ipex.cz
Uptime:	67d 16h 12m 8s
Systémový čas:	11:54:52, 06. 10. 2010
Verze:	2.2.4
Sériové číslo:	10-010-500-VX

Informace o serveru

Průměrná zátěž  0%

Operační paměť  2943/3032MB

Odkladací prostor  0/964MB

Souborové systémy

/ (ext3)  10/292GB

/dev/shm (tmpfs)  0/1GB

Informace o ústředně

Aktivní hovory: 3
 Počet SIP zařízení: 232
 Zaregistrovaná zařízení: 225
 Počet SIP trunků: 2
 Registrované SIP trunky: 1
 Počet IAX trunků: 0
 Registrované IAX trunky: 0

Zařízení

Síťové zařízení eth0
 Příchozí provoz: 1130824782B
 Odchozí provoz: 615682844B
 Síťové zařízení eth1
 Příchozí provoz: 167086702B
 Odchozí provoz: 3655757311B
 Síťové zařízení eth1.11
 Příchozí provoz: 3637274707B
 Odchozí provoz: 817187820B
 Síťové zařízení eth1.12
 Příchozí provoz: 18722B
 Odchozí provoz: 17814B

Obrázek 16 - zátěž ústředny při testování

Bylo tedy nutné upravit frekvenci SOAP komunikace na desítky vteřin. Optimální nastavení je okolo 20 vteřin (méně než 10 vteřin zatěžuje ústřednu a více než 30 vteřin včetně může spustit buzení se zpožděním). O těchto problémech ústředny jsem informoval programátory od IPEXu a ti podnikli kroky ke snížení vytížení úpravou software ústředny.

Zajímavým tématem se stala otázka tarifkace, kterou ústředna standardně zpracovává jednou měsíčně dávkovým systémem. Musel jsem tedy řešit, jak často a za jaký časový úsek nechat tarifkaci od ústředny získávat, stejně tak zabezpečit ochranu proti vzniku dluhů.

Zde jsem nejprve pro zamezení duplicit užil složeného primárního klíče a později jsem připravil metodu „existujeTarifikace“ ve třídě DB s parametry linky, ze které bylo voláno, volaným číslem a dobou volání. Tato metoda vrací boolean hodnotu: v případě, že již záznam existuje, vrátí false a v případě, že záznam ještě není v databázi, vrátí true.

Při testování s více telefony bylo zjištěno, že je nutné přepracovat program na více vláknový, tedy jak pro SOAP komunikaci, tak pro souborovou komunikaci.[22]

V hlavní třídě tedy definuji 2 různě definice chování volaných vláken.

```
public static ExecutorService app = Executors.newCachedThreadPool();
```

Tato definice app definuje paralelní spouštění vláken a je používána při volání požadavků na SOAP komunikaci, například pro iniciaci buzení několika linek ve stejný čas.

```
public static ExecutorService single = Executors.newSingleThreadScheduledExecutor();
```

Naopak tato definice single definuje frontu událostí, které se vykonají jedna po druhé a je používána pro zápis výsledků jednotlivých vláken a pro synchronizaci zápisů pro souborovou komunikaci s hotelovým systémem.

Další problém, se kterým jsem se musel vypořádat je fakt, že se v odpovědích od ústředny vyskytují shodné názvy elementů s jinými typy odpovědí. Proto jsem požádal IPEX o změnu jmenného prostoru odpovědi CDR(jde o výsledek volání na aplikaci, v našem případě jde o odpověď, zda-li byl, či nebyl zvednut telefonní přístroj při buzení telefonem). Proto byl vytvořen další SAX(Simple API for XML) parser LinkyHandler1, který je použit pouze na zhodnocení výsledku buzení telefonem.

Pro souborovou komunikaci s hotelovým telefonním rozhraním bylo nutné ošetření prepisů informací. Tuto problematiku jsem již popsal v bodě 4.2.2. Proto poukážu pouze na princip ošetření souborové komunikace. Jedná se o kontrolu existence souboru pro zápis *host.ftp*. Pokud existuje, jsme ve stavu, kdy si hotelový interface ještě soubor nenačetl a nesmazal, další příkaz se zapíše na konec souboru *host.ft#*, případně tento soubor založí a pokud existuje jen *host.ft#*, přejmenuje jej na *host.ftp* a založí *host.ft#*, do kterého zapíše.

Pro komplexnější zajištění bezpečného přenosu je tato metoda volána přes volání statické fronty single. Nedojde tedy ke sdílení prostředků více vláken, nýbrž k zařazení požadavku na konec fronty.

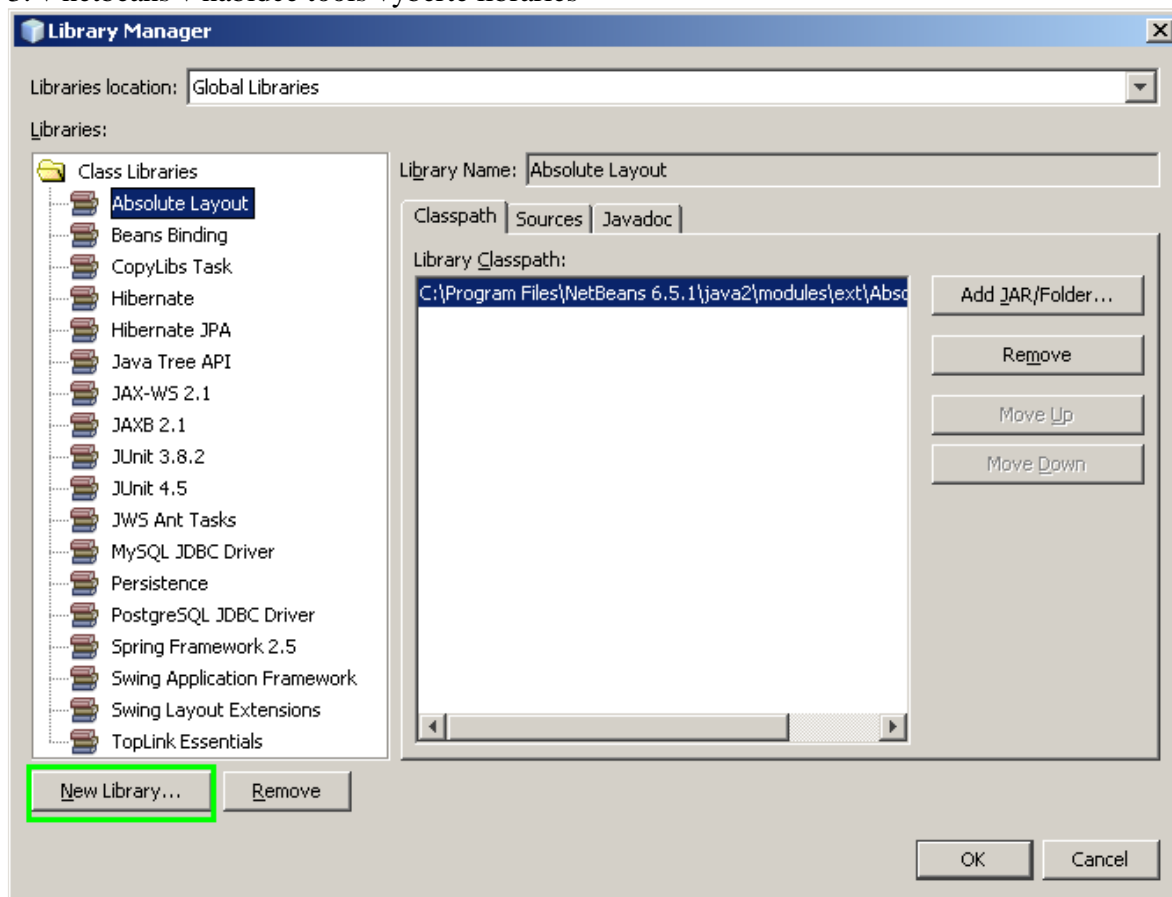
5.3 Zprovoznění nového testovacího prostředí

Obdržel jsem zapůjčenou telefonní ústřednu a nainstaloval jsem NetBeans a MS SQL 2005 Server Express na virtuální stroj speciálně připravený pro mé potřeby.

Popíši zde nezbytné kroky nutné ke zprovoznění ústředny a projektu v prostředí NetBeans.

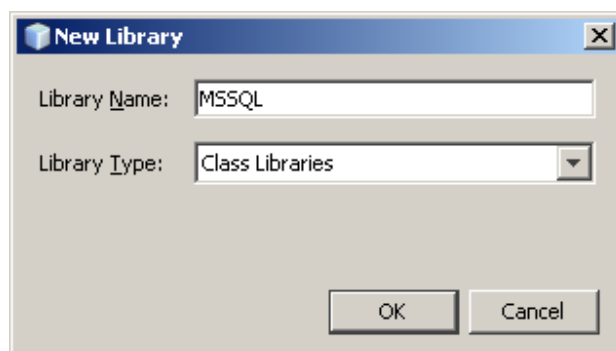
5.3.1 Postup otevření projektu v Netbeans

1. rozbalit projekt a knihovny sqldriver , log4j a httpcomponents-client-4.0.1
2. otevřít projekt SOAP/App - při otevření projektu budete upozorněni na neexistenci potřebných objektů.
3. v netbeans v nabídce tools vyberte libraries



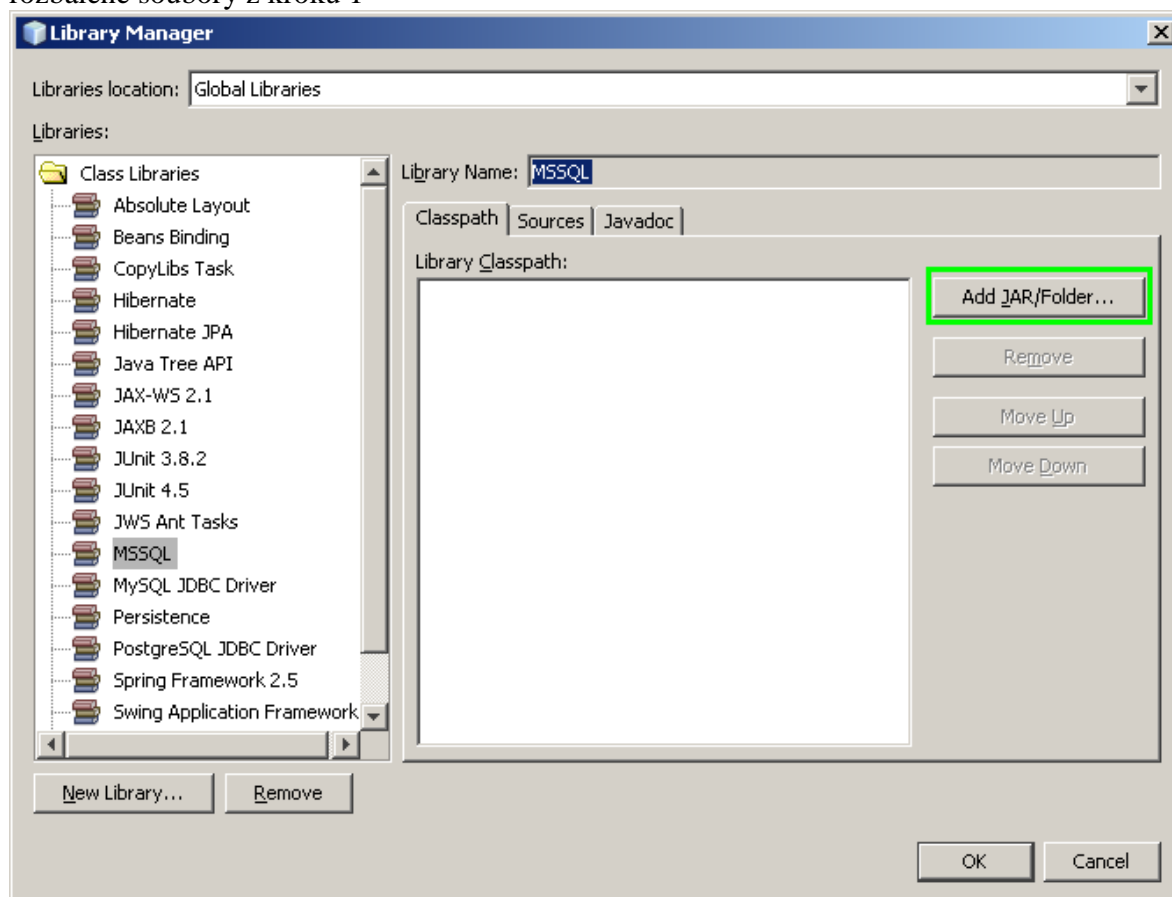
Obrázek 17 - seznam knihoven v NetBeans

Stiskněte tlačítko New Library , zadejte odpovídající název, v našem případě MSSQL

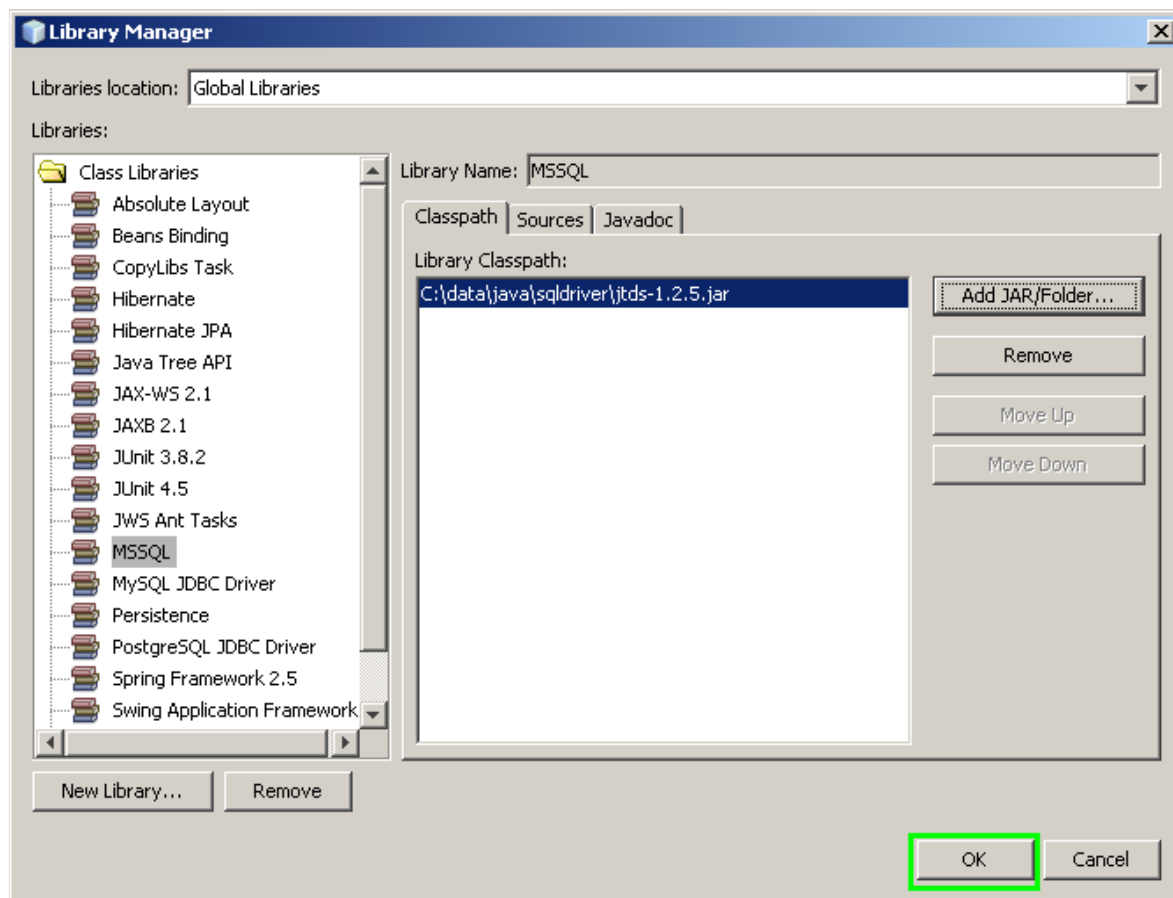


Obrázek 18 - založení nové knihovny

Stiskem OK se dostanete do výběru jednotlivých jar souborů. Zde nalistujte rozbalené soubory z kroku 1

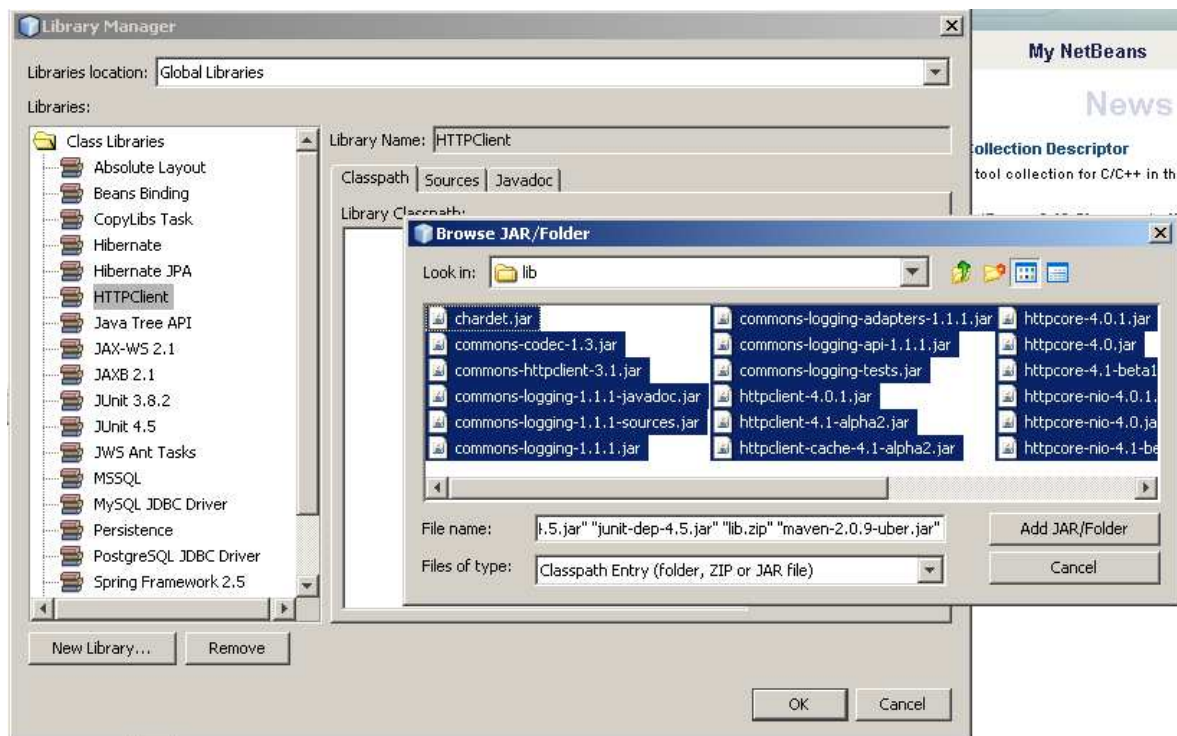


Obrázek 19 - přidání JAR souborů



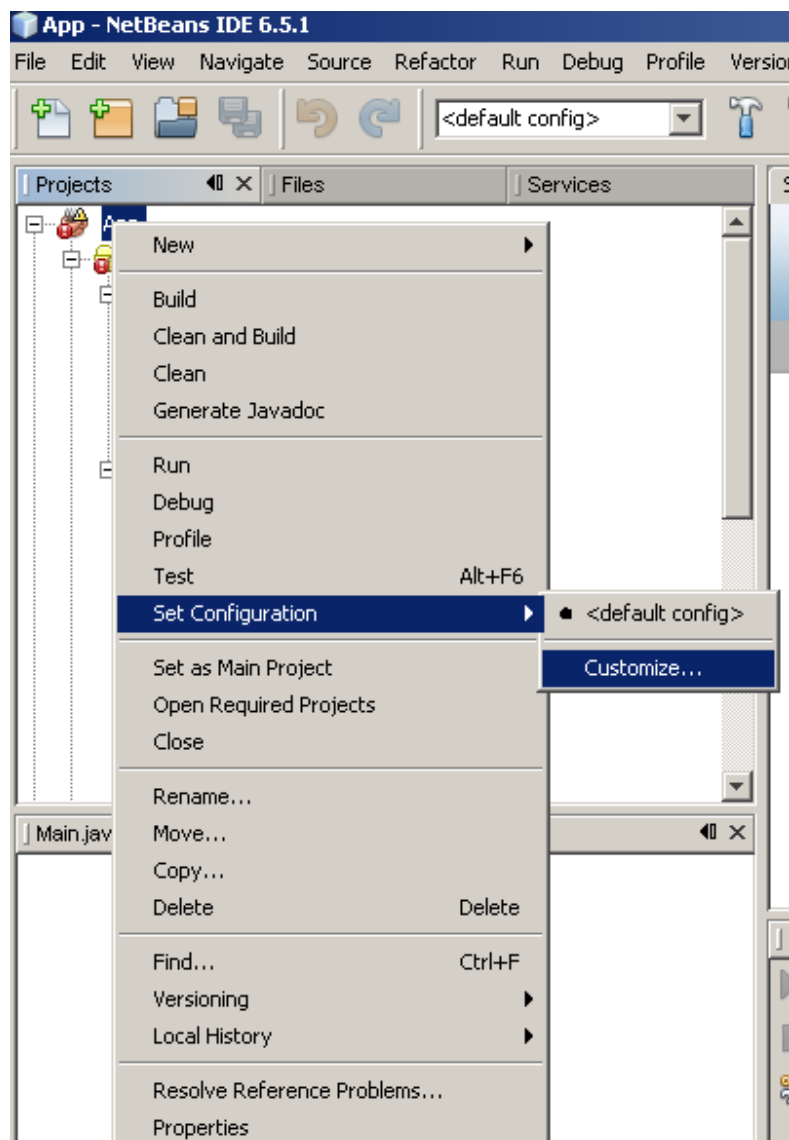
Obrázek 20 - dokončení přidání knihovny

Stiskem tlačítka OK jsme přidali knihovnu pro MSSQL. Nyní opakujeme ty samé kroky pro vytvoření knihovny pro Log4j a pro http klienta. U HTTP klienta při přidávání jar souborů vyberte všechny soubory v podadresáři lib.

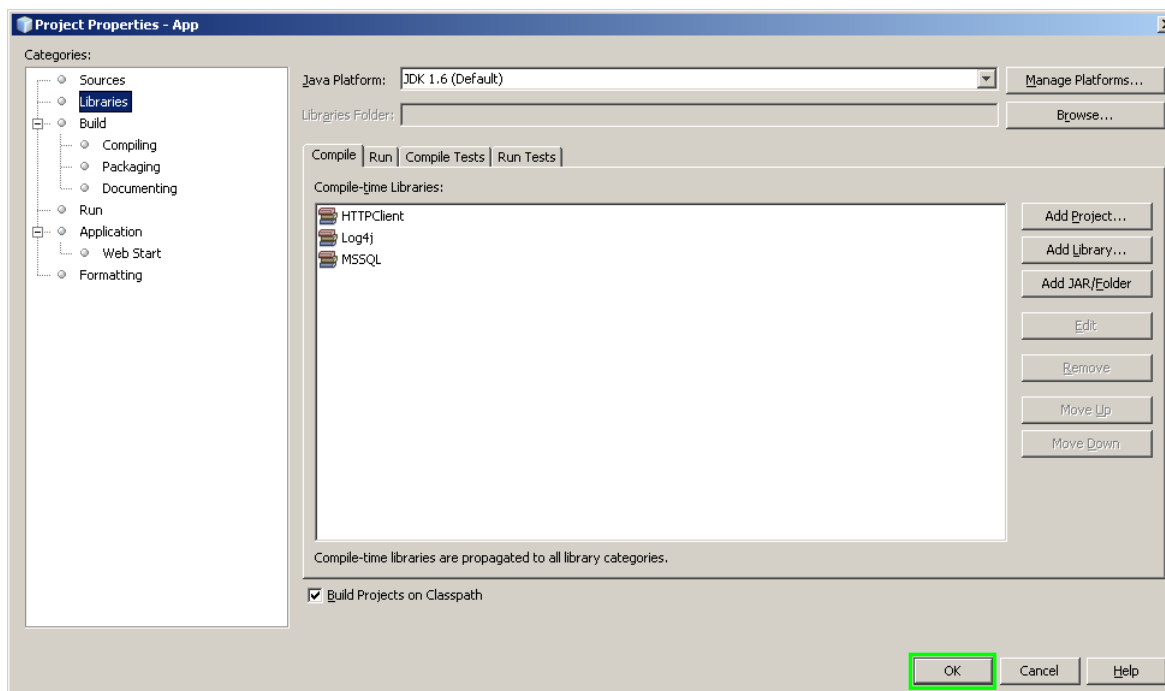


Obrázek 21 - přidání více JAR souborů najednou

4. v nastavení projektu přidejte obě tyto knihovny do projektu



Obrázek 22 - přizpůsobení konfigurace projektu



Obrázek 23 - seznam potřebných knihoven

Pokud jsme se netrefili do přesného názvu knihoven, tak je odmažeme a přidáme dle naší definice. Po stisku tlačítka OK dojde k uložení projektu a zmizí výstražná znaménka u projektu a jeho zdrojových souborů.

6 Odzkoušení funkčnosti s PMS systémy

Díky nastolené spolupráci s Alto Praha s.r.o. (dodavatel restauračního software a pokladen v hotelech OREA, který mimo jiné také dodává hotelový systém Hotel2000) bylo dohodnuto s vedením této společnosti a implementátory tohoto hotelového systému, že nám připraví instalaci na mém pracovním počítači u nás ve firmě. Dostal jsem za úkol kontaktovat specialistu Ing. Ondřeje Pekárka a dohodnout s ním podrobnosti.

Díky způsobeným problémům s testováním komunikace mezi mým testovacím PMS programem Epitome na ostrém provozu telefonní ústředny v administrativní budově na ulici Vyskočilova (zatížení ústředny obr. 16), jsme se s firmou IPEX dohodli o zapůjčení telefonní ústředny jen pro má testování.

Program HIFAS byl úspěšně odzkoušen s hotelovým systémem Epitome. Vše proběhlo bez větších problémů, protože mám jako bývalý zaměstnanec dodavatele tohoto PMS systému expertní znalosti s tímto systémem. Zároveň jsem s tímto hotelovým systémem prováděl veškeré nastavování a vývoj tohoto programu.

Program HIFAS byl v polovině listopadu roku 2010 úspěšně odzkoušen s hotelovým systémem Hotel2000. Na základě testování, byl vypracován akceptační protokol. Toto testování probíhalo formou telefonního hovoru s interaktivním propojením přes software pro vzdálený přístup a podporu přes Internet TeamViewer [7].

Pro tato testování byl připraven testovací scénář, který simuluje provoz hotelu (5).

Akceptační protokol Hifas Interface a Hotel2000
10.11. 2010

aiteco

Popis	Hodnocení	Komentář
Testy zaměřené na funkčnost hotelových funkcí		
1. Test check-in/out, uzamčení, odmítnutí advanced check-in)		
Kontrola stavu před testem	ok	Kontrola přes webové rozhraní ústředny
Přihlášení hosta(2x) v Hotel2000 (101,103)	ok	Kontrola přes webové rozhraní ústředny
Uzamčení/odmítnutí linky interfacem(check-out, advanced check-in)	ok	Kontrola přes webové rozhraní ústředny
Odhlášení hostů z Hotel2000	ok	Kontrola přes webové rozhraní ústředny
2. Test buzení - nastavení, rušení, rušení check-inem, tarifkace		
Přihlášení hosta(2x) v Hotel2000 (101,103)	ok	Kontrola přes webové rozhraní ústředny
Nastavení buzení hromadně	ok	Vše se udržuje v SQL DB programu a na každou jednotlivou linku je v době buzení využito jedno vlákno, jsme omezení jen počtem současných volání do typu ústředny
Jeden telefon zvednout, ostatní ne	ok	Jedno buzení se přeložilo do stavu vzbuzeno
Opakované buzení - jeden telefon zvednout ostatní ne	ok	
Nastavení buzení linky 103	ok	
Zrušení buzení na lince 103 přes Hotel2000 interface	ok	
Odhlášení hostů ze 103,101	ok	
Přihlášení hostů na 103 = očekávané zrušení buzení	ok	
Kontrola tarifkace v Hotel2000	ok	Záleží na nastavení die ini souboru, funguje
3. simulace výpadku interface, načtení zmeškaných tarifkací		
Výpustí interface Hifas	ok	Interface je koncipován na off-line provoz, vše funguje
provedení několika telefonátů (pex-pex, pex-mobil)	ok	Jsou vidět tarifkační údaje na Hotel2000 IPC
vytvoření několika minutů automaticky bere minutu zpětně	ok	ok, pokud by byl itc delší dobu mimo provoz, je dobré načíst tarifkace za datum zpětně a dopředu, nedojde k duplikaci tarifkačních vět, zaplášeno Hifas interfacem
spuštění Hifas a načtení tarifkace za datum od - do (simulace stavu, kdy vám hotel oznámí, v dle netarifkovali)	ok	je možné začít datum ve formátu RRRRMMDD jak od, tak do
Kontrola načtení v Hotel2000	ok	Tarifkace je v hotelovém programu dle očekávání
4. prvotní nastavení systému, popis ini souboru, diskuse		
Kontrola ze strany ústředny, co je třeba znát a mít nastavené na ústředně	ok	viz příložený dokument
nastavení iv souboru pro napojení na ústřednu	ok	
nastavení DB	ok	
inace DB SQL skriptem	ok	
Nastavení Hifas die ústředny plán a odchod routy	ok	vytvořeno jednoduché, ležce srozumitelné grafické rozhraní
nastavení nastavení z ústředny	ok	neprv se musí vyplnit název směrování a odchodní rout

Za Aiteco
Anita Ešer

Za Hotel2000
Ing. Ondřej Pekárek


AITECO S.R.O.
 Na Parku 1062/50, 140 00 Praha 4
 DIČ: CZ27995211


 MOJE PRÁVNÍ
 PODPIS
 Ing. Ondřej Pekárek
 IČO: 27995211

Obrázek 24 - Akceptační protokol komunikace HIFAS se software Hotel2000

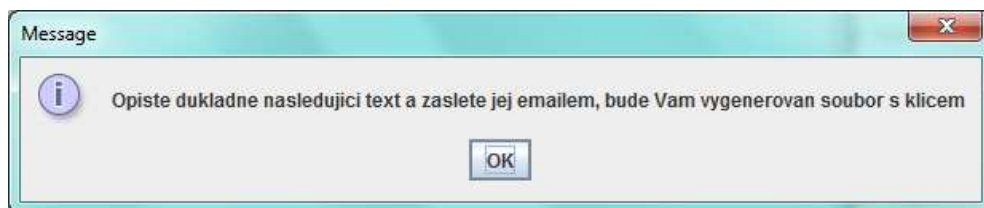
7 Ochranný mechanismus

Program byl na žádost zaměstnavatele opatřen ochranným mechanismem a to dvojího druhu - v iniciačním souboru jsou hesla v kódované podobě a program je chráněn kódovaným složením názvu počítače a MAC adresy.

Důvody k zavedení alespoň jednoduchého ochranného mechanismu jsou dva. První jsem zde již popsal. V textovém souboru s nastavením jsou jak přihlašovací údaje do SQL databáze, tak do telefonní ústředny pro SOAP komunikaci.

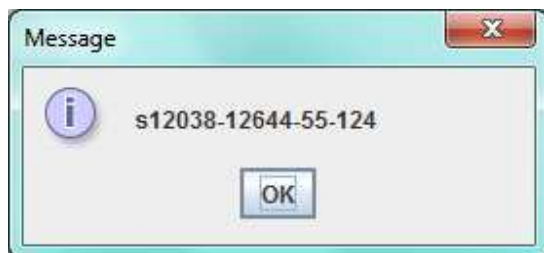
Druhý důvod je finanční, tedy alespoň základní ošetření proti neomezenému kopírování a používání mého produktu, bez zaplacení poplatku, který byl odhadnut při společném jednání s dodavateli ústředny na přibližně 10 000,- Kč bez DPH za jednu instalaci.

V aktuálním adresáři je nutné mít soubor *serial.ser*, obsahující výše zmíněný kód. Pro první spuštění bude třeba ještě program aktivovat - objeví se následující hlášení:



Obrázek 25 - informace od interface uživateli při prvním spuštění

Po stisku klávesy OK důkladně opište následující hlášení a odešlete jej emailem.



Obrázek 26 - identifikace počítače na kterém poběží interface HIFAS

Bude vám vygenerován soubor s kódem pro váš počítač. Ten následně umístíte do adresáře, odkud se bude program spouštět.

Po obdržení a nakopírování souboru *serial.ser* do adresáře již můžete program *APP.jar* spustit.

Program HIFAS pro aktivaci používá stejný kódovací mechanismus, jako program CodeDecode.jar .

Proto stačí před instalací odmazat případný existující soubor *serial.ser* a vytvořit nový soubor *serial.ser* v jednoduchém textovém editoru, například v notepadu.

Spustíte program CodeDecode.jar, kde si necháte zakódovat vygenerovaný řetězec pro aktivaci programu. Nakonec spustíte program. Dále pokračujte dle přílohy (6).

8 Presentace programu dodavateli telefonních ústředen IPEX

26.11.2010 proběhlo předvedení programu s napojením na hotelový systém Epitome dodavateli ústředen IPEX. Zde byl vznesen požadavek o zapracování vynuceného volání na předdefinovanou linku, kde se ozve varování o kontrole hotelového interface při neúspěšném buzení. Tato funkcionality byla do programu ihned začleněna.

Účastní při této prezentaci byli za naši firmu technik spravující telefonní ústředny koncernu CIMEX a řetězce OREA HOTELS Vladimír Honzák, dále ředitel Aiteco s.r.o. Ing. Jiří Matoušek a za firmu IPEX servisní technik a obchodní zástupce. Ti, po předvedení, s ředitelem naší firmy projednávali možný obchodní plán.

9 Závěr

Tento program byl vytvořen pro potřeby zaměstnavatele, který v rámci úsporných opatření pořídil v koncernu CIMEX VoIP telefonní ústředny IPEX.

Mezi společnostmi koncernu CIMEX patří i OREA HOTELS s.r.o. provozující v současnosti 21 hotelů po České republice. Telefonní ústředny IPEX však samy o sobě nenabízejí podporu hotelových funkcí, proto nemohou být nasazeny v hotelovém prostředí.

Tento nedostatek řeší program HIFAS, který telefonním ústřednám IPEX umožní zejména zprostředkování základních hotelových funkcí.

Tento program i bez integrace s hotelovým systémem lze používat jako nástroj pro základní ovládání telefonní ústředny s funkcemi změny oprávnění jednotlivých linek, případně pro funkci „spojovatelka“, kdy může recepční přes program vytočit z určené linky požadovanou linku a požádat hosta, necht' zvedne vyzvánějící telefon a poté bude na požadované číslo spojen.

Práci na programu HIFAS vidím jako velice přínosnou z hlediska seznámení se s nově se rozvíjejícím typem komunikace A2A či B2B. Při tvorbě aplikace jsem zužitkoval nabyté informace na tomto institutu, především programovací jazyk Java.

Výsledek vývoje hodnotím jednoznačně kladně, protože splnil naše očekávání. Díky tomu jsme schopni vlastními silami přizpůsobit telefonní ústřednu IPEX našim potřebám.

Tento program se stane základem pro možné další rozšíření dle aktuálních potřeb i ve sféře kancelářských budov, kde jsou tyto ústředny již nasazeny.

Děkuji škole, že jsem si mohl vybrat právě toto téma a skloubit tím pracovní povinnosti s absolventskou prací. Díky této práci vznikla kompletně zdokumentovaná aplikace (s kompletním [Javadoc](#) [21]) i s návodem k instalaci a užití.

9.1 Možná rozšíření do budoucna

Mezi první možná rozšíření patří přidání uživatelského rozhraní, jenž by umožnilo hotelovému personálu jednoduché ovládání buzení a dalších vstupů směrem k ústředně, jako je například ovládání oprávnění telefonních linek. Toto rozhraní bude vytvořeno ve skriptovacím programovacím jazyku PHP určeném pro programování dynamických internetových stránek.

Coby další možné rozšíření bude následovat přidělení dalších komunikačních protokolů na TCP vrstvě pro interakci s hotelovými systémy Micros Fidelio, protože se na hotelovém řetězci OREA HOTELS nasazuje nový PMS Systém Opera od společnosti Micros.

Mezi dalšími návrhy je použití funkcionality na způsob hromadného buzení pro napojení či náhradu EPS (Elektronická požární signalizace), kde by se rozezvučely telefony po celé budově (či pouze na definovaných linkách) s hlášením o evakuaci budovy či typu poplachu s možnými následnými úkony, třeba i automatickému volání záchranné služby.

Pro kancelářské budovy by bylo možné program upravit například pro tyto účely:

- transformace tarifikačních výpisů pro tarifikační programy
- ovládaní oprávnění linek
- funkce spojovatelka
- telefonní seznam
- napojení na EPS
- případné další možné užití

10 Seznam použitých zkratk

SOAP - (původně Simple Object Access Protocol) je protokolem pro výměnu zpráv založených na XML přes síť, hlavně pomocí HTTP [1]

XML - Extensible Markup Language (zkráceně XML, česky rozšiřitelný značkovací jazyk) je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C [8]

HTTP - Hypertext Transfer Protocol je internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML [9]

VoIP - Voice over Internet Protocol je technologie, umožňující přenos digitalizovaného hlasu v těle paketů rodiny protokolů UDP/TCP/IP prostřednictvím počítačové sítě nebo jiného média prostupného pro protokol IP [10]

RPC – Remote Procedure Call (vzdálené volání procedur) je technologie dovolující programu vykonat proceduru, která může být uložena na jiném místě, než je umístěn sám volající program. Příkladem budiž výpočet funkce na jiném počítači v síti [11]

WSDL - Web Services Description Language popisuje, co nabízí webová služba za funkce a způsob, jak se jí na to zeptat. Zapisuje se v XML formátu. Zpravidla tedy popisuje SOAP komunikaci [12]

PBX - (Private Branch Exchange) je zařízení (počítač), které sjednocuje výstupní body všech firemních telefonů do veřejné telefonní sítě. PBX má kromě zjednodušení obsluhy také ohromný dopad na úspory. Místo vedení poplatků za každé telefonní číslo platí firma pouze přístupový bod [13]

SQL - je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. SQL je zkratka anglických slov Structured Query Language (strukturovaný dotazovací jazyk) [14]

PMS – Hotelový rezervační systém jde o zkratku anglického „Property Management System“

MAC (z anglického „Media Access Control“) je jedinečný identifikátor síťového zařízení, který používají různé protokoly druhé (spojové) vrstvy OSI. Je přiřazována síťové kartě NIC bezprostředně při její výrobě (u starších karet je přímo uložena do EEPROM paměti) a proto se jí také někdy říká fyzická adresa [15]

B2B - Business-to-Business je označení pro obchodní vztahy mezi obchodními společnostmi pro jejich potřeby, které neobsluhují konečné spotřebitele v masovém měřítku. Významným rysem modelu B2B je větší důraz na logistiku a zajištění samotného obchodu [16]

A2A – Application-to-Application. Takto označujeme všechny moduly, které zprostředkovávají komunikaci mezi dvěma aplikacemi: často mohou mít rysy B2B systémů a propojení, ale nejčastěji se jedná o propojení a vzájemnou integraci aplikací jednoho podnikatelského subjektu [17]

EPS - Elektronická požární signalizace

CSÚIS - Centrální systém účetních informací státu

11 Seznam použitých zdrojů

- [1] SOAP. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 10. 11. 2006, last modified on 24. 8. 2011 [cit. 2011-09-10]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/SOAP>>.
- [2] Firewall. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 22. 2. 2006, last modified on 30. 8. 2011 [cit. 2011-09-10]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Firewall>>.
- [3] KOPECKÝ, Jacek. *Zive.cz* [online]. 4. 7. 2001 [cit. 2011-09-10]. SOAP - konečně správné pH pro váš software?. Dostupné z WWW: <<http://www.zive.cz/clanky/soap---konecne-spravne-ph-pro-vas-software/sc-3-a-101819/default.aspx>>.
- [4] B2B. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 13. 9. 2005, last modified on 27. 8. 2011 [cit. 2011-09-10]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/B2B>>.
- [5] *Static.eplanet.sk* [online]. 12.9.2006 [cit. 2011-09-10]. KAPITOLA 32 Tvorba webových služeb. Dostupné z WWW: <http://static.eplanet.sk/files/9788086815381_06.pdf>.
- [6] *ote-cr.cz* : Část D1.4.4 Rozhraní automatické komunikace IS OTE Specifikace pro Ugrade CS OTE [online]. 2.12. 19.8.2011 [cit. 2011-09-10]. D1.4 Externí rozhraní CS OTE. Dostupné z WWW: <http://www.ote-cr.cz/dokumentace/dokumentace-elektřina/files_dokumentace/D1.4.4_Rozhrani_automaticke_komunikace_IS_OTE.doc>.
- [7] *Teamviewer.com* [online]. 21-sep-2004 [cit. 2011-09-10]. TeamViewer – řešení vše v jednom pro vzdálený přístup a podporu přes Internet. Dostupné z WWW: <<http://teamviewer.com/cs/index.aspx>>.
- [8] XML. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2. 12. 2007, last modified on 18. 1. 2011 [cit. 2011-09-11]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/XML>>.
- [9] Http. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 28. 10. 2004, last modified on 3. 1. 2008 [cit. 2011-09-11]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Http>>.

- [10] VoIP. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 19. 9. 2007, last modified on 19. 9. 2007 [cit. 2011-09-11]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/VoIP>>.
- [11] RPC. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2. 6. 2007, last modified on 2. 6. 2007 [cit. 2011-09-11]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/RPC>>.
- [12] WSDL. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 23. 11. 2006, last modified on 23. 11. 2006 [cit. 2011-09-11]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/WSDL>>.
- [13] PBX. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 6. 6. 2010, last modified on 29. 10. 2010 [cit. 2011-09-11]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/PBX>>.
- [14] SQL. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 24. 11. 2004, last modified on 2. 7. 2011 [cit. 2011-09-11]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/SQL>>.
- [15] MAC adresa. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2. 4. 2005, last modified on 16. 5. 2011 [cit. 2011-09-11]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/MAC_adresa>.
- [16] B2B. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 13. 9. 2005, last modified on 27. 8. 2011 [cit. 2011-09-11]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/B2B>>.
- [17] MATĚNA, Roman. *Solisshop.cz* [online]. 2008 [cit. 2011-09-11]. Popis řešení - A2A. Dostupné z WWW: <<http://www.solisshop.cz/Popis-reseni-A2A.htm>>.
- [18] KOSEK, Jiří. *Kosek.cz* [online]. 2002, 1.3 2003/08/21 20:26:46 [cit. 2011-09-13]. Využití webových služeb a protokolu SOAP při komunikaci. Dostupné z WWW: <<http://www.kosek.cz/diplomka/html/websluzby.html>>.
- [19] *Zdrojak.root.cz* [online]. 2010 [cit. 2011-09-14]. Java na webovém serveru: SOAP webové služby. Dostupné z WWW: <<http://zdrojak.root.cz/clanky/java-na-webovem-serveru-soap-webove-sluzby/>>.
- [20] *Root.cz* [online]. 2007 [cit. 2011-09-14]. Webové služby v PHP: XML-RPC a SOAP. Dostupné z WWW: <<http://www.root.cz/clanky/webove-sluzby-php-xmlrpc-soap/>>.

- [21] Javadoc. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 16. 4. 2007, last modified on 10. 4. 2010 [cit. 2011-09-14]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Javadoc>>.
- [22] HEROUT, Pavel. *Učebnice Jazyka Java*. Třetí rozšířené vydání. České Budějovice : Kopp, 2007. 381 s.
- [23] KISZKA, Bogdan. *1001 tipů a triků pro jazyk Java*. Vydání první. Brno : Computer Press, a.s., 2009. 542 s.

12 Seznam příloh

- (1) *prilohy\3Shb_G9742-S100-1-7619.pdf* – Siemens manuál pro instalaci programu Wincal s popisem protokolu Caracas strany 541 až 563 (bod 13.7)
- (2) *prilohy\AkceptacniProtokolHotel2000.jpg* – naskenovaný akceptační protokol z akceptačního testovacího provozu s hotelovým systémem Hotel2000.
- (3) *prilohy\HifasNabidka.doc* – koncept textu nabídky programu HIFAS.
- (4) *prilohy\HifasNavod.doc* – návod na obsluhu programu HIFAS.
- (5) *prilohy\HifasTestovaciScenar.xls* – připravený scénář pro testování.
- (6) *prilohy\PostupZprovozneniNoveUstredny.doc* – návod na zprovoznění nové telefonní ústředny IPEX s programem HIFAS.
- (7) *prilohy\javadoc.zip* – javadoc programu HIFAS
- (8) *prilohy\SQL\ZalozeniDB.sql* – SQL script pro založení databáze IPEX
- (9) *prilohy\program\App.jar* – program HIFAS
- (10) *prilohy\program\CodeDecode.jar* – program na kódování a dekódování textu
- (11) *prilohy\program\serial.ser* – aktivační soubor, který je pro každý PC nutné vygenerovat
- (12) *prilohy\program\soap.properties* – konfigurační soubor
- (13) *zdrojaky\soap\App* – projekt programu HIFAS
- (14) *prilohy\Diagram_tridy.png* – UML diagram tříd programu HIFAS
- (15) *zdrojoveKody\soap\App* - adresář projektu se zdrojovými kódy programu
- (16) *AP_Ecer.doc* – tato práce
- (17) *ZadaniAP.doc* – zadání absolventské práce
- (18) *readme.txt* – soubor s popisem adresářové struktury přiloženého CD