

Inovace aplikace Paveza společnosti NWT a.s. pro přechod na nový databázový engine

Bc. Petr Malý

Diplomová práce
2014



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Bc. Petr Malý
Osobní číslo: A12470
Studijní program: N3902 Inženýrská informatika
Studijní obor: Informační technologie
Forma studia: kombinovaná

Téma práce: Inovace aplikace Paveza společnosti NWT a.s. pro
přechod na nový databázový engine

Zásady pro vypracování

1. Seznamte se současným stavem aplikace Paveza.
2. Analyzujte v současnosti používaný a nově zvolený databázový engine (zvláštní pozornost věnujte rozdílům v těchto databázových enginech).
3. Navrhněte řešení přechodu na nový DB engine popř. změny v aplikaci.
4. Navržené řešení implementujte.
5. Testujte funkčnost implementovaných řešení.
6. Proveďte závěrečnou analýzu kompletního řešení, zhodnoťte přínos projektu.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. EVJEN, Bill, Scott HANSELMAN a Devin RADER. Professional ASP.NET 4 in C# and VB. Indianapolis, IN: Wiley Pub., 2010. ISBN 04-705-0220-7.
2. JORGENSEN, Adam. Professional microsoft sql server 2012 administration. 1st ed. Indianapolis, IN: Wiley Pub., Inc., 2012. ISBN 11-181-0688-1.
3. ROSS MISTRY, Stacia Misner. Introducing Microsoft SQL Server 2012. Redmond, WA: Microsoft Press. ISBN 07-356-6515-X.
4. KUHN, Darl. Pro Oracle database 11g administration. New York, N.Y.: distributed by Springer Science Business Media, 2010. ISBN 14-302-2970-5.
5. PRICE, Jason. Oracle database 11g SQL. New York: Oracle Press/McGraw-Hill, 2008. ISBN 00-714-9850-8.

Vedoucí diplomové práce:

Ing. Bc. Pavel Vařacha, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

21. února 2014

Termín odevzdání diplomové práce:

20. května 2014

Ve Zlíně dne 21. února 2014

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

PROHLÁŠENÍ

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby ¹⁾;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl jsem seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3 ²⁾;
- beru na vědomí, že podle § 60 ³⁾ odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 ³⁾ odst. 2 a 3 mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Ve Zlíně

.....

1) zákon č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, § 47 Zveřejňování závěrečných prací:

(1) Vysoká škola nevydělečně zveřejňuje disertační, diplomové, bakalářské a rigorózní práce, u kterých proběhla obhajoba, včetně posudků oponentů a výsledku obhajoby prostřednictvím databáze kvalifikačních prací, kterou spravuje. Způsob zveřejnění stanoví vnitřní předpis vysoké školy.

(2) Disertační, diplomové, bakalářské a rigorózní práce odevzdané uchazečem k obhajobě musí být též nejméně pět pracovních dnů před konáním obhajoby zveřejněny k nahlížení veřejnosti v místě určeném vnitřním předpisem vysoké školy nebo není-li tak určeno, v místě pracoviště vysoké školy, kde se má konat obhajoba práce. Každý si může ze zveřejněné práce pořizovat na své náklady výpisy, opisy nebo rozmnoženiny.

(3) Platí, že odevzdáním práce autor souhlasí se zveřejněním své práce podle tohoto zákona, bez ohledu na výsledek obhajoby.

2) zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, § 35 odst. 3:

(3) Do práva autorského také nezasahuje škola nebo školské či vzdělávací zařízení, užije-li nikoli za účelem přímého nebo nepřímého hospodářského nebo obchodního prospěchu k výuce nebo k vlastní potřebě dílo vytvořené žákem nebo studentem ke splnění školních nebo studijních povinností vyplývajících z jeho právního vztahu ke škole nebo školskému či vzdělávacímu zařízení (školní dílo).

3) zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, § 60 Školní dílo:

(1) Škola nebo školské či vzdělávací zařízení mají za obvyklých podmínek právo na uzavření licenční smlouvy o užití školního díla (§ 35 odst. 3). Odpírá-li autor takového díla udělit svolení bez vážného důvodu, mohou se tyto osoby domáhat nahrazení chybějícího projevu jeho vůle u soudu. Ustanovení § 35 odst. 3 zůstává nedotčeno.

(2) Není-li sjednáno jinak, může autor školního díla své dílo užít či poskytnout jinému licenci, není-li to v rozporu s oprávněnými zájmy školy nebo školského či vzdělávacího zařízení.

(3) Škola nebo školské či vzdělávací zařízení jsou oprávněny požadovat, aby jim autor školního díla z výdělku jim dosaženého v souvislosti s užitím díla či poskytnutím licence podle odstavce 2 přiměřeně přispěl na úhradu nákladů, které na vytvoření díla vynaložily, a to podle okolností až do jejich skutečné výše; přitom se přihlédne k výši výdělku dosaženého školou nebo školským či vzdělávacím zařízením z užití školního díla podle odstavce 1.

ABSTRAKT

Intranetová aplikace PAVEZA byla vytvořena společností NWT a.s., aby svým zákazníkům pomohl s administrací spojenou se zadáváním a správou veřejných zakázek. V současné době aplikace komunikuje pouze s databázovým enginem Oracle. Společnost NWT se rozhodla rozšířit kompatibilní datové úložiště o Microsoft SQL Server. Cílem této práce je návrh a realizace tohoto přechodu.

Klíčová slova: Microsoft SQL Server, Oracle, PAVEZA, veřejné zakázky, .NET

ABSTRACT

Intranet PAVEZA application was developed by NWT a.s. to help its customers with administration associated with awarding and management of public contracts. By now is the Oracle database only database engine, which PAVEZA application can cooperate with. NWT company wants to extend compatible database engines of Microsoft SQL Server. The aim of this thesis is to design and implement this functionality.

Keywords: Microsoft SQL Server, Oracle, PAVEZA, Public Contracts, .NET

Děkuji panu Ing. Bc. Pavlu Vařachovi, Ph.D. za odborné vedení této práce. Dále bych chtěl poděkovat Lukáši Vičánkovi a zaměstnancům firmy NWT a.s. za vstřícnou spolupráci a poskytnuté konzultace a materiály a Ingrid Vyroubalové za rady a poskytnuté materiály z ekonomicko-právní oblasti.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 APLIKACE PAVEZA	12
1.1 O SPOLEČNOSTI NWT A.S.	12
1.2 O APLIKACI PAVEZA	15
1.2.1 Modul plánování	17
1.2.2 Modul veřejných zakázek	18
1.2.3 Zákazníci	20
1.3 VEŘEJNÉ ZAKÁZKY V ČR	21
1.3.1 Historie veřejných zakázek v ČR	21
1.3.2 Veřejné zakázky	22
1.3.3 Zadávací řízení	23
1.3.4 Zadávání VZ.....	24
1.3.5 Vymezení elektronických prostředků	24
1.3.6 Požadavky na elektronické nástroje	25
1.3.7 Elektronizace veřejných zakázek	26
1.3.8 Problémy se zadáváním VZ	28
2 DATABÁZOVÉ ENGINY	29
2.1 DATABÁZOVÝ ENGINE ORACLE.....	29
2.1.1 Verze Oracle DB	29
2.1.2 Oracle DB 12c	30
2.1.3 PL/SQL	31
2.2 DATABÁZOVÝ ENGINE SQL SERVER.....	31
2.2.1 Verze SQL Server	31
2.2.2 SQL Server 2012.....	32
2.2.3 T-SQL	32
2.3 ODLIŠNOSTI DATABÁZÍ ORACLE A SQL SERVER.....	33
2.3.1 Datové typy	33
2.3.2 DML.....	36
II PRAKTICKÁ ČÁST	39
3 NÁVRH A REALIZACE PŘECHODU NA NOVÝ DB ENGINE	40
3.1 POUŽITÉ TECHNOLOGIE	40
3.1.1 .NET Framework.....	40
3.2 POUŽITÉ VÝVOJOVÉ NÁSTROJE.....	41
3.2.1 Microsoft Visual Studio	41
3.2.2 SQL Server Management Studio	42
3.3 NÁVRH ŘEŠENÍ	43
3.3.1 UniCommand	45
3.3.2 UniInsert.....	45
3.3.3 UniSelect	45
3.3.4 UniUpdate	46
3.3.5 UniDelete	46
3.3.6 UniFunction.....	46

3.3.7	UniProcedure.....	46
3.4	IMPLEMENTACE NAVRŽENÉHO ŘEŠENÍ	47
3.4.1	Implementace UniInsert	47
3.4.2	Implementace UniSelect	49
3.4.3	Implementace UniUpdate.....	51
3.4.4	Implementace UniDelete.....	52
3.4.5	Implementace UniFunction.....	54
3.4.6	Implementace UniProcedure.....	55
3.4.7	Testování provedených změn.....	56
4	VYHODNOCENÍ PROVEDENÝCH ÚPRAV.....	57
	ZÁVĚR	59
	SEZNAM POUŽITÉ LITERATURY.....	60
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	62
	SEZNAM OBRÁZKŮ	63
	SEZNAM TABULEK.....	64

ÚVOD

Trendem několika minulých a budoucích let v oblasti veřejných zakázek je jejich kompletní elektronizace a to nejen v České republice, ale po celém světě. Elektronizace veřejných zakázek přináší řadu výhod, mezi které nesporně patří snížení administrativní zátěže zadavatele i dodavatele, se kterým souvisí dále např. snížení personálních nákladů. Je třeba však řešit i spoustu problémů, které se zadáváním a obecně vedením veřejných zakázek v elektronické podobě souvisejí. Pro efektivní plánování a evidenci veřejných zakázek byla společností NWT a.s. vytvořena aplikace PAVEZA, jež je předmětem této diplomové práce.

Aplikace PAVEZA přináší úsporu nákupů ve formě jejich sdružení, snížení personálních nákladů, eliminuje riziko pokut vlivem porušení zákona o veřejných zakázkách atd. Její využití je však možné pouze se zakoupenými licencemi Oracle databáze, což je jediný databázový engine, nad kterým je schopna aplikace PAVEZA pracovat. Cílem této práce je návrh a realizace řešení přechodu na databázový engine SQL Server, s přihlédnutím k možnosti v budoucnu aplikaci PAVEZA rozšířit pro práci nad dalšími databázovými enginy.

Teoretická část práce se analyzuje současný stav aplikace PAVEZA, okrajově se věnuje společnosti NWT a.s., které je autorem této aplikace, dále se věnuje veřejným zakázkám v České republice včetně fáze jejich elektronizace a vlastnostem elektronických nástrojů, které vyhovují zákonu o veřejných zakázkách a v poslední části zkoumá oba databázové enginy (stávající a DB engine, na který aplikace PAVEZA přechází) a jejich odlišnosti.

V praktické části je navrhnout postup řešení, ukázány principiální části implementace a je zhodnocen přínos realizovaných změn.

I. TEORETICKÁ ČÁST

1 APLIKACE PAVEZA

Proces zadání veřejné zakázky (dále jen VZ) a průběh výběrového řízení závisí na předmětu zakázky a jejich předpokládaných nákladech. Příprava zadání veřejné zakázky s sebou nese také řadu povinností a administrativních úkonů, jimiž se zabývá internetová aplikace PAVEZA společnosti NWT a.s. Následující kapitola se věnuje této aplikaci a společnosti, která ji uvedla na trh a VZ v ČR.

1.1 O společnosti NWT a.s.

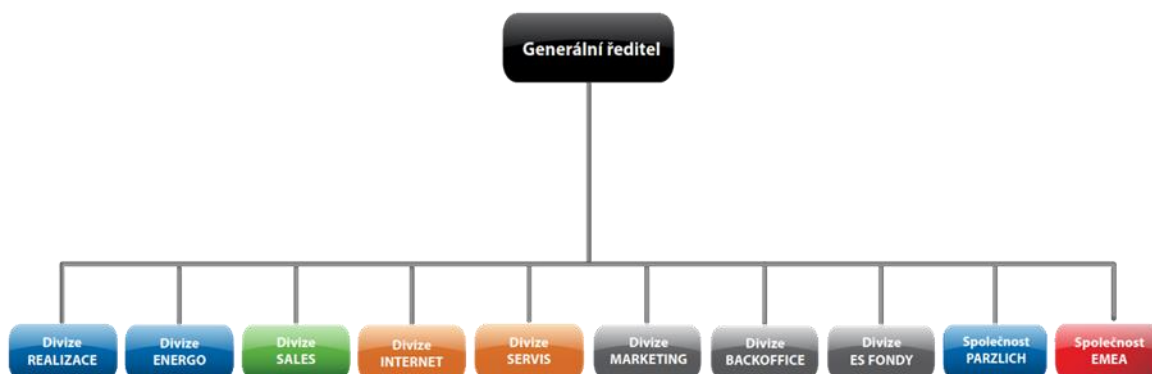
V roce 1992 byla v Kroměříži založena společnost NWT Computer (zkratka z anglických slov New World Technologies) jako ryze česká privátní firma. Ve svých počátcích se firma orientovala na prodej a služeb koncovým zákazníkům v oblasti hardware a software.



Obr. 1 – Logo společnosti NWT a.s. [1]

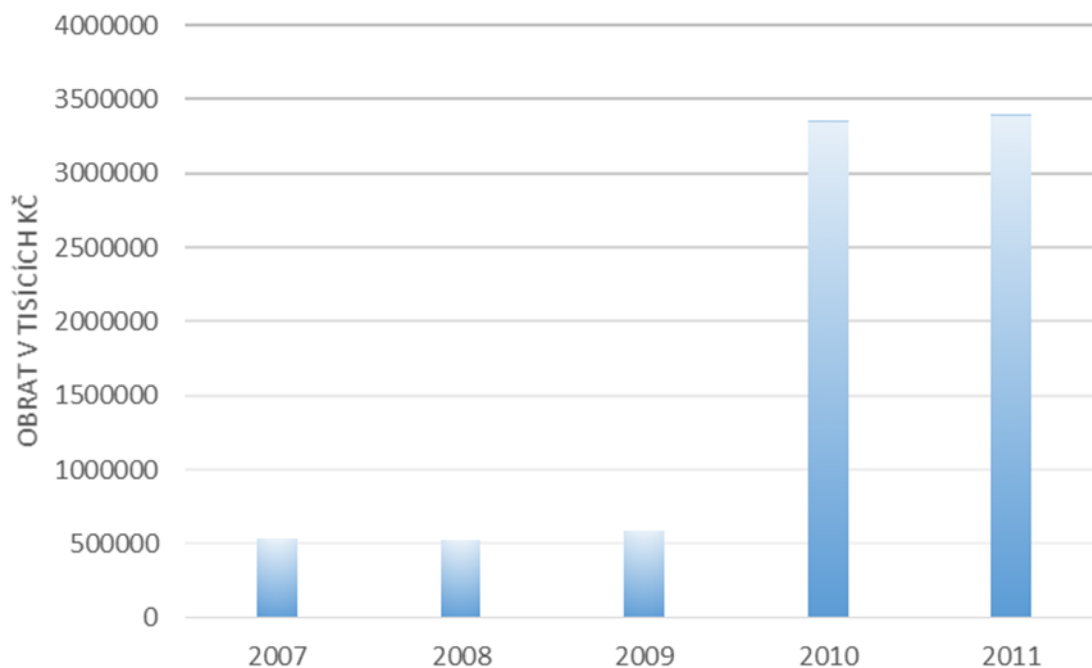
Společnost se nadále dynamicky rozvíjela a v roce 1995 se transformuje na právnickou osobu – společnost s ručením omezeným. Sídlem nově vzniklé s.r.o. se stává Hulín a zahajuje velkoobchodní prodej informačních technologií v regionu Jižní Morava. Na počátku roku 1999 vzniká pobočka ve Zlíně ve snaze přiblížit se více zákazníkům.

V průběhu následujících let vznikaly ve společnosti nové divize a pokračovala expanze divizí stávajících (Obrázek 2 zachycuje současnou organizační strukturu společnosti). Aplikace PAVEZA, které se věnuje tato práce, pochází z dílny divize Servis.



Obr. 2 - Organizační struktura společnosti NWT a.s. [4]

Jsou zřizovány další provozovny a NWT sbírá ocenění a certifikace. V roce 1999 získává NWT certifikát ČSN ISO 9001 a 9002. V témže roce se společnost stává *Microsoft Certified* partnerem a získává ocenění *HP Business Partner roku*. V roce 2005 získává NWT certifikát *Investors in People* a o dva roky později se stává autorizovaným servisním partnerem IBM. Společnost se svým rozvojem navyšuje počet zaměstnanců (v současnosti pracuje pro NWT přes 170 lidí) a její obraty mají rostoucí trend (Obrázek 3). V roce 2011 se společnost NWT umístila v prestižním žebříčku TOP 100 ICT firem v ČR na 9. místě.



Obr. 3 - Obrat společnosti NWT a.s. v letech 2007 až 2011

V září roku 2010 realizovala společnost NWT Computer s.r.o. v souladu s příslušnými ustanoveními zákona č. 125/2008 Sb., o přeměnách obchodních společností a družstev, ve znění pozdějších předpisů, proces změny právní formy ze společnosti s ručeným omezením na akciovou společnost. V souvislosti s touto změnou došlo ke změně názvu společnosti z NWT Computer s.r.o na NWT a.s. a bylo představeno nové logo společnosti (Obrázek 1), které společnost používá dodnes [3][4].

Oblasti podnikání společnosti NWT a.s. prošly od jejího založení výraznou obměnou a nyní se společnost soustředí především na tyto oblasti:

Energetika

- fotovoltaické elektrárny;
- bioplynové stanice;
- pyrolýza;
- těžba skládkového plynu;
- hydrolýza a kotle na biomasu;
- LED – průmyslové haly a veřejné osvětlení;
- Komplexní poradenství energetických úspor.

Výroba

- bowdeny (plastové trubičky) pro stavební průmysl;
- stavební výroba.

Realizace – specializované stavby

- elektrárny na klíč (fotovoltaika, bioplyn);
- technologické celky;
- územní plánování a veškeré inženýrské práce;
- komplexní stavební práce;
- elektroinstalace.

Výzkum a vývoj

- studie energetické účinnosti domů;
- nanotechnologie;
- vertikální rotor a vícepólový generátor větrných elektráren;
- koncentrátová voltaika (CPV).

Služby

- správa nemovitostí;
- vedení účetnictví;
- komplexní finanční poradenství;
- provoz elektrické sítě;
- telekomunikace;
- kamerové systémy;
- komplexní IT služby vč. outsourcingu;
- cloudové technologie [4].

1.2 O aplikaci PAVEZA

Intranetová aplikace PAVEZA a její odlehčená verze PAVEZA LIGHT jsou dílem společnosti NWT, a.s. a byly na trh uvedeny jako elektronické nástroje, který zadavatelům veřejných zakázek usnadní administraci spojenou se zadáním veřejné zakázky. Zároveň si kladou za cíl zvýšit transparentnost celého procesu zadání zakázky s možností vnitřního i vnějšího dohledu nad přípravou a definováním zadávacích podmínek veřejné zakázky, a to již od okamžiku vzniku záměru.

The screenshot displays the PAVEZA application interface. At the top, there is a navigation bar with the PAVEZA logo and several menu items: Plánování, Evidence veřejných zakázek, Administrace, Uživ. nastavení, Profil - demo, and UTR. Below the navigation bar, the main content area is divided into a left sidebar and a main panel.

Detail veřejné zakázky

Dokumentace

- Šablony ke stažení
- Generované dokumenty
- Importované dokumenty
- Publikace dokumentů

Informace o zakázce

- Administráční panel
- Základní parametry
- ✓ Zveřejnění
- ✓ Lhůty
- ✓ Kvalifikační předpoklady
- ✓ Požadavky na nabídku
- ✓ Hodnocení nabídek
- ✓ Komise
- ✓ Oslovení dodavatelé
- ✓ Odpovědné osoby

Detail veřejné zakázky

Zpět na evidenci veřejných zakázek

Evidenční číslo:	OINF/2013/001	Osoba odpovědná za administraci:	Martin Schejbal
Název:	Dodávka multifunkčních kopírek	Administrátor (uživatel) VZ:	Michal Hošek
Způsob zadání:	Výzva bez uveřejnění	Zahájeno dne:	-
Hodnota VZ:	250 000	Podání nabídek:	-
Hodnota plnění:	0	Schváleno dne:	-
Stav:	Tvorba ZD		

Detail veřejné zakázky Dodávka multifunkčních kopírek (OINF/2013/001)

Základní parametry

Název veřejné zakázky *

Evidenční číslo *

Identifikátor VZ na profilu zadavatele *

Zadavatel (vypisující OJ) *

Pro potřeby OJ *

Klasifikace veřejné zakázky (kód CPV) * 30121100-4 Fotokopírovací stroje

Vedoucí střediska *

Referent *

Termín realizace od *

Termín realizace do *

Obr. 4 - PAVEZA: Detail veřejné zakázky [2]

Aplikace PAVEZA je určena především pro města, obce a jiné územně samosprávné celky, dále pro státní podniky a organizace, ale také pro komerční subjekty, které např. v rámci svého podnikání žádají o dotace apod.

Mezi přínosy používání aplikace PAVEZA patří především sdružování dílčích nákupů do větších komplexních celků, čímž je jednak možno dosáhnout nižších cen v rámci množstevních slev a také odpadají náklady na administrativní úkony spojené s jednotlivými dílčími nákupy. Využití aplikace PAVEZA přináší také sekundární úspory personálních nákladů spojené s efektivnější správou VZ.

Další výhody spojené s využíváním této aplikace jsou eliminace pokut vlivem nižšího rizika porušení zákona o veřejných zakázkách, zprůhlednění systému nákupu v celé organizaci včetně dílčích organizačních složek, naplnění zákonných povinností v zadávání VZ na všech organizačních úrovních i pro VZMR a implementace software (k automatizaci nákupního procesu) provázaného na stávající systém organizace.

The screenshot displays the PAVEZA application interface. At the top, there is a navigation bar with tabs for 'Plánování', 'Evidence veřejných zakázek', 'Administrace', 'Uživ. nastavení', 'Profil - demo', and 'UTR'. The main interface is divided into several sections:

- Left sidebar:** Contains a menu with options like 'Plánování - úvod', 'Plánování dle organizační struktury', 'Vkládání a úprav položek', 'Schvalování položek', 'Plánování na projekty', 'Přehledy plánů', and 'Export plánů'.
- Main content area:** Shows a tree view of procurement items under the heading 'Vkládání a úpravy položek'. The tree includes categories like 'Paliva a elektrická energie', 'Kancelářská technika a zařízení', 'Různý papírový materiál', and 'Kancelářské potřeby'.
- Right panel:** Displays a detailed view of a specific item, 'Různý papírový materiál'. It includes a search bar, a table with columns for 'Perioda', 'Cena Kč *', and 'Poznámka', and a 'Zobrazit' button.

Obr. 5 - PAVEZA: Plánování VZ [2]

Výhody řešení PAVEZA jsou podle [2] následující:

- zavedení jednotného systému a postupů na všech organizačních úrovních zadavatele;
- efektivní řízení procesu zadávání a správy veřejných zakázek;
- úspora času při přípravě a administraci zadávacích řízení;
- zjednodušení a zprůhlednění celého procesu zadávání veřejných zakázek;

- zefektivnění rutinních administrativních úkonů při zadávání a hodnocení VZ;
- kompletní vedení agendy dokumentace celého zadávacího řízení;
- správa a evidence všech veřejných zakázek v jednotném softwarovém prostředí;
- předdefinované šablony dokumentů;
- možné modifikace specifických potřeb dle požadavků konkrétního zadavatele;
- manažerské přehledy a statistiky;
- možnost integrace na datová úložiště zadavatele.

Technologické údaje aplikací PAVEZA a PAVEZA LIGHT obsahuje tabulka 1:

Technologie	PAVEZA	PAVEZA LIGHT
DB server	Oracle (plán MSSQL)	Oracle
Umístění fyzické instalace	server zákazníka	hostovaná aplikace https://
Internetová aplikace (tj. bez instalace na PC)	x	x
Počet uživatelů v rámci licence zákazníka (IČ)	neomezeno	10
Navýšení počtu uživatelů		lze navýšit
Možnost integrace k lokálním aplikacím (DMS, spisová služba)	x	nelze
Možnost integrace na ERP, reálné řízení nákupu	x	nelze
Doba implementace	4 měsíce (dle úprav)	2 týdny

Tab. 1 - PAVEZA: Technologie [2]

1.2.1 Modul plánování

Tento modul umožňuje sestavit plán nákupních potřeb ve věcné struktuře dle modelu NIPEZ (Národní infrastruktura pro elektronické zadávání veřejných zakázek). Plán je možno sestavit podle finančního plánu dané organizace (příkladem takového plánu může být např.

rozpočet města), může být ale sestaven i přímo bez vazby na jakýkoliv finanční plán. Tímto způsobem je možné získat přehled o plánovaných nákupních potřebách celé organizace pro zvolené období.

Modul plánování	PAVEZA	PAVEZA LIGHT
Import rozpočtu města nebo jiného finančního plánu organizace	x	
Rozpad vlastního finančního plánu do věcné struktury NIPEZ	x	
Vkládání nákupních potřeb ve struktuře NIPEZ bez vazby na finanční plán	x	x
Uživatelská oprávnění dle organizační struktury	x	x

Tab. 2 - Modul plánování [2]

Ze získaného přehledu dále můžeme sdružit nákupní položky, které spolu věcně souvisí a agregovat jednotlivé nákupy do větších nákupních celků, čímž dosáhneme finanční úspory. Celý proces plánování je nastavitelný pracovní postup, který je sledovatelný na všech organizačních úrovních. Aplikace PAVEZA umožňuje také načtení rozpočtu z účetního systému organizace, to ale pouze v případě serverové instalace na straně zákazníka. Systém může fungovat i druhým směrem, tzn. data z tohoto modulu mohou být použita jako podklad pro sestavení finančního plánu.

1.2.2 Modul veřejných zakázek

Modul veřejných zakázek je vhodný pro podporu VZ vedených v klasické papírové formě. Hlavním cílem tohoto modulu je umožnit přípravu, administraci a evidenci všech nákupních aktivit, které probíhají podle předem stanovených postupů. Jednotlivé nákupní akce jsou uživatelem zadávány do připravených formulářů, které vyžadují zadání podstatných informací. Uživateli je dovoleno zvolit pouze takový výběr dodavatele, který je v souladu s interní směrnicí, případně zákonem.

Modul veřejných zakázek	PAVEZA	PAVEZA LIGHT
Generování dokumentů PDF dle uložených šablon	x	x
Customizace předdefinovaných PDF šablon dle potřeb zákazníka	x	omezeně
Import originálů (scanů) dokumentů	x	x
Vedení elektronického spisu dokumentů k VZ	x	x
Workflow v procesu přípravy a administrace VZ	x	x
Customizace workflow dle potřeb zákazníka	x	omezeně
Integrované propojení na certifikovaný profil zadavatele EVEZA (součást)	x	x
Integrace na aukční portál proe.biz (licence proe.biz není součástí)	x	x
Uživatelská oprávnění dle organizační struktury	x	x

Tab. 3 - Modul veřejných zakázek [2]

Celý proces přípravy veřejné zakázky může být na jednotlivých organizačních úrovních postupně schvalován (dle organizačních zvyklostí). Uživatel může prostřednictvím aplikace vytvářet PDF dokumenty, které jsou ve formátu předepsaném pro daný zadávací postup. Tím je zaručena jednotnost dokumentů a jejich obsahová správnost. Po vyhlášení VZ jsou uživateli zpřístupněny formuláře pro administraci. Pomocí nich je postupně formou generovaných PDF vytvářena požadovaná dokumentace (protokoly, zápisy atd.).

U VZ, u nichž je požadováno povinné uveřejnění na profilu zadavatele, může být tento profil součástí řešení. Zveřejňování dokumentů pak probíhá poloautomaticky, bez nutnosti uživatele vstupovat na profil a dokumenty ručně vkládat.

1.2.3 Zákazníci

Jak již bylo zmíněno výše, aplikace PAVEZA je určena zejména pro státní organizace a podniky a územně samosprávné celky. Hlavními zákazníky, kteří aplikaci PAVEZA využívají, jsou:

Lesy České republiky, s.p.

Lesy České republiky byly vůbec prvním zákazníkem, který začal v roce 2007 aplikaci PAVEZA využívat. Hlavním předmětem podnikání tohoto státního podniku je zajištění provádění činností zabezpečujících optimální plnění všech funkcí lesů a to vlastní režii, nebo prostřednictvím vybraných podnikatelských subjektů.



Obr. 6 - Logo státního podniku Lesy České republiky [5]

Plánování a provádění činností v lesích směřuje k zabezpečení trvalého souladu mezi potřebou, tvorbou a využitím vlastních finančních prostředků a k co nejhospodárnějšímu využití účelových příspěvků ze státního rozpočtu a z jiných zdrojů. Těmto základním principům se trvale přizpůsobuje konkrétní náplň předmětu činnosti Lesů České republiky i jejich organizační uspořádání [5].

Vysoké učení technické v Brně

Nejstarší brněnská vysoká škola využívá aplikaci PAVEZA především pro plánování nákupních potřeb. Jedná se o univerzitu s více než 110letou tradicí, která se skládá z 8 fakult a 3 vysokoškolských ústavů a v roce 2013 nabízela 74 akreditovaných studijních programů a 23 studijních programů vyučovaných v cizím jazyce [6].

Magistrát města Olomouce – Statutární město Olomouc

Statutární město Olomouc je dalším významným zákazníkem využívajícím aplikaci PAVEZA. Jedná se o šesté největší město v České republice (k 1.1.2012 platil údaj 99 529 obyvatel) a po hlavním městě o druhou největší památkovou rezervaci v ČR. Jako samostatná právnická osoba funguje statutární město Olomouc od 24.listopadu 1990 a jeho právní postavení je upraveno zákonem o obcích. Prostřednictvím Magistrátu města Olomouce vykonává státní správu v rozsahu stanoveném zvláštními zákony [7].

1.3 Veřejné zakázky v ČR

1.3.1 Historie veřejných zakázek v ČR

Právo veřejných zakázek má své kořeny již v prvorepublikové době. Tehdy, v roce 1920, bylo vydáno nařízení vlády Československé republiky č. 667/1920 Sb., o zadávání státních dodávek a prací (Zadávací řád). Toto nařízení se vztahovalo na dodávky zboží a prací, které zadávaly orgány státní správy a jimi spravované organizace, podniky a fondy [8].

Rozkvět oblasti práva veřejných zakázek přišel až se vstupem České republiky do Evropské unie v roce 2004. Od té doby může Česká republika čerpat finanční prostředky ze strukturálních fondů v rámci jednotlivých Operačních programů.

Právo veřejných zakázek se výrazně změnilo zákonem č. 40/2004 Sb., aby odpovídalo právním předpisům Evropské unie. Tento zákon byl několikrát novelizován, i přesto se v něm nacházela spousta problematických oblastí [9].

Následně bylo rozhodnuto o zcela nové právní úpravě veřejných zakázek. Byl přijat nový zákon č. 137/2006 Sb., o veřejných zakázkách. I tento právní předpis se však nevyhnul několika novelizacím. Během relativně krátkého období prošel tento právní předpis celkem 17 novelizacemi, které byly většinou motivovány snahou tehdejších vládních stran bojovat proti korupci v oblasti zadávání veřejných zakázek [10].

Poslední novelizace pozměnily oblast zadávání veřejných zakázek v České republice zásadním způsobem. Některé z těchto změn přitížily především malým obcím, které se náhle musely vyrovnávat s prudkým nárůstem administrativních a finančních úkonů, což se zejména pro obce, které doposud se zadáváním veřejných zakázek neměly žádné zkušenosti, představovalo značné komplikace.

1.3.2 Veřejné zakázky

Veřejné zakázky jsou nákupem služeb, zboží či stavebních prací z veřejných prostředků, se kterými disponují nejen státní orgány, ale také orgány státní samosprávy. Při zadávání a plnění VZ se zároveň musí dodržovat pravidla transparentnosti, nediskriminace a efektivní a hospodárné vynakládání veřejných prostředků.

Odborněji můžeme veřejnou zakázku charakterizovat jako simulaci tržních procesů, kdy je poptávajícím subjektem stát. I veřejné zakázky mají charakter směny zboží za peněžní prostředky a zadavatel čelí rozpočtovému omezení a nakupuje dle své potřeby. Tímto však podobnosti s obyčejným nákupem zboží končí. Nakupující subjekt totiž není konečným spotřebitelem. Zadavatel zakázky sice nakupuje zboží, ale jeho konečným uživatelem je někdo jiný [11].

Vymezení pojmu veřejné zakázky najdeme v zákoně o veřejných zakázkách, v § 7. Tam jsou stanoveny následující podmínky:

1. Zakázka musí být zadávána osobou, která je zadavatelem veřejných zakázek.
2. Zakázka musí zahrnovat prvek úplaty na straně zadavatele (byť i jen potenciální).
3. Musí se jednat o zakázku na dodávky, služby nebo stavební práce.

Veřejnou zakázkou je ale také i zakázka, která zahrnuje jen potenciální prvek úplaty ze strany zadavatele, přičemž prvekm úplaty může být například vázání na splnění určité podmínky. Dalším typem veřejné zakázky může být i taková zakázka, která se řadí mezi zákonné výjimky a zadavatel není povinen takové zakázky zadávat.

Dle svého předmětu mohou být VZ rozděleny na:

- veřejné zakázky na dodávky;
- veřejné zakázky na stavební práce;
- veřejné zakázky na služby.

VZ můžeme také rozdělit na VZ nadlimitní, podlimitní, nebo VZ malého rozsahu. Pro toto dělení je třeba vycházet z předpokládané hodnoty veřejné zakázky. Pro realizaci nadlimitní nebo podlimitní VZ má zadavatel povinnost s dodavatelem uzavřít písemnou smlouvu.

Kategorie	Zadavatel	VZ na dodávky	VZ na služby	VZ na staveb. práce
Nadlimitní VZ	ČR a státní příspěvkové organizace	3 395 000 Kč	3 395 000 Kč	131 402 000 Kč
	Územně samosprávné celky, příspěvkové organizace	5 244 000 Kč	5 244 000 Kč	131 402 000 Kč
	Sektorový zadavatel	10 489 000 Kč	10 489 000 Kč	131 402 000 Kč
	Zadavatelé uvedení v § 2 odst. 2 nebo § 2 odst. 6 zákona o VZ	10 489 000 Kč	10 489 000 Kč	131 402 000 Kč
Podlimitní VZ		≥ 2 000 000 Kč	≥ 2 000 000 Kč	≥ 6 000 000 Kč
VZ malého rozsahu		0 - 2 000 000 Kč	0 - 2 000 000 Kč	0 - 6 000 000 Kč

Tab. 4 - Finanční limity VZ platné k 1.1.2014 [12]

1.3.3 Zadávací řízení

Zadávací řízení může nastat až v případě, jsou-li splněny tyto kroky:

1. zadavatel určil, že se jednalo o VZ;
2. dále určil druh VZ;
3. stanovil předpokládanou hodnotu VZ a bude se jednat o nadlimitní VZ nebo podlimitní VZ;
4. zadavatel zjistil, že na danou VZ nešlo použít zákonnou výjimku.

Zadávací řízení upravuje závazný procesní postup zadavatele při zadávání veřejné zakázky.

Zákon definuje šest druhů zadávacích řízení:

1. otevřené řízení,
2. užší řízení,

3. jednací řízení s uveřejněním,
4. jednací řízení bez uveřejnění,
5. soutěžní dialog,
6. zjednodušené podlimitní řízení.

K zadání VZ může také docházet při:

7. řízení, v němž veřejný zadavatel zadává VZ na základě rámcové smlouvy,
8. v dynamickém nákupním systému.

1.3.4 Zadávání VZ

Zadávání veřejných zakázek je upraveno zákonem č. 137/2006 Sb., o veřejných zakázkách. Tímto zákonem je komplexně upravena oblast veřejného zadávání v České republice, především v návaznosti na evropské právní předpisy. Evropské zadávací směrnice (potažmo také Smlouva o založení Evropského společenství) stanovují principy transparentnosti, rovného zacházení, zákazu diskriminace, vzájemného uznávání a proporcionality, jako hlavní principy výše zmíněných zákonů [12].

1.3.5 Vymezení elektronických prostředků

Elektronickými prostředky pro účel tohoto zákona jsou sítě a služby elektronických komunikací. Na rozdíl od zákona č. 127/2005 Sb., o elektronických komunikacích, se jako elektronický prostředek nepovažuje fax.

Sítí elektronických komunikací se rozumí přenosové systémy, spojovací nebo směrovací zařízení a jiné prostředky, které umožňují přenos signálů po vedení, rádiiem, optickými nebo jinými elektromagnetickými prostředky, včetně družicových sítí, pevných sítí s komutací okruhů nebo paketů a mobilních zemských sítí, sítí pro rozvod elektrické energie v rozsahu, v jakém jsou používány pro přenos signálů, sítí pro rozhlasové a televizní vysílání a sítí kabelové televize.

Službou elektronických komunikací se rozumí služba poskytovaná za úplatu (obvykle), která spočívá v přenosu signálů po sítích elektronických komunikací, včetně telekomunikačních služeb a přenosových služeb v sítích používaných pro rozhlasové a televizní vysílání a v sítích kabelové televize, s výjimkou služeb, které nabízejí obsah prostřednictvím sítí a služeb elektronických komunikací nebo vykonávají redakční dohled nad obsahem přenášeným sítěmi a poskytovaným službami elektronických komunikací.

Zadavatel je povinen poskytnout dodavatelům, kteří mohou mít zájem účastnit se řízení či soutěže o návrh, k dispozici veškeré informace technické povahy, včetně nezbytného kódování a šifrování.

Dodavatel má v některých případech povinnost opatřit datovou zprávu zaručeným elektronickým podpisem založeným na kvalifikovaném certifikátu. Těmito právními úkony jsou:

- nabídka;
- žádost o účast;
- námitky proti úkonům zadavatele;
- prokazování splnění kvalifikace elektronickými prostředky;
- návrh v soutěži o návrh.

Tuto povinnost, opatřit datovou zprávu zaručeným elektronickým podpisem založeným na kvalifikovaném certifikátu, má také zadavatel a to v konkrétních úkonech:

- vyhlášení k uveřejnění ve Věstníku veřejných zakázek;
- výzva o zahájení zadávacího řízení;
- výzva k jednání;
- výzva k podání nabídky v zadávacím řízení či k účasti v soutěžním dialogu;
- rozhodnutí o výběru nejvhodnější nabídky (oznámení o zadání VZ);
- oznámení o způsobu vyřízení námitek;
- rozhodnutí o nejvhodnějším návrhu v soutěži o návrh.

Zadavatel může požadovat, aby datová zpráva byla elektronickým podpisem založeným na kvalifikovaném certifikátu opatřena u kterýchkoliv datových zpráv zasláných elektronickou formou [10][13].

1.3.6 Požadavky na elektronické nástroje

Zadavatel musí u elektronických nástrojů zajistit, aby:

- a) byly splněny požadavky na elektronické podpisy, vztahující se k nabídkám, žádostem o účast a k předávání plánů a projektů, uvedené v zákoně o elektronických komunikacích;
- b) mohl být přesně určen čas a datum doručení nabídek, žádostí o účast a předložení plánů projektů;

- c) bylo možné přiměřeně zajistit, že před stanovenými lhůtami nikdo nemůže mít přístup k údajům zaslaným v souladu s těmito požadavky;
- d) v případě porušení zákazu přístupu k zaslaným údajům mohlo být přiměřeně zajištěno, že porušení bude spolehlivě zjistitelné;
- e) pouze oprávněné osoby mohly stanovit nebo změnit data pro zpřístupnění doručených údajů;
- f) během různých fází zadávacího řízení nebo soutěže o návrh, byl přístup ke všem nebo k části předaných údajů možný pouze na základě předchozího rozhodnutí oprávněných osob;
- g) rozhodnutí oprávněných osob o přístupu ke všem nebo k části předaných údajů mohlo umožnit přístup k předaným až po předem stanoveném datu;
- h) údaje doručené a zpřístupněné v souladu s těmito požadavky, mohly zůstat přístupné pouze osobám, které jsou oprávněné se s nimi seznamovat;
- i) byly chráněny proti neoprávněnému přístupu třetích osob;
- j) byla pro ně zajištěna technická podpora a servis v případě poruchy;
- k) použitím elektronických nástrojů zadavatel neporušuje zásady rovného přístupu a zákazu diskriminace;
- l) použité elektronické nástroje jsou obecně dostupné a slučitelné s běžně užívanými informačními a komunikačními technologiemi;
- m) zadavatel poskytne dodavatelům veškeré informace technické povahy, které jsou nezbytné pro komunikaci prostřednictvím elektronických nástrojů.

Splnění těchto požadavků lze prokázat tzv. certifikátem shody, který je vydáván subjektem posuzování shody anebo i jiným způsobem, například znaleckým posudkem. Zadavatel může být veřejný, dotovaný anebo sektorový [10].

1.3.7 Elektronizace veřejných zakázek

Komplexní rámec pro elektronické zadávání veřejných zakázek poskytují směrnice Evropského parlamentu a Rady č. 2004/18/ES, o koordinaci postupů při zadávání veřejných zakázek na stavební práce, dodávky a služby a č. 2004/17/ES, o koordinaci postupů při zadávání zakázek subjekty působícími v odvětví vodohospodářství, energetiky, dopravy a poštovních služeb.

Směrnice stanovují pravidla pro elektronické nabídkové řízení a určují podmínky pro moderní nákupní metody založené na elektronických komunikačních prostředcích.

Tyto směrnice, včetně komplexního rámce elektronického zadávání veřejných zakázek, byly přeneseny do zákona o veřejných zakázkách.

Vzhledem k tomu, že Evropská komise svým Akčním plánem požadovala po členských státech zpracování národních plánů zavedení elektronického zadávání veřejných zakázek, schválila vláda České republiky Národní plán zavedení elektronického zadávání veřejných zakázek pro období let 2006 až 2010.

Národní plán je strategickým dokumentem vlády České republiky pro oblast zavádění moderních informačních a komunikačních technologií do procesu zadávání VZ. Elektronické zadávání se vztahuje i na veřejné zakázky financování či spolufinancování ze strukturálních fondů EU. Po ukončení účinnosti Národního plánu se stala novým strategickým dokumentem vlády ČR pro oblast zavádění moderních informačních a komunikačních technologií do procesu zadávání veřejných zakázek Strategie elektronizace veřejných zakázek pro období let 2011 až 2015.

Elektronické zadávání veřejných zakázek je úzce svázáno s cíli strategie Smart Administration pro období 2007 až 2015, která se snaží o zefektivnění veřejné správy a poskytovaných veřejných služeb. Mezi cíle strategie patří:

- racionalizace administrativní procedury s cílem zajistit větší efektivitu a transparentnost;
- minimalizace byrokratických prvků uvnitř veřejné správy;
- zavedení systému strategického plánování ve státní správě a zajištění jeho provázanosti na finanční řízení;
- vytvoření jednoduššího prostředí pro podnikatele;
- realizace preventivních opatření v boji s korupcí.

Charakteristické vazby strategie Smart Administration a procesu elektronizace zadávání veřejných zakázek:

- V rámci elektronizace zadávání VZ bude realizována procesní podpora zadávacích zařízení VZ. Tento procesní pohled povede k optimalizaci životního cyklu VZ u jednotlivých zadavatelů. Rovněž dojde k optimalizaci jednotlivých úkonů zadávacích řízení tak, aby je bylo možno činit plně elektronicky s podstatně nižšími transakčními náklady na straně veřejné správy i dodavatelů.

- Elektronizace zadávání VZ povede k výraznému urychlení komunikace v průběhu zadávacích řízení. Dojde ke snížení administrativní zátěže dodavatelů a snížení transakčních nákladů.
- Elektronizace zadávání VZ umožní snazší plánování VZ. Zadavatelům bude poskytnuta informační podpora při plánování jednotlivých VZ i skupin VZ (tj. plánování za resort, organizaci, jednotlivé útvary organizace). Rovněž bude možnost využívat ucelenou evidenci zadávacích řízení v organizaci/resortu zadavatele pro účely řízení, monitoringu, statistických rozborů, analýz o skladbě prováděných nákupů.
- Pro podnikatele bude elektronizace zadávání VZ znamenat snížení formálních požadavků na komunikaci a provádění dílčích úkonů v zadávacím řízení. Např. zadávací podmínky budou poskytovány v elektronické podobě, bude umožněno podávat nabídku v podobě elektronického katalogu, což povede ke snížení transakčních nákladů dodavatele souvisejících s jeho účastí v zadávacím řízení.
- Plně elektronická zadávací řízení, tj. s využitím automatické metody hodnocení, významně sníží prostor pro manipulaci s výsledkem hodnocení nabídek. Jedním z prvků v rámci elektronizace VZ je i rejstřík dodavatelů se zákazem plnění VZ (tzv. black list) [14][15].

1.3.8 Problémy se zadáváním VZ

Hodnota veřejných zakázek je obvykle značná, což s sebou nese velké korupční příležitosti. Důsledkem korupčního jednání jsou předražené zakázky, nevhodné nakládání s veřejnými prostředky, méně kvalitní realizované služby a zvýšení nedůvěry veřejnosti ve veřejnou správu.

2 DATABÁZOVÉ ENGINY

Následující kapitola je zaměřena na analýzu v současnosti využívaného databázového systému Oracle, nově zvoleného databázového systému Microsoft SQL Server a především na odlišnosti obou DB enginů.

DB engine v obecném pojetí je služba pro ukládání, zabezpečení a manipulaci s daty. Poskytuje kontrolovaný přístup a nástroje pro rychlé transakční zpracování dat takovým způsobem, aby odpovídal požadavkům podnikových aplikací, které s těmito daty pracují.

DB engine se využívá pro tvorbu relačních databází pro OLTP (On-line Transaction Processing) nebo OLAP (On-line Analytical Processing). Tento proces zahrnuje tvorbu tabulek pro ukládání dat a dalších databázových objektů jako jsou indexy, pohledy nebo uložené procedury pro náhled, zabezpečení nebo manipulaci s daty.

2.1 Databázový engine Oracle

První databázový engine byl americkou společností Oracle uveden na trh v roce 1978. Jednalo se tehdy o verzi Oracle V1 napsanou v assembleru. V současnosti je nejvyšší uvedenou verzí verze 12c, která byla představena v roce 2013 [16][17].

2.1.1 Verze Oracle DB

V tabulce 5 je soupis databázových enginů vydaných společností Oracle od roku 1985. První verze byla v roce 1978 napsaná v assembleru a běžela na stroji, který měl 128 kB paměť. Tato verze nebyla nikdy uvedena oficiálně. Pro aplikaci PAVEZA volíme verzi 12c z roku 2013.

Verze	Rok vydání	Význam označení
12c	2013	c = cloud
11g	2007	g = grid computing ready
10g	2003	g = grid computing ready
9i	2001	i = internet
8i	1999	i = internet
8	1997	
7	1992	
6	1988	
5	1985	

Tab. 5 - Přehled verzí Oracle databáze od roku 1985

2.1.2 Oracle DB 12c

Tato verze databázového enginu Oracle je v současnosti v aplikaci PAVEZA brána jako referenční a všechny dotazy, které jsou v aplikaci sestavovány, jsou optimalizovány pro jazyk PL/SQL se všemi jeho rozšířeními uvedenými společně s Oracle databází 12c. V následujícím seznamu je soupis vybraných změn, které s sebou verze 12c přinesla:

1. **Neviditelné sloupce** – nyní mají uživatelé možnost definovat sloupec jako neviditelný. Do verze 12c se bylo možno setkat pouze se systémovými neviditelnými sloupci, s jejichž viditelností není možné manipulovat. Pro zobrazení nebo nastavení hodnoty neviditelného sloupce, musí uživatel explicitně zadat jeho jméno. Například dotaz `SELECT * FROM tabulka`, zobrazí pouze sloupce, které jsou definované jako viditelné. Jako neviditelný může být nastavený i virtuální sloupec, není však možné, aby neviditelný sloupec obsahovaly dočasné nebo externí tabulky.
2. **Příkazy FETCH FIRST ROWS a OFFSET** – Oracle představil novou syntaxi pro limitování řádku ve výsledku PL/SQL dotazu. Například dotaz `SELECT * FROM tabulka ORDER BY sloupec FETCH FIRST 5 ROWS ONLY`, vrátí uživateli ve výsledku pouze prvních 5 řádků dle požadovaného řazení. Využití příkazu `OFFSET` můžeme demonstrovat například na dotazu `SELECT * FROM tabulka ORDER BY sloupec OFFSET 2 ROWS FETCH NEXT 5 ROWS ONLY`, který uživateli ve výsledku vrátí třetí až sedmý řádek dle požadovaného řazení.
3. **Vícenásobné indexy** – před verzí 12c nebylo možné vytvořit více indexů na jeden sloupec nebo na stejnou skupinu sloupců. Pokud byl vytvořen index například nad sloupcem `a`, nebo nad sloupci `a,b`, nebylo už možné vytvořit další index nad sloupcem `a`, nebo sloupci `a a b` ve stejném pořadí. Ve verzi 12c můžeme takto definovat více indexů, pokud jsou různého typu (`INDEX`, `BITMAP INDEX` apod.). V každém okamžiku je však viditelný (tj. použitelný) pouze jeden z těchto indexů.
4. **Kaskádový TRUNCATE** – před verzí 12c nebylo možné přímo zavolat `TRUNCATE` nad rodičovskou (master) tabulkou, pokud existovaly tabulky z ní odvozené, které obsahovaly nějaká data. Ve verzi tato možnost je díky příkazu `TRUNCATE TABLE` s volbou `CASCADE`, který volá rekurzivně `TRUNCATE` nad všemi tabulkami odvozenými od master tabulky.

2.1.3 PL/SQL

Tento jazyk je procedurální nástavbou standardizovaného SQL. Byl vyvinut společností Oracle a v Oracle databázích se využívá od verze 7. S každou další vydanou verzí Oracle databáze přichází obvykle i nějaké nové rozšíření jazyka PL/SQL.

PL/SQL umožňuje pracovat s procedurálními prvky, jako jsou podmínky a cykly, uživatel může deklarovat konstanty a proměnné, procedury a funkce, uživatelské typy a proměnné těchto typů, trigery a další. Databázový engine Oracle umožňuje práci s výjimkami (chyby, které nastanou za běhu), prostřednictvím PL/SQL kolekcí je také možno pracovat s poli. Od verze 8 se Oracle zaměřil na orientaci na objekty, v databázi mohou být od této verze ukládány procedury, funkce, balíčky, typy a trigery tak, aby mohly být znovupoužity jakoukoliv aplikací, která se k databázi připojí.

2.2 Databázový engine SQL Server

Americká společnost Microsoft od roku 1989 pravidelně uvádí na trh nové verze tohoto databázového engine. V současnosti je dostupná nejvyšší verze 2014 [18][19].

2.2.1 Verze SQL Server

Následující tabulka obsahuje přehled vydaných verzí DB engine SQL Server od roku 1995. Vůbec první verze engine SQL Server byla uvedena už v roce 1989, jednalo se tehdy o 16 bitovou verzi 1.00. Jako cílová DB platforma pro aplikaci PAVEZA je zvolena verze SQL Server 2012 vydaná téhož roku.

Verze	Rok vydání	Název	Kódové označení
12.00	2014	SQL Server 2014	Hekaton
11.00	2012	SQL Server 2012	Denali
10.50	2010	SQL Server 2008 R2	Kilimanjaro
10.00	2008	SQL Server 2008	Katmai
9.00	2005	SQL Server 2005	Yukon
8.00	2000	SQL Server 2000	Shiloh
7.00	1998	SQL Sever 7.0	Sphinx
6.50	1996	SQL Server 6.5	Hydra
6.00	1995	SQL Server 6.0	SQL95

Tab. 6 - Přehled verzí SQL Serveru od roku 1995

2.2.2 SQL Server 2012

Tato verze engine SQL Server byla vybrána jako cílová při přechodu aplikace PAVEZA na nový databázový engine. Vytvářené databázové dotazy a struktury budou optimalizovány pro tuto verzi. Zde je soupis vybraných novinek, které s sebou tato verze SQL Serveru přinesla:

1. **Non-Clustered Columnstore indexy** – v relačních databázích jsou data ukládána po řádcích do datových stránek o velikosti 8 kB. Mohou a nastávají tak situace, kdy jsou dotazovány data z jednoho sloupce, ke kterým je třeba provést přístup přes několik datových stránek. Non-Clustered Columnstore indexy umožňují uložení dat nikoliv po stránkách, ale po sloupcích, což může přinést v některých případech značné zvýšení výkonu při získávání dat.
2. **Sekvence** – tento typ objektu už nějakou dobu existuje v Oracle databázích, Microsoft ho představil až ve verzi 2012. Jedná se o objekt, který nám umožňuje vygenerovat sekvenci jedinečných čísel dle zadané specifikace. Nese podobné znaky jako identifikátor tabulky (sloupec `Identity`), na rozdíl od identifikátoru je ale sekvence nezávislá na tabulce.
3. **Throw** – od verze SQL Server 2005 lze využívat zpracování chyb pomocí bloků `Try` a `Catch`. Problematickým místem však nadále byla propagace chyby ke klientovi. K vyvolání zprávy o chybě se používal příkaz `RAISERROR`. Ten sice uměl generovat jak systémové tak uživatelské výjimky, pokud chtěl ale vývojář přidat vlastní popis výjimky, musel předtím vytvořit záznam v systémové tabulce `sys.messages`. S verzí 2012 přichází příkaz `Throw`, který umožňuje zpropagovat výjimku až ke klientovi a přidat jí vlastní popis bez nutnosti vytváření záznamu v systémových tabulkách.
4. **Uživatelsky definované role** - v předešlých verzích SQL serveru jsme byli omezeni předdefinovanými rolemi, které nám připravili vývojáři Microsoftu. Od verze 2012 jsme schopni definovat vlastní uživatelské role, čímž se podstatným způsobem zvýšila škálovatelnost oprávnění a obecně správy uživatelů.

2.2.3 T-SQL

Jedná se o proprietární rozšíření standardizovaného dotazovacího jazyka SQL od společností Microsoft a Sybase. T-SQL rozšiřuje SQL o procedurální programování, možnost deklarace

lokálních proměnných, funkce pro práci s řetězci a daty, matematické funkce, upravuje příkazy DELETE a UPDATE a přináší další rozšíření.

Jazyk T-SQL je používán pro komunikaci s databázovým enginem Microsoft SQL Server a všechny aplikace, které chtějí komunikovat s tímto enginem, tak činí prostřednictvím dotazů, které odpovídají syntaxi jazyka T-SQL.

2.3 Odlišnosti databází Oracle a SQL Server

V této práci nás budou zajímat především syntaktické odlišnosti při sestavování dotazů a definování databázových objektů nad oběma databázovými enginy. V této části se tedy zaměříme na konkrétní syntaktické rozdíly v obou využívaných mutacích (T-SQL, PL/SQL) dotazovacího jazyka SQL.

2.3.1 Datové typy

První zleva je v této části vždy uveden datový typ využívaný SQL Serverem, jako druhý jeho ekvivalent z Oracle databáze.

INTEGER x NUMBER(10)

- `INTEGER` – 4 Bytové číslo, skládá se z 31 bitů a jednoho bitu pro znaménko, od verze SQL Server 5 lze použít také zkratku `INT`.
- `NUMBER(10)` – lze využít jako identifikátor tabulky, může obsahovat čísla v rozsahu -2^{31} až 2^{31} , číslovka 10 znamená, že do tohoto typu jsme schopni uložit jakékoliv celé číslo zapsané pomocí 10 číslic.

SMALLINT x NUMBER(6)

- `SMALLINT` – 2 Bytové číslo, skládá se z 15 bitů a jednoho bitu pro znaménko.
- `NUMBER(6)` – lze využít jako identifikátor tabulky, může obsahovat čísla v rozsahu -2^{15} až 2^{15} .

TINYINT x NUMBER(3)

- `SMALLINT` – 1 Bytové číslo, skládá se z 8 bitů, pro znaménko není určen žádný bit.
- `NUMBER(3)` – může obsahovat celá čísla v rozsahu 0 až 255.

REAL x FLOAT

- REAL – číslo s plovoucí desetinnou čárkou uložené na 4 Bytech. Hodnoty se mohou pohybovat v rozmezí $-3,40^{38}$ až $3,40^{38}$.
- FLOAT – podle ANSI je k typu REAL v databázích Oracle ekvivalentem datový typ FLOAT (63). Parametr u datového typu FLOAT znamená binární přesnost a může se pohybovat v rozmezí 1 až 126 (výchozí je 126). Kolik čísel v desítkové soustavě jsme schopni v daném datovém typu zobrazit, zjistíme vynásobením binární přesnosti číslem 0,30103 a zaokrouhlením na nejbližší vyšší celé číslo. V Oracle DB ovšem také datový typ NUMBER ukládá jak čísla s pevnou, tak čísla s plovoucí desetinnou čárkou.

BIT x NUMBER(1)

- BIT – pro uložení logické hodnoty True nebo False (0/1). Využívá jeden bit z Bytového čísla. Tento datový typ nemůže být nastaven na hodnotu NULL (kromě verze SQL Server 7, kde bylo toto umožněno).
- NUMBER (1) – v Oracle DB se BIT ukládá buď jako NUMBER (1) nebo CHAR.

CHAR(n) x CHAR(n)

- CHAR (n) – pro uložení řetězce pevné délky. Skládá se přesně z n 8 bitových znaků. Lze také zapsat jako CHARACTER. Hodnota n může nabývat hodnot 1 až 8000.
- CHAR (n) – maximální velikost 2000, větší řetězce jsou např. při migraci pomocí nástroje Oracle Migration Workbench konvertovány na VARCHAR2.

VARCHAR(n) x VARCHAR2(n)

- VARCHAR (n) – pro uložení řetězce proměnné délky. Maximální velikost je n 8 bitových znaků. Hodnota n může nabývat hodnot 1 až 8000.
- VARCHAR2 (n) – řetězec proměnné délky. Spolu s řetězcem je uložena také informace o jeho délce. Maximální velikost 32 767 Bytů.

TEXT x CLOB

- TEXT – zastaralý datový typ pro uložení řetězců až o $2^{31} - 1$ znacích.
- CLOB – do sloupce tohoto typu může být uloženo až 4GB dat.

IMAGE x BLOB

- IMAGE – zastaralý datový typ pro uložení binárních dat o velikosti až $2^{31} - 1$ Bytů.
- BLOB – do sloupce tohoto typu může být uloženo až 4GB dat.

DATETIME x DATE

- DATETIME – tento datový typ je uložen jako dvě celá 4 Bytová čísla. První část obsahuje datum, přičemž povolená data jsou z rozmezí 1. ledna 1753 až 31.12.9999. Část obsahující čas může nabývat hodnot 0 až 25 920 000. Přesnost času je 3,33 milisekund. Výchozí hodnota pro sloupce tohoto typu je 1.ledna 1990 00:00:00.000.
- DATE – ve srovnání s datovým typem SQL Serveru DATETIME má DATE menší přesnost (1 sekunda), není tedy vhodné ho využívat tam, kde chceme mít například uložen jedinečný datum a čas. V takovém případě je lepší možností využití typu TIMESTAMP, nebo rozdělení data a času do dvou sloupců, na základě kterých pak bude porovnávána jedinečnost.

MONEY x NUMBER(19,4)

- MONEY – datový typ pro uložení peněžních hodnot. Vnitřně je uložen jako dvě 4 Bytová celá čísla, jedna reprezentuje celou část a druhá desetinnou.
- NUMBER (19, 4) – pro uložení čísla skládajícího se maximálně z 19 číslic, z nichž 4 tvoří desetinnou část.

Datové typy SQL Serveru IMAGE a TEXT jsou zastaralé a v budoucích verzích SQL Serveru budou odstraněny. Sloupce tohoto typu jsou dále omezeny nemožností vytvářet nad nimi indexy, tyto sloupce nemohou být primárními klíči, nemohou být použity v klauzulích GROUP BY, ORDER BY, HAVING a DISTINCT, nemůže na ně být aplikován operátor LIKE, ani operace SUBSTR a LENGTH.

Na Microsoft SQL Serveru mohou hodnotu NULL ukládat jen sloupce, jejichž datový typ má proměnnou délku. Pokud vytvoříme sloupec, který povoluje hodnoty NULL a je datového typu s pevnou délkou, datový typ je automaticky konvertován na typ s proměnnou délkou [20].

2.3.2 DML

V této sekci se budeme věnovat syntaktickým rozdílům v DML (Data Manipulation Language). Zaměříme se především na základní, nejvíce používané dotazy typu SELECT, INSERT, UPDATE a DELETE [20].

Připojení k databázi

Microsoft SQL Server

```
USE database_name
```

Oracle

```
CONNECT user_name/password
```

```
SET role
```

Každému DB uživateli je přiřazena výchozí databáze, ke které je připojen po připojení na server. Příkazem USE se může uživatel přepínat mezi jednotlivými databázemi.

Microsoft SQL Server dovoluje uživatelům se na straně serveru přepínat mezi databázemi, na které má daný uživatel práva. V Oracle se lze připojit pouze k jedné databázi, koncept přepínání databází tu neexistuje. Uživatel se k jiné databázi může dostat změnou role (příkaz SET role) nebo znovuvoláním příkazu CONNECT.

SELECT bez FROM

Microsoft SQL Server

```
SELECT getdate()
```

Oracle

```
SELECT sysdate FROM dual;
```

Oracle nepodporuje SELECT bez FROM. V případě Oracle je ale možné využít tabulky DUAL, která vždy obsahuje právě jeden řádek a jeden sloupec a je ji možno využít v případech jako je např. výše uvedené získání aktuálního data a času.

SELECT INTO

Microsoft SQL Server

```
SELECT col1, col2, col3
```

```
INTO target_table
```

```
FROM source_table  
WHERE where_clause
```

Oracle

```
INSERT INTO target_table  
SELECT col1, col2, col3  
FROM source_table  
WHERE where_clause
```

Microsoft SQL Server dovoluje vložit záznamy do tabulky pomocí dotazu `SELECT INTO` (což je vlastně kombinace dotazů `SELECT` a `INSERT`). Tato konstrukce ale není podporovaná ANSI. V Oracle tedy musí být přepsaná způsobem, který je naznačen výše.

INSERT

Rozdíl mezi oběma enginy je v případě dotazu `INSERT` v klauzuli `INTO`, která je na SQL Serveru nepovinná na rozdíl od Oracle, kde se musí vždy vyskytovat. Na obou enginech mohou být v části `VALUES` použity funkce. Specifické funkce Oracle musí být adekvátním způsobem, je-li to možné, nahrazeny SQL Server funkcemi.

UPDATE

U tohoto typu dotazu je rozdíl v použití klauzule `FROM`, kterou Oracle v rámci `UPDATE` nedovoluje a takovýto dotaz musí být přepsán například tímto způsobem:

Microsoft SQL Server

```
UPDATE t  
SET t.pub_id = p.pub_id  
FROM titles t, publishers p  
WHERE t.title LIKE 'C%' AND p.pub_name = 'new age'
```

Oracle

```
UPDATE titles  
SET pub_id =  
( SELECT a.pub_id FROM publishers a
```

```
WHERE publishers.pub_name = 'new age')  
WHERE titles.title like 'C%'
```

DELETE

Dotaz DELETE může na SQL Server obsahovat dvě klauzule FROM, první je nepovinná a druhá má podobnou funkcionalitu jako klauzule FROM u dotazu UPDATE. Oracle povoluje pouze první klauzuli FROM, která je stejně jako na SQL Server nepovinná. Přepis dotazu se dvěma klauzulemi FROM může vypadat například takto:

Microsoft SQL Server

```
DELETE  
  
FROM sales  
  
FROM sales, titles  
  
WHERE sales.title_id = titles.title_id  
      AND titles.type = 'business'
```

Oracle

```
DELETE  
  
FROM sales  
  
WHERE title_id in  
  
(   SELECT title_id  
     FROM titles  
     WHERE type = 'business'  
  )
```

II. PRAKTICKÁ ČÁST

3 NÁVRH A REALIZACE PŘECHODU NA NOVÝ DB ENGINE

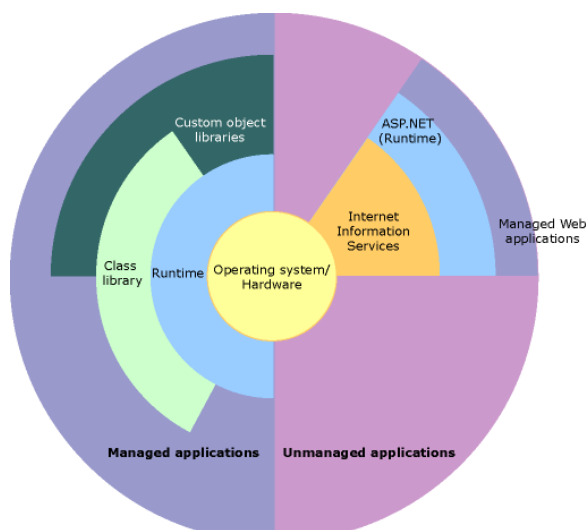
V této kapitole se zaměříme na popis technologií a nástrojů, které budou při implementaci využity a návrh konkrétních postupů při přechodu na nový databázový engine. Poslední část této kapitoly obsahuje popis implementace řešení včetně ukázek zdrojového kódu.

3.1 Použité technologie

3.1.1 .NET Framework

Jedná se o technologii umožňující překlad a běh nové generace aplikací a XML webových služeb. Aplikace PAVEZA je napsaná pro .NET Framework verzi 4. Framework je navržen tak, aby splňoval následující požadavky:

- poskytnutí konzistentního, objektově-orientovaného vývojového prostředí, které je nezávislé na tom, zda je zdrojový kód objektů uložen a spouštěn lokálně, spouštěn lokálně ale uložen v síti, nebo je spouštěn vzdáleně;
- poskytnutí vývojového prostředí, které minimalizuje problémy s nasazením a verzováním software;
- poskytnutí vývojového prostředí, které zaručí bezpečné spuštění kódu a to včetně kódu, u kterého je neznámý autor, nebo je autorem třetí strana, která je nedůvěryhodná;
- zajištění stejných vývojových metodik napříč různými typy aplikací, např. WPF aplikace a webové aplikace [21].



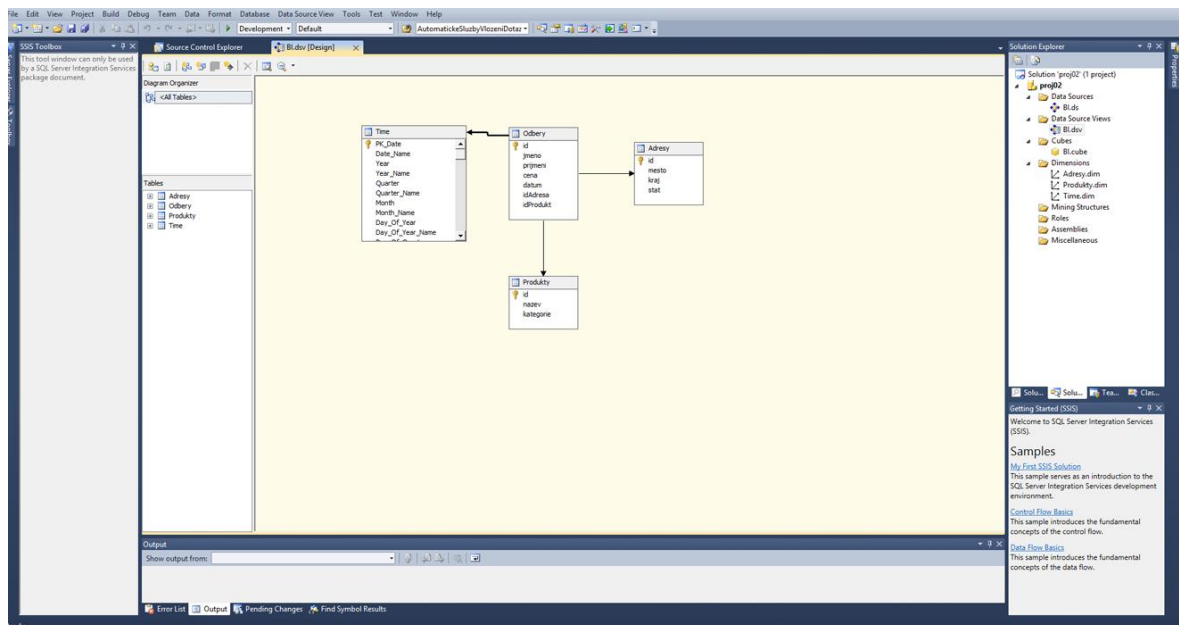
Obr. 7 - Obecný pohled na .NET Framework [21]

3.2 Použité vývojové nástroje

3.2.1 Microsoft Visual Studio

Balík nástrojů a služeb, který je určen k vývoji softwarových aplikací pro desktopové i dotykové prostředí Windows, pro web, SharePoint, mobilní zařízení i cloud prostředí. Lze vytvářet také multiplatformní HTML5 aplikace. Pro Visual Studio je také dostupná celá řada doplňků a rozšíření. Toto vývojové prostředí je dostupné v několika edicích:

- **Professional** – nejnižší edice pro jednotlivce a junior programátory;
- **Test Professional** – Speciální edice pro profesionální testery v týmech;
- **Premium** – Tvůrci komplexních aplikací, jednotlivci i pracovníci v týmech;
- **Ultimate** – Senior vývojáři, systémoví architekti, QA manažeři a vedoucí týmů;
- **Team Foundation Server** – pro správu kódu a kompletní řízení procesu vývoje.

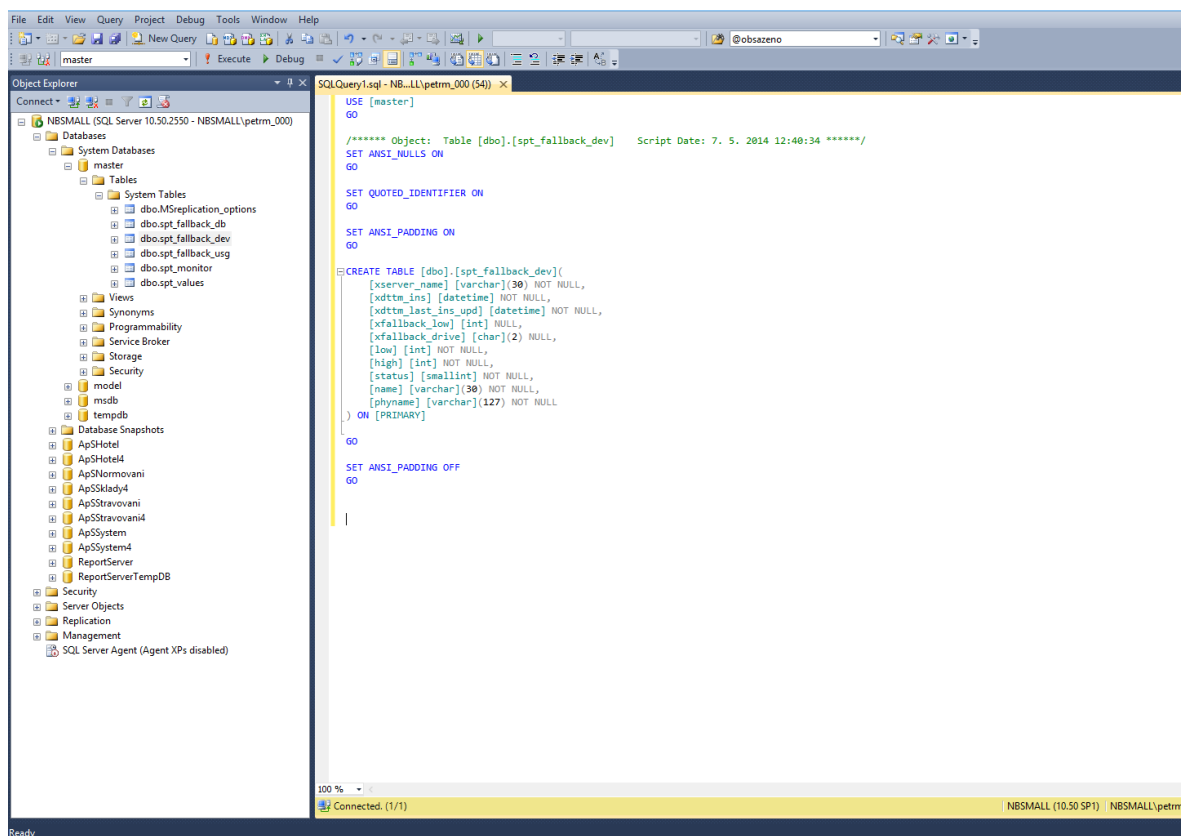


Obr. 8 - Vývojové prostředí Microsoft Visual Studio 2010

Visual Studio podporuje celou řadu programovacích jazyků, jako jsou např. C, C++, C#, F#, Visual Basic a další. V implementační části této práce bylo využito Microsoft Visual Studia 2010 s Team Foundation Serverem a programovacího jazyku C# [22].

3.2.2 SQL Server Management Studio

Integrované prostředí pro přístup, konfiguraci, administraci a vývoj všech komponent SQL Serveru. Kombinuje v sobě sadu grafických nástrojů a množství editorů skriptů, aby byl umožněn přístup a administrace SQL Serveru všem uživatelům a administrátorům bez ohledu na jejich dosavadní zkušenosti se správou SQL Serveru.

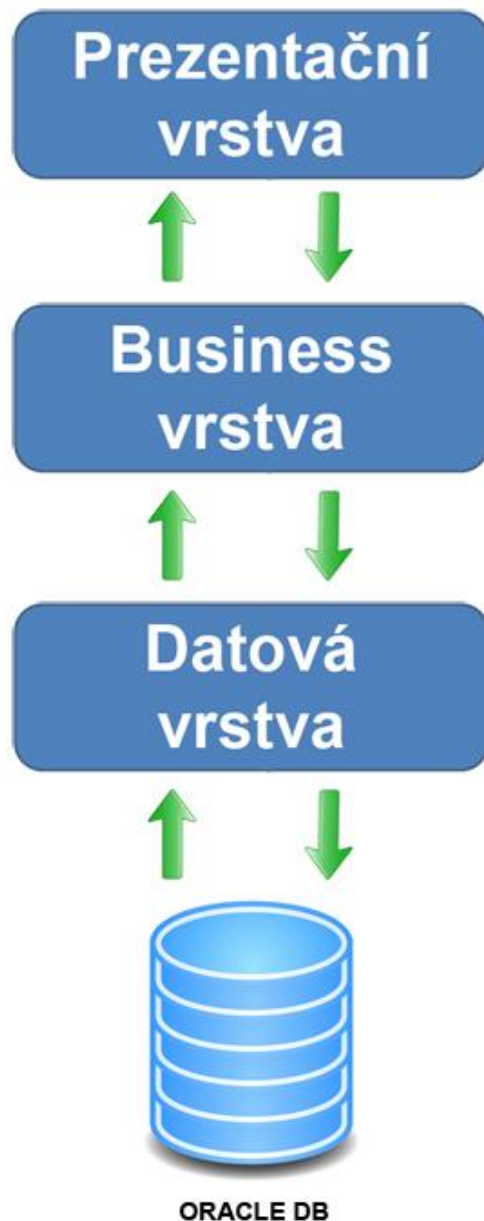


Obr. 9 - SQL Server Management Studio UI

Hlavní komponentou SSMS je prohlížeč objektů, který umožňuje uživateli prohlížet a provádět operace nad jakýmkoliv objekty na serveru. SSMS je možné získat také v „express“ verzi, která je k dostání zdarma. Od SQL Sever Management Studio verze 11 je toto prostředí přepsáno do WPF, změnil se i design UI, který se nyní více podobá UI Microsoft Visual Studia 2010 [23].

3.3 Návrh řešení

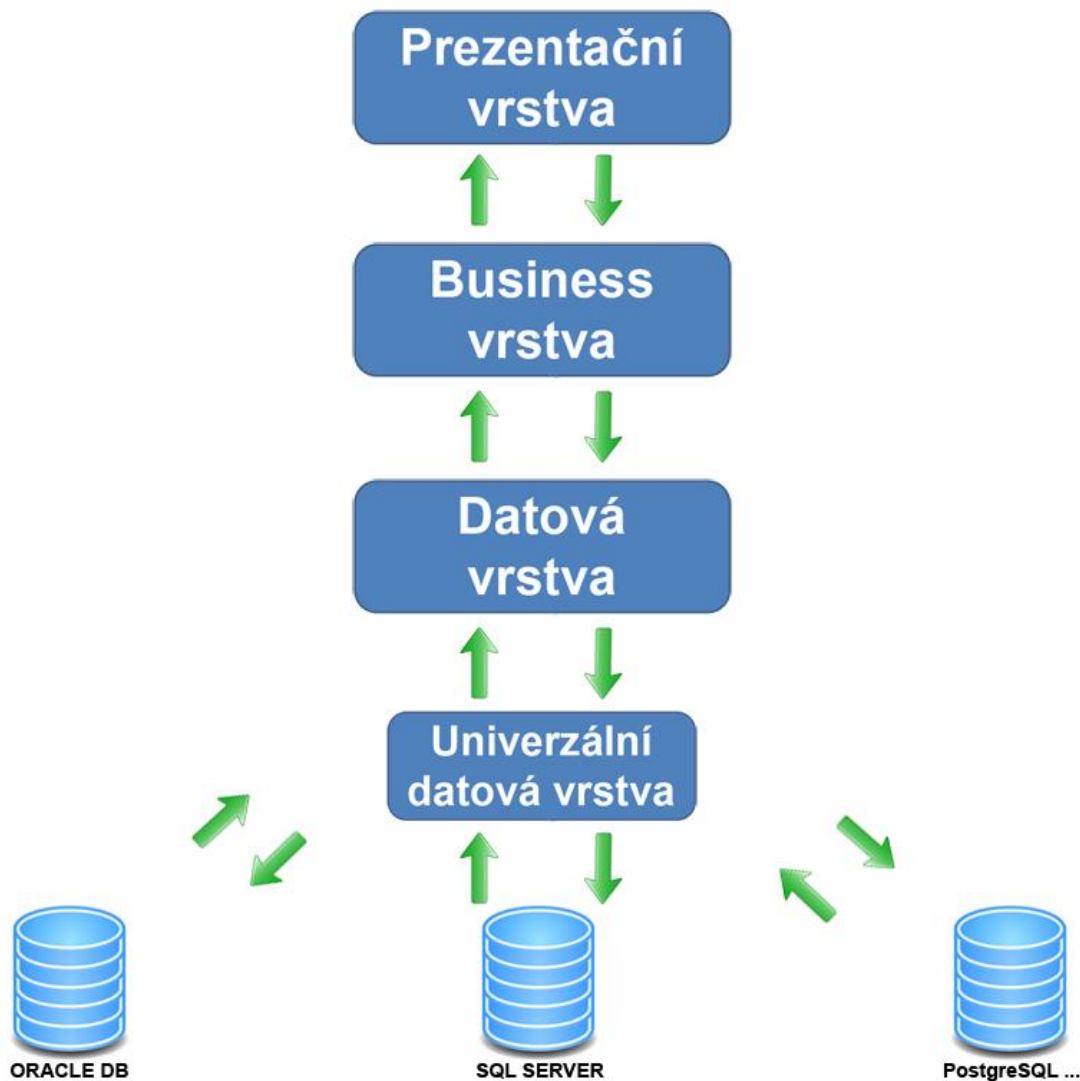
Na obrázku 10 je abstraktně znázorněna současná 3 vrstvá architektura aplikace PAVEZA. Na vrcholu architektury je prezentační vrstva, která se skládá z webových formulářů a



Obr. 10 - Současné zobrazení 3 vrstvé architektury aplikace PAVEZA

ovládacích prvků a zdrojového kódu, který tyto prvky obsluhuje. Business vrstva obsahuje celou aplikační logiku, dále jsou zde uloženy objekty reprezentující databázové tabulky a metody těchto objektů. Při volání většiny z těchto metod dochází k operacím s daty na úrovni databáze. Datová vrstva zahrnuje komunikaci s databázovým serverem a konkrétní databázový server.

Cílem je dosáhnout stavu, který je znázorněn na obrázku 11, tedy stavu, kdy je aplikace zcela nezávislá na typu použitého databázového engine.



Obr. 11 - Architektura aplikace PAVEZA s využitím univerzální datové vrstvy

Zvolil jsem variantu vytvoření univerzální datové vrstvy, která bude obsahovat objekty pro databázové operace a na základě globálního parametru, který bude určovat typ použitého databázového engine, a který bude nastavován pouze jedenkrát, a to při instalaci aplikace (případně při přechodu na jiný databázový engine), sestavovat databázové dotazy v syntaxi odpovídající dané SQL mutaci.

Při návrhu počítáme s existencí všech vrstev (kromě navrhované mezivrstvy) i s existencí objektů, které reprezentují databázové tabulky.

3.3.1 UniCommand

Objekt, který bude jako jednu z vlastností obsahovat SQL dotaz v textové podobě a metody pro volání dotazů nad databází a pro získání výsledků, je-li to vyžadováno. Ostatní objekty budou od tohoto objektu podděně.

Prostřednictvím tohoto objektu půjde také sestavit dotaz do řetězce přímo v kódu a to v případech, kdy nebude možné nebo vhodné využít připravených objektů a dotaz bude sestavitelný způsobem, který podporují všechny vybrané databázové enginy. Při zadání dotazu přímou formou, je třeba věnovat zvýšenou pozornost otázce bezpečnosti (např. zamezení SQL injection).

3.3.2 UniInsert

Vkládání záznamů do databáze bude probíhat prostřednictvím instancí třídy UniInsert. Třída bude obsahovat minimálně:

- název tabulky, do které záznam vkládáme;
- kolekci vkládaných záznamů, která uvnitř obsahuje jednotlivé vkládané atributy.

3.3.3 UniSelect

Objekt tohoto typu bude reprezentovat instanci dotazu `SELECT`. Bude nutné, aby obsahoval následující vlastnosti:

- název tabulky, nad kterou dotaz provádíme;
- kolekci názvů sloupců, nebo výrazů, které bude dotaz vracet;
- nastavení hodnoty pro limitaci počtu vrácených řádků;
- nastavení hodnoty offsetu pro limitaci počtu vrácených řádků;
- parametr pro zachování jedinečnosti výsledků (`DISTINCT`);
- kolekce joinovaných tabulek;
- kolekce CTE;
- kolekce agregací;
- kolekce omezení agregací (`HAVING`);
- kolekce podmínek;

- kolekce sloupců výsledku, dle kterých budeme výsledek řadit.

3.3.4 UniUpdate

Třída reprezentující dotaz UPDATE bude obsahovat minimálně tyto vlastnosti:

- název tabulky, ve které změnu provádíme;
- kolekce nastavovaných vlastností;
- kolekce joinovaných tabulek;
- kolekce podmínek.

3.3.5 UniDelete

Dotaz DELETE bude realizován prostřednictvím instancí třídy UniDelete jejíž minimální vlastnosti budou:

- název tabulky, ze které vybíráme záznamy určené ke smazání;
- kolekce joinovaných tabulek;
- kolekce podmínek.

3.3.6 UniFunction

U volání funkcí je třeba rozlišit volání funkce, jejímž výsledkem je tabulka, a který budeme s největší pravděpodobností ukládat do tabulkové struktury a volání funkce, jejímž výsledkem je skalární hodnota. Tato třída bude tedy obsahovat nejméně dvě metody volání funkce, dále bude obsahovat minimálně:

- název schématu/balíčku;
- název funkce;
- kolekce parametrů;

3.3.7 UniProcedure

Třída reprezentující volání procedur obsahuje:

- název procedury;
- kolekci parametrů.

3.4 Implementace navrženého řešení

Společnost NWT a.s. si nepřeje zveřejňovat konkrétní části implementace, ani jakákoliv databázová schémata. V této části tak uvedu principiální postup implementace nad abstraktními objekty, které nekorespondují, nebo vzdáleně korespondují s objekty použitými v aplikaci PAVEZA.

V rámci implementace byla přidána metadata k aktuálně vytvořeným třídám reprezentujícím databázové tabulky dle aktualizovaného databázového schématu a byla vytvořena knihovna obsahující objekty a metody univerzální datové vrstvy, které budou využívány pro generování dotazů do databází nezávisle na tom, jaký typ databázového enginu bude použit.

3.4.1 Implementace UniInsert

```
public class UniInsert {  
    public UniInsert(string TableName) {  
        ...  
    }  
    public UniInsert(object DBObj) {  
        ...  
    }  
    public string TableName;  
    public string AutoIncrementCol;  
    public colAttributes Attributes;  
    public int? InsertedID;  
}
```

Vložení záznamu do databázové tabulky je možné dvěma způsoby:

1. Vytvoření instance třídy `UniInsert`, předáním názvu tabulky, do které záznam budeme vkládat, v konstruktoru a nastavením kolekce vkládaných atributů.
2. Vytvoření instance třídy `UniInsert`, předáním databázového objektu v konstruktoru. Pokud se jedná o objekt reprezentující databázovou tabulku, má

třída tohoto objektu v metadatech uložené informace o názvu tabulky, názvech sloupců i informaci o tom, který sloupec je primárním klíčem a zda je jeho hodnota nastavovaná automaticky.

Pokud je vložení záznamu úspěšné a je nastavena vlastnost `AutoIncrementCol`, která nese název sloupce, který je primárním klíčem, jehož hodnota je generovaná na serveru automaticky, nastaví se do vlastnosti `InsertedID` hodnota tohoto primárního klíče. V případě, že bylo vložení prováděno prostřednictvím objektu, je i tomuto objektu nastavena hodnota primárního klíče (dle metadat třídy).

Instance třídy `UniInsert`, nastavená pro vložení záznamu do tabulky „VerejneZakazky“, bude vypadat následujícím způsobem:

```
UniInsert lInsert = new UniInsert("VerejneZakazky");
lInsert.AutoIncrementCol = "IDZakazky";
lInsert.Attributes.Add("Nazev", "Zakázka č. 1");
lInsert.Attributes.Add("Popis", "Popis zakázky č. 1");
lInsert.Attributes.Add("KontaktOsoba", "Ing. Josef Novák");
lInsert.Attributes.Add("JeAktivni", True);
lInsert.Execute();
```

Dotaz sestavený pro DB engine Oracle bude vypadat:

```
VAR new_id NUMBER(10);

INSERT INTO VerejneZakazky(Nazev,Popis,KontaktOsoba,JeAktivni)
VALUES ('Zakázka č. 1','Popis zakázky č. 1','Ing. Josef Novák',1)
RETURNING IDZakazky INTO :new_id;

SELECT :new_id FROM DUAL;
```

Dotaz sestavený pro DB engine SQL Server bude vypadat:

```
INSERT INTO VerejneZakazky (Nazev,Popis,KontaktOsoba,JeAktivni)
VALUES ('Zakázka č. 1','Popis zakázky č. 1','Ing. Josef Novák',1)
```



```
SELECT SCOPE_IDENTITY()
```

Tento jednoduchý dotaz vloží záznam do tabulky a zpět vrátí hodnotu primárního klíče vloženého záznamu, pokud je požadovaná. Jelikož v Oracle DB neexistuje přímý způsob, jak automaticky vkládat automaticky inkrementovanou hodnotu, bude tohoto dosaženo pomocí vytvořené sekvence, ze které bude generována rostoucí posloupnost jedinečných čísel a triggeru nad tabulkou, který zajistí jeho vložení při každém vytvoření záznamu v tabulce.

3.4.2 Implementace UniSelect

```
public class UniSelect {  
    public UniSelect(string TableName) {  
        ...  
    }  
    public string TableName;  
    public colColumns Columns;  
    public colCTE CTE;  
    public colJoinTables JoinTables;  
    public colGroupBy GroupBy;  
    public colHaving Having;  
    public colConditions Conditions;  
    public int? Limit;  
    public int? LimitOffset;  
    public bool Distinct;  
    public colOrderCols OrderBy;  
}
```

Pro vytvoření komplexních Select dotazů, je možné do sebe instance této třídy také vnořovat, vyhodnocení pak probíhá rekurzivně. Použití UniSelect demonstrujeme na následujícím příkladu:

```
UniSelect lSelect = new UniSelect("VerejneZakazky VZ ");
lSelect.Columns.Add("VZ.*")
lSelect.JoinTables.Add("Uzivatele U")
    .On("U.ID=VZ.IDUzivateleZadal");
lSelect.Conditions.AddCondGreaterOrEqual("VZ.Castka", 1e6);
lSelect.Conditions.AddCondEqual("U.Prijmeni", "Novák");
lSelect.Limit = 10;
lSelect.Limit = 5;
lSelect.OrderBy.Add("VZ.Castka", "DESC");
lSelect.Execute();
```

Dotaz sestavený pro DB engine Oracle bude vypadat:

```
SELECT VZ.*
FROM VerejneZakazky VZ
INNER JOIN Uzivatele U ON U.ID=VZ.IDUzivateleZadal
WHERE VZ.Castka>= 1000000 AND U.Prijmeni='Novák'
ORDER BY VZ.Castka DESC
OFFSET 5 ROWS FETCH NEXT 10 ROWS ONLY
```

Dotaz sestavený pro DB engine SQL Server bude vypadat:

```
WITH Result AS
(
    SELECT VZ.*, ROW_NUMBER() OVER(ORDER BY VZ.Castka DESC)
        AS RowNum
    FROM VerejneZakazky VZ
    INNER JOIN Uzivatele U ON U.ID= VZ.IDUzivateleZadal
    WHERE VZ.Castka>=1000000 AND U.Prijmeni='Novák'
    ORDER BY VZ.Castka DESC)
SELECT *
FROM Result
WHERE RowNum<@Offset+@Limit AND RowNum>=@Offset
```

3.4.3 Implementace UniUpdate

```
public class UniUpdate {  
    public UniUpdate (string TableName) {  
        ...  
    }  
    public UniUpdate (object DBObj) {  
        ...  
    }  
    public string TableName;  
    public string FromTableName;  
    public colAttributes Attributes;  
    public colJoinTables JoinTables;  
    public colConditions Conditions;  
}
```

Změna záznamu v databázové tabulce je proveditelná dvěma způsoby:

1. Vytvoření instance třídy `UniUpdate`, předáním názvu tabulky, do které záznam budeme vkládat, v konstruktoru a nastavením kolekce měněných atributů.
2. Vytvoření instance třídy `UniUpdate` a předáním databázového objektu v konstruktoru. Pokud se jedná o objekt reprezentující databázovou tabulku, má třída tohoto objektu v metadatech uložené informace o názvu tabulky, názvech sloupců i informaci o tom, který sloupec je primárním klíčem. Pomocí primárního klíče je vytvořena podmínka na výběr řádku a změněné jsou všechny atributy daného záznamu v databázi podle aktuálních hodnot nastavených v objektu.

Změnu záznamu budeme demonstrovat na následující instanci třídy `UniUpdate`. Cílem je zneaktivnit všechny zakázky vytvořené uživatelem Josefem Novákem:

```
UniUpdate lUpdate = new UniUpdate("VZ");
```

```
lUpdate.FromTableName = "VerejneZakazky VZ";  
lUpdate.Attributes.Add("JeAktivni", False);  
lUpdate.JoinTables.Add("Uzivatele U")  
.On("U.ID=VZ.IDUzivateleZadal");  
lUpdate.Conditions.AddCondEqual("U.Jmeno", "Josef");  
lUpdate.Conditions.AddCondEqual("U.Prijmeni", "Novák");  
lUpdate.Execute();
```

Dotaz sestavený pro DB engine Oracle bude vypadat:

```
UPDATE (  
    SELECT VZ.JeAktivni  
    FROM VerejneZakazky VZ  
    INNER JOIN Uzivatele U ON U.ID=VZ.IDUzivateleZadal  
    WHERE U.Jmeno='Josef' AND U.Prijmeni='Novák'  
) VZ  
SET VZ.JeAktivni=0
```

Dotaz sestavený pro DB engine SQL Server bude vypadat:

```
UPDATE VZ  
SET VZ.JeAktivni=0  
FROM VerejneZakazky VZ  
INNER JOIN Uzivatele U ON U.ID= VZ.IDUzivateleZadal  
WHERE U.Jmeno='Josef' AND U.Prijmeni='Novák'
```

3.4.4 Implementace UniDelete

```
public class UniDelete {  
    public UniDelete (string TableName) {  
        ...  
    }  
}
```

```
public UniDelete (object DBObj) {  
    ...  
}  
  
public string TableName;  
  
public string FromTableName;  
  
public colJoinTables JoinTables;  
  
public colConditions Conditions;  
  
}
```

Stejně jako u operace Update je i mazání záznamu proveditelné dvěma způsoby:

1. Vytvoření instance třídy `UniDelete`, předáním názvu tabulky, ze které budeme záznamy mazat, v konstruktoru a nastavením požadovaných podmínek.
2. Vytvoření instance třídy `UniDelete` a předáním databázového objektu v konstruktoru. Opět budou načteny informace o objektu z metadat třídy a bude smazán příslušný záznam na základě porovnání primárního klíče.

Smazání záznamů bude demonstrováno na instanci třídy `UniDelete`, kterou nastavíme tak, aby smazala všechny veřejné zakázky vytvořené uživatelem Josefem Novákem:

```
UniDelete lDelete = new UniDelete("VZ");  
lDelete.FromTableName = "VerejneZakazky VZ";  
lDelete.JoinTables.Add("Uzivatele U")  
.On("U.ID=VZ.IDUzivateleZadal");  
lDelete.Conditions.AddCondEqual("U.Jmeno", "Josef");  
lDelete.Conditions.AddCondEqual("U.Prijmeni", "Novák");  
lDelete.Execute();
```

Dotaz sestavený pro DB engine Oracle bude vypadat:

```
DELETE FROM (  
    SELECT VZ.*  
    FROM VerejneZakazky VZ  
    WHERE VZ.ID IN (  
        ...  
    )  
)
```

```
SELECT VZ.ID
FROM VerejneZakazky VZ
INNER JOIN Uzivatele U ON U.ID=VZ.IDUzivateleZadal
WHERE U.Jmeno='Josef' AND U.Prijmeni='Novák'
))
```

Dotaz sestavený pro DB engine SQL Server bude vypadat:

```
DELETE VZ
FROM VerejneZakazky VZ
INNER JOIN Uzivatele U ON U.ID= VZ.IDUzivateleZadal
WHERE U.Jmeno='Josef' AND U.Prijmeni='Novák'
```

3.4.5 Implementace UniFunction

```
public class UniFunction {
    public UniFunction(string FunctionName) {
        ...
    }
    public string FunctionName;
    public string Schema;
    public colParameters Parameters;
}
```

Volání funkce je proveditelné dvěma způsoby v závislosti na tom, jaký typ výsledku očekáváme. Jedním typem návratové hodnoty je tabulka, druhým typem je skalární veličina, v následujícím příkladu ukážeme volání funkce, jejíž návratovou hodnotu je číslo, která přijímá jediný parametr typu datum:

```
UniFunction lFunction = new UniFunction ("PocetAktVZvMesici");
lFunction.Attributes.Add(gActualDate);
int lResult = lFunction.ExecuteRetScalar();
```

Dotaz sestavený pro DB engine Oracle bude vypadat:

```
SELECT PocetAktVZvMesici(SYSDATE) FROM DUAL;
```

Dotaz sestavený pro DB engine SQL Server bude vypadat:

```
SELECT PocetAktVZvMesici(GETDATE())
```

Funkce, která vrací aktuální datum a čas, bude v programu uložena v globálním parametru `gActualDate`, který bude nastavován jedinkrát a to při instalaci aplikace dle zvoleného DB engine.

3.4.6 Implementace UniProcedure

```
public class UniProcedure {  
    public UniProcedure(string ProcedureName) {  
        ...  
    }  
    public string ProcedureName;  
    public colParameters Parameters;  
}
```

Volání procedury ukážeme na volání následující procedury, jejímž jediným parametrem je proměnná typu datum (v tomto případě předáme aktuální serverový datum a čas):

```
UniProcedure lProcedure = new UniProcedure("PrepocetVZDleDPH");  
lProcedure.Attributes.Add(gActualDate);  
lInsert.Execute();
```

Dotaz sestavený pro DB engine Oracle bude vypadat:

```
EXEC PrepocetVZDleDPH(SYSDATE);
```

Dotaz sestavený pro DB engine SQL Server bude vypadat:

```
EXEC PrepocetVZDleDPH GETDATE()
```

3.4.7 Testování provedených změn

Testování implementovaných změn probíhalo na straně společnosti NWT a.s. jejími zaměstnanci. Ti následně zasílali reporty s výsledky testů, na základě kterých byly prováděny další úpravy, bylo-li třeba.

4 VYHODNOCENÍ PROVEDENÝCH ÚPRAV

Očekávaným přínosem tohoto projektu je především zvýšení flexibility společnosti NWT a.s. v reakci na požadavky zákazníků a zvýšení konkurenceschopnosti na trhu.

Podle SWOT analýzy elektronizace veřejných zakázek provedené v [14] je jedním ze slabých míst nedostatečná infrastruktura pro elektronické zadávání VZ a nepřipravenost informačních systémů. Aplikace PAVEZA se přechodem na nový engine snaží reagovat na požadavky zákazníků, kteří se vždy nemusí spokojit s řešením fungujícím pouze nad databází Oracle.

Prvním zákazníkem využívajícím aplikaci PAVEZA byl státní podnik Lesy České republiky, který měl zakoupené licence na Oracle DB a pro nějž se aplikace přizpůsobila. Licence Oracle DB jsou ale v porovnání s licencemi ostatních DB engineů finančně náročnější a nedostatek finančních prostředků může být důvodem pro potenciálního zákazníka aplikaci PAVEZA nevyužívat.

Navržené řešení momentálně počítá s využitím DB engineů Oracle a SQL Server, struktura řešení ale počítá do budoucna s potenciálním rozšířením na další DB enginey, které by byly zákazníky požadovány. Časová náročnost implementace dalšího rozšíření nad aktuálně implementovanou univerzální datovou vrstvou je odhadována asi na 6 týdnů práce, což umožňuje společnosti NWT a.s. poměrně flexibilně reagovat na požadavky zákazníků.

S

- El. zadávání VZ a „klasickým“ způsobem založeným na listinných dokumentech jsou legislativně postaveny naroveň.
- Funguje centrálně provozovaný IS pro uveřejňování informací o VZ.
- Usnesením vlády byla schválena Strategie používání e-tržišť. Toto usnesení upravuje regulaci nového systému elektronických tržišť.

W

- Nedostatečná infrastruktura pro el. zadávání VZ, nedostatky v připravenosti inf. systémů na komplexnější informační podporu.
- Nedůvěra a neochota pracovníků zadavatele využívat el. nástroje.
- Neochota dodavatelů účastnit se el. zadávacích řízení.
- Nízká počítačová gramotnost pracovníků zadavatele a dodavatele a nedostatek školení na téma el. zadávání VZ.
- Nízká metodická podpora ze strany gestora zákona.
- Nestabilní legislativní prostředí.

- Pokračování projektu NIPEZ.
- Využití dosavadních výstupů standardizační a metodické účinnosti EU a zapojení do aktuálních projektů a iniciativ EU.
- Využití výstupů ostatních projektů v rámci Smart Administration.

O

- Zadavatelé budou stávající elektronické nástroje používat neefektivně (např. úkony prováděné elektronicky budou provádět i „klasickým“ způsobem), čímž se zvýší administrativní náklady celého zadávacího procesu, či způsobem, který sníží transparentnost (např. výzvy k podání nabídek uzavřenému počtu dodavatelů prostřednictvím stávajících e-tržišť pro veřejnou správu).
- Některé oblasti nezbytné pro efektivní fungování el. zadávání VZ (např. problematika dlouhodobého ukládání a archivace el. dokumentů) se nepodaří uspokojivě dořešit a zadavatelé tak nebudou mít k el. zadávání důvěru.
- Zadavatelé nebudou ze své vůle implementovat el. nástroje (např. strach z úniku informací)

T

ZÁVĚR

V rámci této práce byla analyzována aplikace pro plánování a evidenci veřejných zakázek PAVEZA, která pochází z dílny společnosti NWT a.s. Cílem bylo navrhnout způsob přechodu z aktuálně používaného databázového engine Oracle na zvolený databázový engine, kterým byl SQL Server a toto řešení implementovat a zhodnotit.

V první části práce byla provedena analýza aktuálního stavu aplikace PAVEZA, dále veřejných zakázek v České republice a procesu jejich elektronizace včetně požadavků na elektronické nástroje vyhovujícím zákonu o veřejných zakázkách. Pozornost byla věnována databázovému engine, nad kterým aplikace PAVEZA aktuálně pracuje a nově zvolenému databázovému engine a rozdílům mezi oběma enginey.

Druhá část práce popisuje návrh univerzální datové vrstvy, která by měla přinést požadovanou funkcionalitu a popisuje princip její implementace. V závěru druhé části jsou zhodnoceny implementované změny.

Aplikace PAVEZA byla naprogramovaná pouze pro databázový engine Oracle, což bylo výsledkem požadavků prvního a na delší dobu jediného zákazníka využívajícího této aplikace. Cena licencí a nutný zásah do stávající infrastruktury v případě používání jiného databázového engine potenciálních zákazníků jsou však zásadní problémy, které potenciální zákazníky od používání aplikace PAVEZA mohou odradit.

V rámci této práce byla navržena a implementována univerzální datová vrstva, která umožňuje jak využívání aplikace nad Oracle databází, tak nad databázovým engine SQL Server. Při implementaci úprav bylo počítáno s možností rozšíření aplikace PAVEZA na další databázové enginey, čímž se rozrůstá okruh potenciálních zákazníků a PAVEZA se stává na trhu konkurenceschopnější.

SEZNAM POUŽITÉ LITERATURY

- [1] NWT, a.s. *O společnosti* [online]. 2012 [cit. 2014-03-02]. Dostupné z: <http://www.nwt.cz/>
- [2] NWT, a.s. *PAVEZA & EVEZA: Produktové portfolio elektronických nástrojů pro správu veřejných zakázek*. 2012.
- [3] NWT, a.s. *Historie NWT* [online]. [cit. 2014-03-05]. Dostupné z: <http://www.nwt.cz/img/static/file/historie%20NWT.pdf>
- [4] NWT, a.s. *Výroční zpráva NWT a.s. 2011/2012* [online]. [cit. 2014-03-05]. Dostupné z: http://www.nwt.cz/img/static/file/vyrocnizprava_nwt_2011.pdf
- [5] Lesy České republiky, s.p. *Výroční zpráva 2012* [online]. [cit. 2014-03-10]. Dostupné z: http://www.lesy.cz/o-nas/dokumenty-ke-stazeni/Documents/Lesy_Ceske_republiky_Vyrocnizprava_2012.pdf
- [6] Vysoké učení technické v Brně. *Prezentace VUT 2013* [online]. [cit. 2014-03-10]. Dostupné z: <https://www.vutbr.cz/fakulty-a-soucasti/rektorat/marketingove-oddeleni/multimedialni-prezentace-vut/prezentace-vut-2013-p75451>
- [7] Statutární město Olomouc. *O městě* [online]. [cit. 2014-03-10]. Dostupné z: <http://www.olomouc.eu/o-meste>
- [8] Předpis 667/1920 Sb., nařízení vlády Československé republiky o zadávání státních dodávek a prací. (Zadávací řád). In: *Sbírka zákonů*. 31.12.1920.
- [9] Zákon č. 40/2004 Sb., o veřejných zakázkách. In: *Sbírka zákonů*. 5.2.2004.
- [10] Zákon č. 137/2006 Sb., o veřejných zakázkách. In: *Sbírka zákonů*. 14.3.2006.
- [11] Transparency International – Česká republika. *Veřejné zakázky v České republice: Korupce nebo transparentnost?* [online]. 2005. [cit. 2014-03-18]. Dostupné z: http://www.transparency.cz/doc/vz_studie_text.pdf
- [12] Ministerstvo pro místní rozvoj ČR. *Portál o veřejných zakázkách a koncesích: Legislativa* [online]. [cit. 2014-03-18]. Dostupné z: <http://www.portal-vz.cz/cs/Jak-na-zadavani-verejnych-zakazek/Legislativa-a-Judikatura/Legislativa>
- [13] Zákon č. 127/2005 Sb., o elektronických komunikacích. In: *Sbírka zákonů*. 22.2.2005.

- [14] Ministerstvo pro místní rozvoj ČR. *Portál o veřejných zakázkách a koncesích: Strategie elektronizace zadávání veřejných zakázek pro období let 2011 až 2015* [online]. [cit. 2014-03-20]. Dostupné z: <http://www.portal-vz.cz/cs/Jak-na-zadavani-verejnych-zakazek/Elektronicke-zadavani-verejnych-zakazek/Strategie-elektronizace-zadavani-VZ-2011-2015>
- [15] Ministerstvo pro místní rozvoj ČR. *Portál o veřejných zakázkách a koncesích: Národní plán elektronizace VZ 2006-2010* [online]. [cit. 2014-03-20]. Dostupné z: <http://www.portal-vz.cz/cs/Jak-na-zadavani-verejnych-zakazek/Elektronicke-zadavani-verejnych-zakazek/Narodni-plan-elektronizace-VZ-2006-2010>
- [16] PRICE, Jason. *Oracle database 11g SQL*. New York: Oracle Press/McGraw-Hill, c2008, xxviii, 656 p. ISBN 00-714-9850-8.
- [17] KUHN, Darl. *Pro Oracle database 12c administration*. 2nd ed. New York, NY: Apress, 2013, xxviii, 656 p. ISBN 978-143-0257-288.
- [18] JORGENSEN, Adam. *Professional microsoft sql server 2012 administration*. 1st ed. Indianapolis, IN: Wiley Pub., Inc., 2012, p. cm. ISBN 11-181-0688-1.
- [19] ROSS MISTRY, Stacia Misner. *Introducing Microsoft SQL Server 2012*. 1st ed. Redmond, WA: Microsoft Press, 2012, p. cm. ISBN 07-356-6515-X.
- [20] ORACLE. *Oracle Migration Workbench: Reference Guide for SQL Server and Sybase Adaptive Server Migrations*. 2002.
- [21] Microsoft. *.NET Framework Conceptual Overview* [online]. [cit. 2014-04-22]. Dostupné z: [http://msdn.microsoft.com/en-us/library/zw4w595w\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.100).aspx)
- [22] Microsoft. *Microsoft Visual Studio* [online]. [cit. 2014-04-22]. Dostupné z: <https://www.microsoft.com/cze/msdn/vstudio/>
- [23] Microsoft. *Introducing SQL Server Management Studio* [online]. [cit. 2014-04-22]. Dostupné z: [http://technet.microsoft.com/cs-cz/library/ms174173\(v=sql.100\).aspx](http://technet.microsoft.com/cs-cz/library/ms174173(v=sql.100).aspx)

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ANSI	American National Standards Institute.
CPV	Concentrated PhotoVoltaic.
CTE	Common Table Expression.
DB	Databáze.
DML	Data Manipulation Language.
ISO	International Organization for Standardization.
NIPEZ	Národní infrastruktura pro elektronické zadávání veřejných zakázek.
NWT	New World Technologies.
OLAP	On-line Analytical Processing.
OLTP	On-line Transaction Processing.
PL/SQL	Procedural Language/ Structured Query Language.
RTM	Release to Manufacturing.
SSMS	SQL Server Management Studio.
SP	Service Pack.
T-SQL	Transact Structured Query Language.
VZ	Veřejná zakázka.
VZMR	Veřejná zakázka malého rozsahu.
WPF	Windows Presentation Foundation.

SEZNAM OBRÁZKŮ

Obr. 1 – Logo společnosti NWT a.s.	12
Obr. 2 - Organizační struktura společnosti NWT a.s.	13
Obr. 3 - Obrat společnosti NWT a.s. v letech 2007 až 2011	13
Obr. 4 - PAVEZA: Detail veřejné zakázky	15
Obr. 5 - PAVEZA: Plánování VZ	16
Obr. 6 - Logo státního podniku Lesy České republiky	20
Obr. 7 - Obecný pohled na .NET Framework	40
Obr. 8 - Vývojové prostředí Microsoft Visual Studio 2010	41
Obr. 9 - SQL Server Management Studio UI	42
Obr. 10 - Současné zobrazení 3 vrstvé architektury aplikace PAVEZA	43
Obr. 11 - Architektura aplikace PAVEZA s využitím univerzální datové vrstvy	44

SEZNAM TABULEK

Tab. 1 - PAVEZA: Technologie	17
Tab. 2 - Modul plánování	18
Tab. 3 - Modul veřejných zakázek	19
Tab. 4 - Finanční limity VZ platné k 1.1.2014	23
Tab. 5 - Přehled verzí Oracle databáze od roku 1985.....	29
Tab. 6 - Přehled verzí SQL Serveru od roku 1995	31