

Propojení systému SAP a aplikace O2 Car Control

Radim Zlámal

Bakalářská práce
2015



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Radim Zlámal**

Osobní číslo: **A12604**

Studijní program: **B3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Forma studia: **kombinovaná**

Téma práce: **Propojení systému SAP a aplikace O2 Car Control**

Téma anglicky: **The Interlinkage of a SAP System and an O2 Car Control Application**

Zásady pro vypracování:

1. Popište základní vlastnosti aplikací O2 Car Control a SAP.
2. Analyzujte potřeby společnosti pro vyhodnocení ekonomiky provozu automobilů.
3. Popište technologickou platformu SAP a možnost vývoje v systému.
4. Navrhněte architekturu uživatelského modulu pracující v systému SAP.
5. Realizujte v uživatelském modulu import dat ze systému O2 Car Control.
6. Vytvořte reporty nad společnými daty z aplikací SAP a O2 Car Control.
7. Zhodnoťte přínos aplikace pro vyhodnocení ekonomiky provozu aut.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **MAASSEN, A., SCHOENEN, M., GADATSCH, A. SAP R/3: kompletní průvodce. Computer Press, 2007. ISBN 978-80-251-1750-7.**
2. **KUHNHAUSER, K. ABAP: výukový kurz. Computer Press, 2009. ISBN 978-80-251-2117-7.**
3. **GULSEN, A. ABAP Workbench -100 things you should know about. Galileo Press, 2012. ISBN 978-1-59229-427-5.**
4. **ERL, T. SOA, servisně orientovaná architektura: kompletní průvodce. Computer Press, 2009. ISBN 978-80-251-1886-3.**
5. **SAP Help Portal [online]. Dostupné z: <http://help.sap.com>.**

Vedoucí bakalářské práce:

doc. Ing. Zdenka Prokopová, CSc.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

6. března 2015

Termín odevzdání bakalářské práce:

22. května 2015

Ve Zlíně dne 6. března 2015

doc. Mgr. Milan Adámek, Ph.D.
děkan



L.S.

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

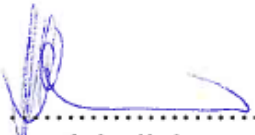
Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské 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;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou 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;
- beru na vědomí, že podle § 60 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 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen přípouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské 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/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské 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.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně



.....
podpis diplomanta

ABSTRAKT

Bakalářská práce se zabývá tvorbou transakce ZCAR, která slouží pro vyhodnocování dat o provozu vozidel z aplikace CarControl a účetních dat ze systému SAP R/3. Transakce byla vytvořena z důvodu sjednocení dvou typů reportingu do jednoho výstupu za účelem rychlejšího a efektivnějšího vyhodnocování nákladů na dopravní vozidla společnosti.

Klíčová slova: SAP, ABAP, ZCAR, CarControl

ABSTRACT

Bachelor's thesis is dealing with the creation of ZCAR transaction. It will be used for data evaluation of car operation from application CarControl and accounting data in SAP R/3 system. Transaction was made due to unification of two reporting types to one outcome with quicklier and more effective cost evaluation of firm's cars.

Keywords: SAP, ABAP, ZCAR, CarControl

Nejprve bych chtěl poděkovat doc. Ing. Zdeňce Prokopové, CSc. za vedení mé práce, za připomínky a nasměrování.

Dále bych chtěl poděkovat své ženě a rodině za podporu během studia. Současně chci poděkovat firmě Navos, a.s. za umožnění studia a hlavně ing. Pavlu Trefilovi za podporu.

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

OBSAH

I TEORETICKÁ ČÁST.....	10
1 ZÁKLADNÍ VLASTNOSTI SAP A CARCONTROL	11
1.1 SAP R/3	11
1.1.1 Historie SAP R/3	11
1.1.2 Moduly SAP R/3	11
1.1.3 Architektura systému SAP R/3	13
1.2 CARCONTROL	14
2 ANALÝZA STAVU A POTŘEB SPOLEČNOSTI NAVOS, A.S.....	17
2.1 VYHODNOCENÍ EFEKTIVNOSTI PROVOZU AUTOMOBILŮ	17
2.2 MOŽNOSTI VÝVOJE SYSTÉMŮ	19
2.2.1 Transakce	19
2.2.2 ABAP Workbench	20
2.2.3 Databázové tabulky systému SAP	21
2.2.4 Interní tabulky	22
2.2.5 Události reportu.....	23
2.2.6 Oprávnění	25
2.2.7 Výměna zpráv s protokolem SOAP	26
II PRAKTICKÁ ČÁST	27
3 NÁVRH ARCHITEKTURY UŽIVATELSKÉHO MODULU	28
3.1 DATOVÁ STRUKTURA APLIKACE ZCAR	30
3.2 VAZBY MEZI TABULKAMI	32
4 REALIZACE UŽIVATELSKÉHO MODULU	35
4.1 USPOŘÁDÁNÍ VSTUPNÍCH SOUBORŮ	35
4.2 UDÁLOSTI REPORTU A DEFINICE POLÍ VÝBĚROVÝCH OBRAZOVEK	37
4.3 ADMINISTRACE	38
4.3.1 Ovládání administrace.....	38
4.3.2 Propojení na SAP přes controllingovou zakázku.....	40
4.3.3 Mapování účtů.....	42
4.3.4 Import dat	44
4.3.4.1 Import přes XML soubor	45
4.3.4.2 Import přes webovou službu.....	46
4.3.4.3 Zpracování importních dat.....	47
5 REPORTY	49
5.1.1 Report On-line náhled přes webovou službu	50
5.1.2 Report Statistika dle vozidel	51
5.1.3 Report Náklady	52
5.1.3.1 Výběr dat z tabulek SAPu.....	53
5.1.3.2 Vytvoření struktury a načtení dat.....	53
5.1.3.3 Dotažení dat z controllingových zakázek ze SAPu	55
ZÁVĚR	56
SEZNAM POUŽITÉ LITERATURY.....	58
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	59
SEZNAM OBRÁZKŮ	60

SEZNAM TABULEK.....	61
SEZNAM PŘÍLOH.....	62

Úvod

Tématem bakalářské práce je naprogramování uživatelské transakce v systému SAP R/3, jejíž cílem je propojení dat ze systému O2 CarControl a systému SAP R/3. Projekt vychází z požadavků a potřeb firmy Navos, a.s., kde je potřeba konsolidovat a centralizovat informace o provozu služebních referenčních vozidel.

Společnost Navos, a.s. má v současnosti 149 osobních referenčních vozidel. Provoz těchto vozidel je monitorován systémem CarControl společnosti O2. Na trhu je vícero výrobců a provozovatelů podobných systémů, jelikož je ale společnost Navos, a.s. členem holdingu Agrofert, je řešení vybráno centrálně pro celý holding.

Každé vozidlo je opatřeno komunikační jednotkou. Tato jednotka komunikuje přes GPS modul s družicí, kdy získává informace o aktuální poloze, rychlosti, apod. a takto získaná data posílá přes mobilní službu na centrální servery společnosti O2. Zde dochází k dalšímu zpracování, tj. evidenčním operacím typu jaké vozidlo, v jakém čase, na jakém místě, jaká rychlost, atd. K těmto datům pak přes webovou aplikaci uživatel definuje další údaje, jako je typ cesty, spotřeba pohonných hmot, informace o servisních úkonech. Vedle toho jsou k dispozici ekonomické informace o jednotlivých vozidlech v systému SAP R/3, kdy se přes controllingové zakázky evidují náklady na pohonné hmoty, spotřeba olejů včetně dalších provozních kapalin, atd. Problémem je, že tyto dvě evidence nejsou propojeny. V SAPu nejsou informace o technickém provozu vozidel a není tedy možné současně reportovat a vyhodnocovat ekonomické i technické ukazatele o provozu vozidel.

Cílem bakalářské práce je tedy vytvořit transakci v systému SAP R/3, která bude ze systému CarControl načítat informace o provozu referenčních vozidel, a bude je pro další využití importovat v reportech do systému SAP.

I. TEORETICKÁ ČÁST

1 ZÁKLADNÍ VLASTNOSTI SAP A CARCONTROL

1.1 SAP R/3

1.1.1 Historie SAP R/3

Firma SAP Byla založena v roce 1972 s cílem vytvořit plně funkční systém pro řízení podnikové ekonomiky. První verze systému, která přišla na svět v roce 1973, SAP R/1, byla tvořena finančním účetnictvím. Další verze SAP R/2 již můžeme nazývat za první funkční ERP systém (Enterprise resources planning), ovšem nevýhodou tohoto systému byla nutnost využívání sálových počítačů. Verzi SAP R/3 z roku 1992 byla změněna architektura SAPu, kdy se zaměnily sálové počítače na architekturu klient-server a začaly se využívat relační databáze. Výhoda této architektury byla především v kompatibilitě s různými platformami a operačními systémy Microsoft Windows nebo Unix. Další verze byla v roce 2002 spuštěna pod názvem SAP R/3 Enterprise. V roce 2004 byly nově uspořádány komponenty, čímž vznikl centrální produkt mySAP Business Suite. Došlo k oddělení aplikačních komponent od technických, přičemž se nadále označují jako SAP NetWeaver. [1]

1.1.2 Moduly SAP R/3

Systém SAP je díky své architektuře a robustnosti používán v mnoha nadnárodních společnostech. Pokrývá procesy, které jsou klíčové pro činnost společností, ale nabízí i komponenty, které přesahují potřeby a požadavky ERP systému.

Celá skupina produktů SAP je horizontálně rozdělena do tří vrstev.

- SAP NetWeaver: nejnižší úroveň technologických komponent. SAP NetWeaver je charakteristický dělením na další čtyři vrstvy, a to:
 - Integrace osob
 - Integrace informací
 - Integrace procesů
 - Aplikační platforma
- Komponenty k řízení podnikové ekonomiky – nacházejí se ve střední vrstvě, která je ještě rozdělena horizontálně do tří oblastí, a to:
 - xApps (Extended Applications) – přesah přes rámce jednotlivých komponent, ty jsou propojeny přes xApps.

- mySAP Business Suite – je nutný pro nasazení xApps, obsahuje v sobě komponenty pro řízení podnikové ekonomiky, a to:
 - mySAP ERP: funkce ERP systému
 - mySAP CRM: vztah se zákazníky
 - mySAP SCM: rozšířená funkcionalita logistiky
 - mySAP SRM: řízení vztahů s dodavateli
 - mySAP PLM: životní cyklus výrobků
- SAP Smart Business Solutions – řešení určená pro střední a menší podniky, SAP ALL-in-one, SAP Business One
- Průmyslová řešení – nacházejí se ve třetí úrovni. Jedná se o dodatková řešení využívaná pouze v některých odvětvích průmyslu.[1]

Systém SAP R/3 je vnitřně rozdělen do modulů. Každý z těchto modulů řeší danou část činnosti firmy:

- **FI (Financial Accounting)** Finanční účetnictví
- **CO (Controlling)** Controlling
- **AM (Asset Management)** Evidence majetku
- **PS (Project system)** Plánování dlouhodobých projektů
- **WF (Workflow)** Řízení oběhu dokumentů
- **IS (Industry Solutions)** Specifická řešení různých odvětví
- **HR (Human Resources)** Řízení lidských zdrojů
- **PM (Plant Maintenance)** Údržba
- **MM (Materials Management)** Skladové hospodářství a logistika
- **QM (Quality Management)** Management kvality
- **PP (Production Planning)** Plánování výroby
- **SD (Sales and Distribution)** Podpora prodeje [1]

1.1.3 Architektura systému SAP R/3

Příchodem verze SAP R/3 se změnila architektura SAPu na architekturu klient-server, která je tvořena třemi vrstvami:

- Prezenční vrstva
- Aplikační vrstva
- Databázová vrstva

Prezenční vrstva slouží pro komunikaci mezi uživatelem a počítačem. Vlastní komunikace probíhá na klientské části – prezentačním serveru. Nedílnou součástí prezentačního serveru je SAP GUI rozhraní, které se stará o komunikaci mezi prezentačním a aplikačním serverem. Aplikační vrstva je tvořena aplikačním serverem, který jednak přes SAP GUI komunikuje s klientem a jednak komunikuje s databází přes systém pro správu databáze. Vlastní programy, vytvořené v systému ABAP, jsou uloženy na aplikačních serverech. Databázová vrstva je tvořena vlastními databázovými servery, které slouží pro ukládání dat. Jelikož SAP je multiplatformní systém, vývojáře nemusí zajímat, na jaké databázové platformě (UNIX, ORACLE, SUN, MICROSOFT) databázová vrstva běží, aplikační vrstva bude vypadat vždy stejně.

Z hlediska řízení organizace, ve které se SAP R/3 nasazen, můžeme ještě architekturu systému popsat následovně:

- Systém pro testování a vývoj
- Konsolidační systém
- Produkční systém [2]

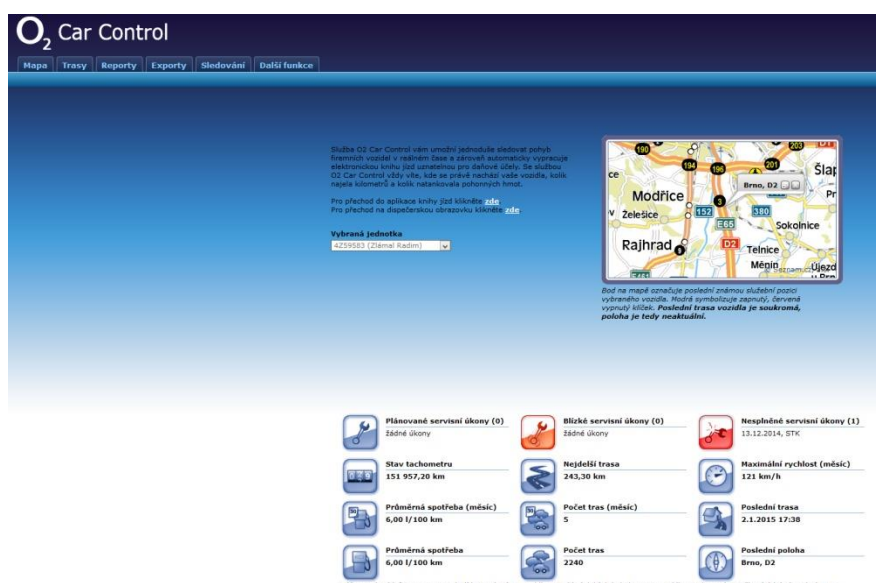
Systém chápeme jako uzavřený konsolidovaný celek, na kterém běží samostatná instalace systému SAP R/3. Počty systémů záleží na druhu organizace, kde je systém SAP nasazen. Na každém systému může běžet několik klientů. Klient je z hlediska hierarchie nejvyšší úroveň v systému. Data, která využívány napříč jednotlivými moduly, jsou uloženy v klientech. Jsou-li data klientově závislá (databázová tabulka obsahuje pole MANDT), jsou použitelná pouze na úrovni klienta. Jsou ale i data klientově nezávislá, která jsou viditelná v celém systému napříč klienty.

1.2 CarControl

Bakalářská práce nemá za cíl studovat technickou problematiku nastavení systému CarControl. Technické standardy jsou brány jako fakt, pro další zpracování jsou důležitá data, která ze systému CarControl lze získat. Proto i popis funkčnosti vlastní aplikace je omezen jen na části, které budou potřeba pro další zpracování.

Jak již bylo zmíněno v úvodu, každé vozidlo je opatřeno komunikační jednotkou, která jednak komunikuje s globálním družicovým polohovým systémem GPS a také přes mobilní služby s datovým centrem O2, kam získané informace předává. Dle nastavení uživatelských práv se pak k těmto datům přes webový portál a zabezpečenou SSL komunikaci přihlašují jednotliví uživatelé.

Po přihlášení na adrese <https://carcontrol.cz.o2.com/web/> a zadání přihlašovacího jména a hesla se uživatel dostane na úvodní stranu portálu, kde v mapě uvidí základní informace o vozidlech, na která má práva.



Obr.1. Úvodní obrazovka webového portálu CarControl

Aplikace má několik záložek. Zobrazování záložek opět záleží na nastavení uživatelských oprávnění.

Důležitou záložkou je záložka Trasy. Zde je vedena evidence o provozu vozidla, tj. ke každé trase vozidla jsou vidět následující informace:

- Typ cesty – soukromá nebo služební

- Čas – den, datum a čas odjezdu a příjezdu
- Místo – místo odjezdu a příjezdu
- Řidič – řidič vozidla na dané trase
- Doba trasy – čas strávený na cestě
- Celkem km – počet kilometrů ujetý na trase
- Tachometr – konečný stav tachometru na trase

Typ	Čas	Místo	Řidič	Účel	Doba trasy	Celkem km	Tachometr
	pá 13.03.2015 11:32:3	[CZ] Kroměříž, Na Nohýláku	Velčovský Pav	Služební jízda	00:12:44	3,60	146720,60
	pá 13.03.2015 11:45:2	[CZ] Kroměříž, Čelakovského	Velčovský Pav	Služební jízda	00:04:16	1,80	146717,00
	pá 13.03.2015 11:21:1	[CZ] Kroměříž, Páleníčkova#Slá	Velčovský Pav	Služební jízda	00:03:56	1,60	146715,20
	pá 13.03.2015 11:25:3	[CZ] Kroměříž, Na Nohýláku	Velčovský Pav	Služební jízda	00:05:25	1,50	146713,60
	pá 13.03.2015 11:06:4	[CZ] Kroměříž, Havlíčkova#Souc	Velčovský Pav	Služební jízda	00:17:01	5,20	146712,10
	pá 13.03.2015 11:10:4	[CZ] Kroměříž, Páleníčkova#Slá	Velčovský Pav	Služební jízda	00:04:19	2,50	146706,90
	pá 13.03.2015 10:46:4	[CZ] Kroměříž, Pilařova#Sněmo	Velčovský Pav	Služební jízda	00:04:39	2,20	146704,40
	pá 13.03.2015 10:52:1	[CZ] Kroměříž, Havlíčkova#Souc	Velčovský Pav	Služební jízda	00:06:48	2,80	146702,20
	pá 13.03.2015 10:27:2	[CZ] Kroměříž, Čelakovského#R	Velčovský Pav	Služební jízda	00:05:52	0,70	146699,40
	pá 13.03.2015 10:44:2	[CZ] Kroměříž, Pilařova#Sněmo	Velčovský Pav	Služební jízda	00:06:42	2,90	146698,70
	pá 13.03.2015 09:59:1	[CZ] Kroměříž, Ovocná#Skopal	Velčovský Pav	Služební jízda	00:01:35	0,30	146695,80
	pá 13.03.2015 09:38:4	[CZ] Kroměříž, Čelakovského	Velčovský Pav	Služební jízda	00:14:53	3,60	146695,50
	pá 13.03.2015 10:03:3	[CZ] Kroměříž, Čelakovského#R	Velčovský Pav	Služební jízda			
	pá 13.03.2015 09:43:1	[CZ] Kroměříž, Ovocná#Skopal	Velčovský Pav	Služební jízda			
	čt 12.03.2015 13:11:01	[CZ] Kroměříž, Prusinovského#	Velčovský Pav	Služební jízda			
	čt 12.03.2015 12:58:21	[CZ] Kroměříž, Čelakovského	Velčovský Pav	Služební jízda			
	čt 12.03.2015 13:17:46	[CZ] Kroměříž, Čelakovského	Velčovský Pav	Služební jízda			
	čt 12.03.2015 12:46:41	[CZ] Kroměříž, Prusinovského#	Velčovský Pav	Služební jízda			
	čt 12.03.2015 13:04:11	[CZ] Kroměříž, Čelakovského#R	Velčovský Pav	Služební jízda			
	čt 12.03.2015 12:46:41	[CZ] Kroměříž, Prusinovského#	Velčovský Pav	Služební jízda			
	čt 12.03.2015 10:53:21	[CZ] Kroměříž, Kpt. Jaroše#Vele	Velčovský Pav	Služební jízda			
	čt 12.03.2015 10:15:46	[CZ] Kroměříž, Čelakovského	Velčovský Pav	Služební jízda			
	čt 12.03.2015 10:17:21	[CZ] Kroměříž, Čelakovského#R	Velčovský Pav	Služební jízda			
	čt 12.03.2015 09:58:11	[CZ] Kroměříž, Kpt. Jaroše	Velčovský Pav	Služební jízda			

Obr. 2. Kniha jízd ve webovém portálu CarControl

Další záložkou jsou Výdaje. Zde jsou definovány informace o nákladech na provoz vozidel, jako je tankování včetně ostatních provozních nákladů.

Uživatel s oprávněním Admin má přístup do centrálního nastavení aplikace. Zde jsou důležité záložky Vozidla a Uživatelé. Tyto informace zpracovává správce systému.

- Vozidla:
 - Zde je uveden číselník vozidel
 - Každý záznam obsahuje Typ vozidla, SPZ, řidiče, číslo jednotky, číslo SIM karty v jednotce pro komunikace přes GPRS, aktuální stav tachometru

- Uživatelé:
 - Zde je uveden číselník uživatelů – řidičů
 - Každý záznam obsahuje jméno a příjmení řidiče, email

Data uložená v databázových tabulkách CarControlu, obsahují další sloupce, které pouze nejsou v portálu zobrazeny.

Pro provázání dat v reportu bude potřeba exportovat následující informace:

- Vozidla: seznam vozidel v systému
- Uživatelé: seznam uživatelů v systému
- Trasy: vlastní seznam jízd vozidel
- Náklady: náklady na PHM, jiné náklady

Trasy a náklady lze do SAPu importovat přes webovou službu, import vozidel a uživatelů systém CarControl přes webovou službu neumožňuje, zde musí být použit import přes XML soubory.

2 ANALÝZA STAVU A POTŘEB SPOLEČNOSTI NAVOS, A.S.

2.1 Vyhodnocení efektivnosti provozu automobilů

Každé vozidlo v systému SAP má definovanou controllingovou zakázku. Zakázek může být více druhů, např. jsou zakázky termínové pro evidenci prodeje a nákupu zboží, zakázky servisní používané pro práci v CS modulu pro servis strojů. V případě CarControl se jedná o nákladové statistické zakázky. Když je potřeba odepsat příslušný náklad, odepisuje se s vazbou na danou statistickou controllingovou zakázku. Výsledkem tedy je, že na dané statistické zakázce je evidence provozních nákladů, které byly na dané vozidlo zúčtovány. Jedná se o:

- Náklady na PHM: v případě, že se jedná o tankování v areálu společnosti na vlastním stojanu, dochází k odpisu zásoby PHM do vlastní spotřeby, tedy v transakci MIGO účetní odepisuje dané množství PHM ze skladu a v dokladu uvede číslo statistické zakázky příslušného vozidla. Jedná-li se o tankování na veřejných benzínových stanicích, zúčtuje se dodavatelská faktura a náklad na PHM za dané vozidlo se zúčtuje na číslo statistické zakázky příslušného vozidla.
- Opravy, další poplatky, dálniční známky: zúčtuje se dodavatelská faktura a náklad je zúčtován na číslo statistické zakázky příslušného vozidla.
- Odpisy: zúčtovávají se interním zpracováním na číslo statistické zakázky příslušného vozidla.

Pro vyhodnocení se používá transakce KOB1, kde jsou vidět finanční náklady na zvolené statistické zakázky, tedy na odpovídající automobily.

Zakázka	Označení objektu	Číslo dokladu	Dat.účt.	Období	Rok	Nákl.druh	Popis nákl.druhu	Označení průběhu
97104Z59583	Octavia Zlámal	72166975	18.03.2013	3	20...	501000102	Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72222762	31.03.2013	3	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72290713	17.04.2013	4	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72367409	30.04.2013	4	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72367582	30.04.2013	4	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72457121	17.05.2013	5	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72542245	31.05.2013	5	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72613689	18.06.2013	6	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72668668	30.06.2013	6	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72744603	16.07.2013	7	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72824173	31.07.2013	7	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72899196	20.08.2013	8	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72971561	30.08.2013	8	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	73041773	16.09.2013	9	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	73122215	30.09.2013	9	20...		Spotřeba pohonných hmot	Zboží na skl. PHM
97104Z59583	Octavia Zlámal	72289224	16.04.2013	4	20...	5010001	Opravy a udržování dopravních prostředků	Agrotec A.S.
97104Z59583	Octavia Zlámal	72289224	16.04.2013	4	20...	511000200	Opravy a udržování dopravních prostředků	Agrotec A.S.
97104Z59583	Octavia Zlámal	72604606	14.06.2013	6	20...		Opravy a udržování dopravních prostředků	Agrotec A.S.
97104Z59583	Octavia Zlámal	72604606	14.06.2013	6	20...		Opravy a udržování dopravních prostředků	Agrotec A.S.
97104Z59583	Octavia Zlámal	72391695	30.04.2013	4	20...	5110002	Ostatní služby	AFEED, a.s.
97104Z59583	Octavia Zlámal	72514937	29.05.2013	5	20...	518300119	Ostatní služby	Pokladna CZK
97104Z59583	Octavia Zlámal	72514937	29.05.2013	5	20...		Ostatní služby	Pokladna CZK
97104Z59583	Octavia Zlámal	72809733	31.07.2013	7	20...		Ostatní služby	Pokladna CZK

Obr. 3. SAP report z transakce KOB1 – Jednotlivé položky skutečných nákladů za zakázku

Systém CarControl nabízí velké množství logistických a manažerských sestav, jako jsou reporty zobrazující počty najetých soukromých a služebních kilometrů, zobrazování tras, kde vozidla překročila maximální povolenou rychlost, zobrazuje duplicitní cesty - zda ve stejném čase na totožné místo nejedí více služebních vozidel, atd. Ale reporty pro finanční vyhodnocení provozu vozidel nenabízí potřebné informace.

Výstupní data z obou systémů jsou dále ručně zpracovávána, kdy jsou ze SAPu získávány finanční údaje, které se ovšem musí dle vozidel a měsíců sumarizovat a dle účtů vyčíslovat měsíční náklady za:

- Finanční náklady na pohonné hmoty
- Odpisy drobného nehmotného majetku a drobného investičního majetku
- Odpisy a udržování dopravních prostředků
- Ostatní služby
- Další poplatky (dálniční známky, apod.)

K těmto zpracovaným datům jsou pak dále ručně doplňovány data z CarControlu, kdy se k daným vozidlům za dané časové období ručně přiřadí informace o provozu a nákladech za kilometr. Poté je získán finální report o nákladech na provoz vozidel.

2.2 Možnosti vývoje systémů

Pro vývoj reportů v systému SAP je používán programovací jazyk ABAP. Vlastní kódy reportů nejsou uloženy fyzicky v souborech, ale přímo v databázi na databázových serverech. Spouštění programů zabezpečuje a provádí runtime systém, který dohlíží nad logickým během programů, reakci uživatelů v dialogových oknech, apod.

V rámci SAPu rozlišujeme vlastní aplikační programy:

- Reporty
- Dynamické programy

Reportem je myšlena přesná struktura programu, kdy uživatel při spuštění reportu zadá vstupní data, systém provede zpracování a nakonec vypíše výsledná data buď v textové formě nebo v grafickém formátu ALV. Dynamické programy jsou aplikace, které jsou řízeny systémem dialogových oken – dynper, a je na uživateli, které z dialogů bude používat. Při vytváření programu je potřeba definovat název vlastního programu. Vytváříme-li vlastní report, jedná se o report zákaznický. Proto název takového reportu musí vždy začínat buď „Y“ nebo „Z“. Pakliže nemáme příslušný klíč pro přístup do editace standardu SAPu, systém nám neumožní název programu, který by nezačínal jedním z výše uvedených znaků, uložit. Tím je zabezpečeno, že uživatelské reporty nebudou v případě upgrade celého systému přepsány.

2.2.1 Transakce

Každý program vytvořený v jazyce ABAP je charakterizován svým názvem. Tento název může být někdy dosti komplikovaný na zapamatování a jednoduché spuštění programu (např. program pro založení dodavatele do systému se jmenuje SAPMF02K). Proto existuje možnost každému programu definovat tzv. transakci, tedy jednoduchý kód pro spuštění (v tomto případě XK01). Transakci můžeme definovat např. přes ABAP DEVELOPMENT WORKBENCH (transakce SE80), kdy je potřeba zadat název programu a číslo původní obrazovky. Název transakce pro vytvářenou aplikaci bude ZCAR.

2.2.2 ABAP Workbench

ABAP Workbench je skupina nástrojů, pomocí kterých můžeme vytvářet a editovat vlastní programy, datové prvky, formuláře, apod. Patří sem:

- **ABAP Dictionary, tr. SE11** – knihovna domén, datových typů, view a vlastních databázových tabulek. Logika systému je taková, že jednotlivé typy, domény, atd. jsou uloženy centrálně v Dictionary a jsou tedy připraveny pro použití jak ve standardních programech SAPu, ale i v uživatelských. Je-li do reportu potřeba např. datový prvek BUKRS (účetní okruh), je vhodné jej použít z Dictionary. Je tak zajištěno, že v reportu při zobrazení v ALV, bude mít tento prvek správný typ, název (v závislosti na jazyku přihlášení), apod.
- **ABAP Editor, tr. SE38** – editor pro psaní nových a editaci již existujících programů
- **Menu painter, tr. SE41** – editor pro vytváření menu
- **Screen painter, tr. SE51** – editor pro vytváření uživatelských obrazovek a jejich logiky (obsluha událostí PROCESS BEFORE OUTPUT a PROCESS AFTER INPUT).
- **Function builder, tr. SE37** - editor pro vytváření funkčních modulů
- **Editor pro údržbu uživ. rozhraní, tr. SE41**- editor pro definici StatusBar, funkčních kláves, apod.
- **Editory pro formuláře: SapScript, tr. SE71, SapSmartforms, tr. SMARTFORMS**
- **Class Builder, tr. SE24** – editor pro definici tříd. Je-li třída uložena v Dictionary, pak je přístupná pro všechny programy v rámci systému. Pokud je třída definována na úrovni reportu, je přístupná pouze pro daný report.
- **Object navigator, tr. SE80** - souhrnný nástroj, který logicky integruje většinu z výše uvedených nástrojů [2]

2.2.3 Databázové tabulky systému SAP

Aby aplikace mohla zpracovávat data, musí být tato data uložena v tabulkách. Rozlišujeme 3 základní druhy tabulek, a to:

- Transparentní tabulky
- Tabulky poolu
- Clusterové tabulky

V rámci aplikace CarControl budou využívány pouze transparentní tabulky, jejichž fyzická struktura v databázi odpovídá přesně definici v ABAP Dictionary. U tabulek poolu a u clusterových tabulek jsou data uložena přímo ve fyzické databázi.

Pro vlastní vytvoření tabulky v SAPu používáme transakci SE11. Jedna z tabulek, kterou aplikace využívá, je tabulka ZCAR_CARS, která slouží pro uložení dat uživatelů systému CarControl.

Při definici tabulky se v SAPu postupuje podobně jako v jiných databázových systémech, kdy je potřeba definovat mimo jiné tyto položky:

- Pole – vlastní atribut tabulky
- Key – určuje, jestli dané pole bude primárním klíčem. Primárním klíčem může být více polí, musí být ovšem uvedeny za sebou.
- Iniciální hodnoty – hodnota určuje, jestli pole bude povinně vyplněno
- Datový prvek – určuje vlastnosti pole, tj. o jaký datový typ se jedná, v jaké délce, apod.

Po definici jednotlivých polí tabulky je potřeba definovat technické nastavení tabulky. Toto nastavení je důležité, protože definujeme paměťové parametry, jakým způsobem má být datový záznam během čtení ukládán do vyrovnávací paměti. V technických nastaveních definujeme:

- Datový druh: APPL0, APPL1, APPL2, DDIM, atd., dle toho, jedná-li se o transparentní tabulku pro kmenová nebo pohybová data, tabulku dimenzí, atd.
- Kategorie velikosti: 0 – 8, volíme předpokládaný počet záznamů v tabulce. Při založení tabulky je pro tuto rezervováno v databázi počáteční místo.
- Použití bufferu: zde definujeme, jestli dané tabulce přidělíme buffer. Tím určíme, jestli daná tabulka bude při čtení nahrána do paměti celá, nebo jen její část. Volba

má vazbu na výkon systému, je vhodné ji zapínat pro tabulky s větším počtem záznamů. [2]

Po definici technických nastavení je potřeba tabulku Aktivovat, čímž se změní status tabulky na Active a od tohoto okamžiku je možno tabulku používat.

Obsah takto vytvořené tabulky je možno prohlížet pomocí transakce SE16N. Potřebujeme-li do tabulky zapisovat data, musíme použít příkazů jazyka SQL. Někdy je ale potřeba do tabulky zapsat data mimo vlastní reporty pomocí transakce SM30, např. jedná-li se o customizační tabulku, která může sloužit pro různá nastavení. V tomto případě je potřeba přes Generátor údržby tabulek definovat dialog údržby. Jednou z položek, kterou je potřeba určit, je druh obrazovky údržby:

- Je-li zvolena jednokroková obrazovka údržby, je vygenerována pouze jedna obrazovka, kdy záznamy jsou zobrazeny a upravovány ve formě seznamu.
- Je-li zvolena dvoustupňová obrazovka údržby, jsou generovány dvě obrazovky údržby. Jedna pro přehledovou tabulku záznamů a druhá pro zobrazení a úpravu detailu jednotlivého záznamu.[3]

Po jeho definici bude možno přes transakci SM30 definovat data do tabulky ručně.

2.2.4 Interní tabulky

Interní tabulky chápeme jako datové struktury pro dočasné uložení dat po dobu běhu reportu. Velká výhoda interních tabulek spočívá v tom, že jsou uloženy v paměti. Přístup k datům je velmi rychlý a nemusí se zatěžovat databázový server SAPu dílčími SQL dotazy. Interní tabulka se definuje klausulí BEGIN OF:

```
DATA: BEGIN OF ls_rep1,  
      jednotka_id LIKE zcar_cars-jednotka_id,  
      user_id LIKE zcar_cars-user_id,  
END OF ls_rep1.  
DATA: lt_rep1 LIKE TABLE OF ls_rep1,  
      ll_rep1 LIKE LINE OF lt_rep1.
```

Proměnná *LT_REP1* představuje interní tabulku, která má strukturu definovanou v *LS_REP1*. Proměnná *LL_REP1* představuje pracovní oblast interní tabulky, slouží pro práci se záznamy v interní tabulce.

Pro uložení dat do interní tabulky lze použít několik příkazů, např. příkazem *SELECT * FROM Z_CAR_CARS INTO CORRESPONDING FIELDS OF TABLE LT_REPI* se říká, aby systém vybral SQL dotazem všechna data z databázové tabulky *Z_CAR_CARS* a pole, která jsou totožná jak pro databázovou, tak pro interní tabulku, byla naplněna danými hodnotami. Hodnoty z interní tabulky se získávají buď pomocí smyčky *LOOP ... ENDLOOP*, která prochází celou interní tabulku a testuje aktuální hodnotu na vstupní podmínky, ale lze také využít klauzuli

```
sort lt_zcar_cars by aufnr.  
READ TABLE lt_zcar_cars INTO ll_zcar_cars  
WITH KEY jednotka_id = l_jednotka_id  
BINARY SEARCH.  
IF sy-subrc = 0.  
    l_zak = ll_zcar_cars-aufnr.  
ENDIF.
```

která říká, že do struktury *LL_ZCAR_CARS* systém uloží řádek z interní tabulky *LT_ZCAR_CARS*, kde *JEDNOTKA_ID = L_JEDNOTKA_ID*. Jestliže takový záznam existuje, je systémová proměnná *SY-SUBRC* nastavena na hodnotu 0 a může se tedy lokální proměnná *L_ZAK* naplnit příslušnými daty. Aby vše proběhlo jak má, je vhodné interní tabulku pomocí příkazu *SORT* seřadit dle parametrů, podle kterých je vyhledává.

2.2.5 Události reportu

Události reportu jsou potřebné k seskupování běhu a řízení programu. U jednoduchých reportů je možné vystačit s událostí *START-OF-SELECTION*, můžeme ale využívat i další, a to:

- **LOAD-OF-PROGRAM** – toto slovo definuje blok příkazů, který je spouštěn ABAP-runtime v momentě, kdy je program nahrán do interní session. V rámci bloku lze inicializovat globální datové objekty programu. Při prvním volání externí procedury (podprogramu, funkčního modulu) je program této procedury nahrán do interní session volajícího programu a vyvolána událost *LOAD-OF-PROGRAM*. Blok události je proveden před procedurou.
- **INITIALIZATION** – toto slovo definuje blok příkazů, který je spouštěn ABAP-runtime u programů ihned po *LOAD-OF-PROGRAM* a před zpracováním standardní výběrové obrazovky. Obvykle se v tomto bloku provádí inicializace vstup-

ních polí výběrové obrazovky, případně polí logické databáze, která je propojena s programem.

- AT SELECTION-SCREEN OUTPUT – tato událost je vyvolána během PBO (process before output) výběrové obrazovky. Umožňuje modifikovat výběrovou obrazovku před tím, než je zobrazena.
- AT SELECTION-SCREEN – tato událost je základní formou množiny událostí, které jsou vyvolány během zpracovávání výběrové obrazovky. Po vyvolání výběrové obrazovky a interakci uživatele ABAP runtime generuje události výběrové obrazovky, které se vyskytnou mezi INITIALIZATION a START-OF-SELECTION.
- START-OF-SELECTION – vyvolána, po tom co byla zpracována výběrová obrazovka a případně před čtením dat z logické databáze. V rámci bloku příkazů této události se provádějí v případě logických databází přípravy na čtení dat, u programů bez připojené logické databáze se jedná de facto o hlavní program.
- END-OF-SELECTION – vyvolána, po tom co byla načtena data z logické databáze a před tím, co je řízení předáno list procesoru (odpovědný za výstup sestav na obrazovku). [6]

Pomocí příkazu jazyka ABAP PARAMETERS je možné na výběrové obrazovce deklarovat vstupní pole pro zadávání parametrů – hodnot – pro řízení běhu programu a výběru dat. Příkazem se vytvářejí pole pro zadání pouze jedné hodnoty, kdy tato pole jsou globálními proměnnými v rámci celého reportu.

Parametry lze deklarovat příkazy LIKE nebo TYPE s odkazem na příslušné datové prvky či pole v DDIC nebo jako typy deklarované v programu příkazem TYPES. Délka pojmenování je omezena na 8 znaků.

Příkazem SELECT-OPTIONS je možno vytvářet vstupní pole s komplexnějším zadáním vstupních parametrů. Pole může obsahovat kombinaci zahrnutých/vyloučených hodnot či intervalů hodnot. Vlastně se jedná o interní tabulku s definovanými sloupci SIGN, OPTION, LOW, HIGH. Deklarace se provádí pro pole programu či struktury z DDIC. [6]

2.2.6 Oprávnění

Oprávnění v systému SAP je řešeno na aplikační vrstvě. Aplikace má do databáze plný přístup, přičemž vlastní oprávnění je řízeno na úrovni ABAP příkazu AUTHORITY-CHECK, kdy ošetření této funkce je plně v pravomoci programátora.

V souvislosti s oprávněními jsou definovány:

- Objekty oprávnění
- Profil
- Role

Objekt oprávnění určuje stanovený okruh působnosti, např. možnost spuštění transakce, oprávnění k závodu, materiálu, apod. Objekt oprávnění obsahuje příslušná pole oprávnění, např. objekt S_TCODE, který kontroluje kód transakce pro spuštění, obsahuje pole oprávnění TCD – kód transakce. Kód oprávnění M_MSEG_LGO, které slouží pro pohyby materiálů na skladě, obsahuje pole oprávnění ACTVT – aktivita, WERKS – závod, LGORT – sklad, BWART – druh pohybu. Nad těmito všemi poli oprávnění se řídí činnosti, např.:

- 01 – založení nebo vygenerování
- 02 – změna
- 03 – zobrazení
- 08 - výmaz

Tyto činnosti jsou potom dále testovány ve funkci AUTHORITY-CHECK. Vždy lze povolit přístup pouze k objektu, výsledná oprávnění jsou sjednocením jednotlivých oprávnění. Jednotlivá nastavení objektů jsou definována v profilech. Kontrola oprávnění spočívá v porovnání hodnot přiřazeného profilu oprávnění s hodnotami programu, který se uživatel snaží spustit. Role je objekt, který propojuje příslušný profil s menu, které se uživateli zobrazuje.

Vlastní vyhodnocení oprávnění zabezpečuje funkce AUTHORITY-CHECK.

AUTHORITY-CHECK OBJECT 'M_MSEG_LGO'

ID 'ACTVT' FIELD '_____'

ID 'WERKS' FIELD '_____'

ID 'LGORT' FIELD '_____'

ID 'BWART' FIELD '_____':

Návratové hodnoty:

0 – Objekt nalezen, hodnoty souhlasí (OK)

4 – Objekt nalezen, hodnoty nesouhlasí

12 – Objekt pro uživatele nenalezen

2.2.7 Výměna zpráv s protokolem SOAP

Pro výměnu dat mezi externími aplikacemi ve formátu XML lze použít protokolu SOAP. Pomocí tohoto protokolu lze zprávy přenášet pomocí protokolů, které jsou běžně používány (HTTP/HTTPS, SMTP, FTP). Každá zpráva SOAP je zabalena do obalu, který je známý jako *obálka*. Obálka zodpovídá za uchování všech částí zprávy. Každá zpráva může obsahovat *hlavičku*, což je oblast pro uchování metadat. Obsah zprávy je uložen v *těle zprávy*, které se běžně skládá z formátovaných dat XML. Základní vlastností komunikace SOAP architekturou je důraz na tvorbu zpráv, které budou samostatné a inteligentní. To vede k tomu, že zprávy SOAP dosahují takové úrovně nezávislosti, která zvyšuje robustnost a rozšiřitelnost systému výměny zpráv. Tato nezávislost je implementována pomocí *bloku hlaviček*. Bloky hlaviček vybavují zprávu všemi informacemi vyžadovanými službami, se kterými zpráva přijde do styku, aby mohly zpracovat a směřovat danou zprávu v souladu s jejími pravidly, instrukcemi a pravidly. To znamená, že díky blokům hlaviček můžou SOAP zprávy obsahovat velké množství pomocných informací, které souvisí s doručením a zpracováním obsahu zpráv. Zprávy SOAP umožňují přidat logiku *výjimek* tak, že to tělo zprávy přidáme nepovinnou chybovou část. Do této části lze většinou uložit jednoduchou zprávu, která má objasnit příčinu selhání doručení, pokud nastane výjimka.

[4]

II. PRAKTICKÁ ČÁST

3 NÁVRH ARCHITEKTURY UŽIVATELSKÉHO MODULU

Transakce ZCAR není pouze běžným reportem, který na základě definice vstupních parametrů zobrazí data. Nedílnou součástí je Administrace, která slouží pro import dat a jejich uložení do uživatelských tabulek. Jedná se o data:

- Uživatelé systému CarControl
- Vozidla v systému CarControl
- Data o provozu
- Data o nákladech
- Mapování účtů

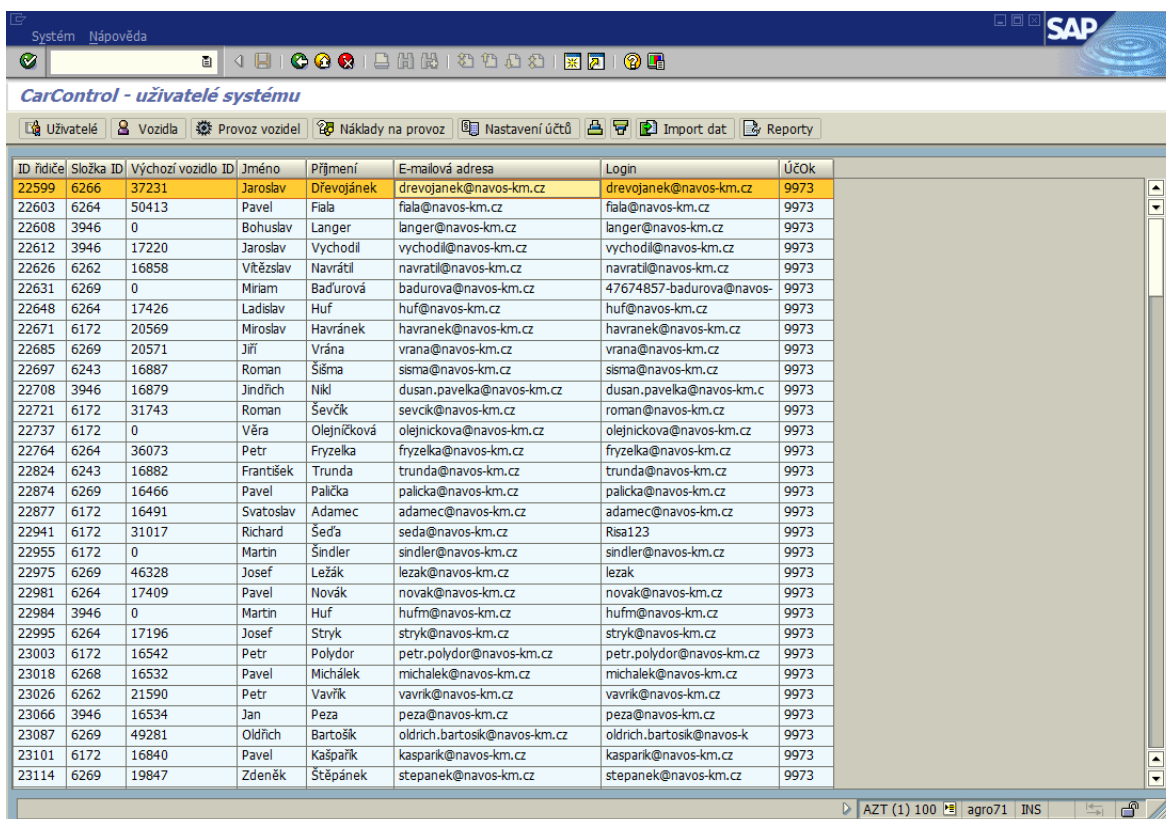
Jelikož aplikace bude v ostrém provozu nasazena na systému SAP AZP, na kterém na klientu 100 běží 25 účetních okruhů, je počítáno s tím, že report bude využívat vícero společností. Proto musí být zajištěno, že záznamy v databázových tabulkách budou mít pole BUKRS, kde bude uvedeno číslo daného účetního okruhu a vlastní výběr dat bude probíhat v závislosti na zvoleném účetním okruhu. Nepočítá se tedy s tím, že data budou zobrazována křížem přes všechny účetní okruhy, ale vždy jen za zvolený účetní okruh. Nad celým reportem jsou aplikována přístupová práva. Definováním rolí je zajištěno, že se do reportů i administrace dostanou jen určené pracovníci.

Na systému AZP pracuje 25 společností se svými kompletními účetními daty a sdílí veškerý programový kód systému. Každý špatně napsaný report, špatně nadefinovaný select do databáze, apod. znamená snížení výkonu databázového serveru, což má za následek snížení rychlosti odezvy celého systému. Při psaní každého reportu by měl tedy mít programátor na paměti nároky na výkon. V aplikaci ZCAR je administrace napsána jako samostatný modul s vlastním oprávněním, kdy do jednotlivých částí se uživatel dostane klikem na tlačítka Uživatelé, Vozidla, Provoz vozidel a Náklady na provoz. Kdyby při každém kliku došlo k výběru příslušných dat selectem do databáze, docházelo by ke zbytečnému zatěžování databázového serveru. Tohoto faktu je využito i v reportech, kde jsou hlavní data nachystána v interních tabulkách a přes select se doplňují data dle požadavků ve vstupních parametrech reportu.

Logika dat postavena následovně:

- Při spuštění reportu je vstupním povinným parametrem číslo účetního okruhu

- Pod tímto číslem jsou do příslušných interních tabulek načtena data uživatelů, vozidel, skupiny namapovaných účtů a index pro import provozních dat a nákladů. Tato data jsou tedy načtena pouze jednou, a jsou dostupná po celou dobu práce s reportem, nemusí se opakovaně číst z databáze
- Data, která se načítají dle vstupních parametrů, jako např. data o provozu za zvolený měsíc nebo data ze zakázek ze SAPu, se pak už načítají přes SQL příkazy z databáze.



The screenshot shows the SAP CarControl - uživatelé systému window. The window has a menu bar with options like 'Uživatelé', 'Vozidla', 'Provoz vozidel', 'Náklady na provoz', 'Nastavení účtů', 'Import dat', and 'Reporty'. Below the menu bar is a table with the following columns: ID řidiče, Složka ID, Výchozí vozidlo ID, Jméno, Příjmení, E-mailová adresa, Login, and Účok. The table contains 25 rows of user data.

ID řidiče	Složka ID	Výchozí vozidlo ID	Jméno	Příjmení	E-mailová adresa	Login	Účok
22599	6266	37231	Jaroslav	Dřevojanek	drevojanek@navos-km.cz	drevojanek@navos-km.cz	9973
22603	6264	50413	Pavel	Fiala	fiala@navos-km.cz	fiala@navos-km.cz	9973
22608	3946	0	Bohuslav	Langer	langer@navos-km.cz	langer@navos-km.cz	9973
22612	3946	17220	Jaroslav	Vychodil	vychodil@navos-km.cz	vychodil@navos-km.cz	9973
22626	6262	16858	Vítězslav	Navrátil	navratil@navos-km.cz	navratil@navos-km.cz	9973
22631	6269	0	Miriám	Badurová	badurova@navos-km.cz	47674857-badurova@navos-	9973
22648	6264	17426	Ladislav	Huf	huf@navos-km.cz	huf@navos-km.cz	9973
22671	6172	20569	Miroslav	Havranek	havranek@navos-km.cz	havranek@navos-km.cz	9973
22685	6269	20571	Jiří	Vrána	vrana@navos-km.cz	vrana@navos-km.cz	9973
22697	6243	16887	Roman	Šisma	sisma@navos-km.cz	sisma@navos-km.cz	9973
22708	3946	16879	Jindřich	Nikl	dusan.pavelka@navos-km.cz	dusan.pavelka@navos-km.c	9973
22721	6172	31743	Roman	Ševčík	sevck@navos-km.cz	roman@navos-km.cz	9973
22737	6172	0	Věra	Olejníčková	olejnickova@navos-km.cz	olejnickova@navos-km.cz	9973
22764	6264	36073	Petr	Fryzelka	fryzelka@navos-km.cz	fryzelka@navos-km.cz	9973
22824	6243	16882	František	Trunda	trunda@navos-km.cz	trunda@navos-km.cz	9973
22874	6269	16466	Pavel	Palčka	palicka@navos-km.cz	palicka@navos-km.cz	9973
22877	6172	16491	Svatoslav	Adamec	adamec@navos-km.cz	adamec@navos-km.cz	9973
22941	6172	31017	Richard	Šedá	seda@navos-km.cz	Risa123	9973
22955	6172	0	Martin	Sindler	sindler@navos-km.cz	sindler@navos-km.cz	9973
22975	6269	46328	Josef	Ležák	lezak@navos-km.cz	lezak	9973
22981	6264	17409	Pavel	Novák	novak@navos-km.cz	novak@navos-km.cz	9973
22984	3946	0	Martin	Huf	hufm@navos-km.cz	hufm@navos-km.cz	9973
22995	6264	17196	Josef	Stryk	stryk@navos-km.cz	stryk@navos-km.cz	9973
23003	6172	16542	Petr	Polydor	petr.polydor@navos-km.cz	petr.polydor@navos-km.cz	9973
23018	6268	16532	Pavel	Michálek	michalek@navos-km.cz	michalek@navos-km.cz	9973
23026	6262	21590	Petr	Vavřík	vavrik@navos-km.cz	vavrik@navos-km.cz	9973
23066	3946	16534	Jan	Peza	peza@navos-km.cz	peza@navos-km.cz	9973
23087	6269	49281	Oldřich	Bartošik	oldrich.bartosik@navos-km.cz	oldrich.bartosik@navos-k	9973
23101	6172	16840	Pavel	Kašpařík	kaspark@navos-km.cz	kaspark@navos-km.cz	9973
23114	6269	19847	Zdeněk	Štěpánek	stepanek@navos-km.cz	stepanek@navos-km.cz	9973

Obr. 4. Transakce ZCAR - okno administrace

3.1 Datová struktura aplikace ZCAR

Aplikace ZCAR používá pro zpracování dat následující tabulky:

Tab. 1. ZCAR_USERS: uživatelé systému CarControl

Pole	Klíč	Inic.hod.	Dat. prvek	Dat. typ	Popis
MANDT	A	A	MANDT	CLNT	Klient
BUKRS	A	A	BUKRS	CHAR	Účetní okruh
USER_ID	A	A	Z_USER_ID	CHAR	ID řidiče
FOLDER_ID			Z_FOLDER_ID	CHAR	Složka ID
DEFAULTCAR_ID			Z_DEFAULTCAR_ID	CHAR	Výchozí vozidlo ID
NAME_FIRST			NAME_FIRST	CHAR	Jméno
NAME_LAST			NAME_LAST	CHAR	Příjmení
SMTP_ADDR			AD_SMTPADR	CHAR	E-mailová adresa
LOGIN			Z_LOGIN	CHAR	Login

Tab. 2. ZCAR_CARS: vozidla v systému CarControl

Pole	Klíč	Inic.hod.	Dat. prvek	Dat. typ	Popis
MANDT	A	A	MANDT	CLNT	Klient
BUKRS	A	A	BUKRS	CHAR	Účetní okruh
JEDNOTKA_ID	A	A	Z_JEDNOTKA_ID	CHAR	Jednotka ID
SPZ	A	A	ZSPZ	CHAR	SPZ vozidla
USER_ID	A	A	Z_USER_ID	CHAR	ID řidiče
NAME			Z_NAZEV	CHAR	Název
VIN			Z_VIN	CHAR	VIN
VYROBCE			Z_VYROBCE	CHAR	CarControl - výrobce
ZNACKA			Z_ZNACKA	CHAR	CarControl - značka
MODEL			Z_MODEL	CHAR	CarControl - Model
PALIVO			Z_PALIVO	CHAR	CarControl - palivo
SPOTREBA			Z_SPOTREBA	DEC	CarControl - spotřeba
NADRZ			Z_NADRZ	CHAR	CarControl - velikost nádrže
TACHOMETR			Z_TACHOMETR1	DEC	CarControl - Tachometr
DRUHVOZIDLA			Z_DRUHVOZIDLA	CHAR	CarControl - Druh vozidla

Tab. 3. ZCAR_ZAK: vazba mezi vozidlem v CarControlu a controllingovou tabulkou v SAPu

Pole	Klíč	Inic.hod.	Dat. prvek	Dat. typ	Popis
MANDT	A	A	MANDT	CLNT	Klient
BUKRS	A	A	BUKRS	CHAR	Účetní okruh
JEDNOTKA_ID	A	A	Z_JEDNOTKA_ID	CHAR	Jednotka ID
AUFNR	A	A	AUFNR	CHAR	Číslo zakázky

Tab. 4. ZCAR_IMP_INDEX: časový index pro import provozních dat

Pole	Klíč	Inic.hod.	Dat. prvek	Dat. typ	Popis
MANDT	A	A	MANDT	CLNT	Klient
BUKRS	A	A	BUKRS	CHAR	Účetní okruh
ID	A	A	INTEGER	INT4	Celé číslo
TYP	A	A	CHAR1	CHAR	Typ importu
MESIC			LFMON	NUMC	Aktuální období (účetní období)
ROK			LFGJA	NUMC	Fiskální rok současného období

Tab. 5. ZCAR_IMP_DATA: importovaná data o provozu a nákladech vozidel

Pole	Klíč	Inic.hod.	Dat. prvek	Dat. typ	Popis
MANDT	A	A	MANDT	CLNT	Klient
BUKRS	A	A	BUKRS	CHAR	Účetní okruh
SPZ	A	A	ZSPZV	CHAR	SPZ vozidla
ID	A	A	INTEGER	INT4	Celé číslo
MESIC	A	A	LFMON	NUMC	Aktuální období (účetní období)
ROK	A	A	LFGJA	NUMC	Fiskální rok současného období
TYP	A	A	CHAR1	CHAR	Typ importu
NAME			Z_NAZEV	CHAR	Název
TACHOMETR			Z_TACHOMETR1	DEC	CarControl - Tachometr
DELKA_OSOBNI			Z_DELKA_OSOBNI	DEC	soukromé KM
DELKA_FIREMNI			Z_DELKA_FIREMNI	DEC	fíremní KM
DELKA_SUMA			Z_DELKA_SUMA	DEC	Suma KM
LITRY			Z_LITRY	DEC	Počet litrů
DPH_SAZBA			Z_DPH_SAZBA	DEC	Sazba DPH

CASTKA_DPH			Z_CASTKA_DPH	DEC	Částka DPH
CENA_PALIVO			Z_CENA_PALIVO	DEC	Cena paliva
CENA_ZAKLAD			Z_CENA_ZAKLAD	DEC	Cena základ
CENA_CELKEM			Z_CENA_CELKEM	DEC	Cena celkem
TANKOVANI			Z_TANKOVANI	CHAR	Tankování

Tab. 6. ZCAR_IMP_UCTY: tabulka pro mapování účtů

Pole	Klíč	Inic.hod.	Dat. prvek	Dat. typ	Popis
MANDT	A	A	BUKRS	INT4	Celé číslo
BUKRS	A	A	BUKRS	CHAR	SPZ vozidla
KEY0	A	A	CHAR10	CHAR	Účetní okruh
NAME	A	A	CHAR25	CHAR	Jméno mapy
MNOZSTVI	A	A	CHAR1	CHAR	Sloupec množství
ID	A	A	Z_CAR_IMPORTID	INT4	CarControl - ID importu
SIGN			CHAR0001	CHAR	Jednotl., libovolný znak
OPTION0			CHAR2	CHAR	Jednotl., libovolný znak
LOW			CHAR0010	CHAR	Jednotl., libovolný znak
HIGH			CHAR0010	CHAR	Jednotl., libovolný znak

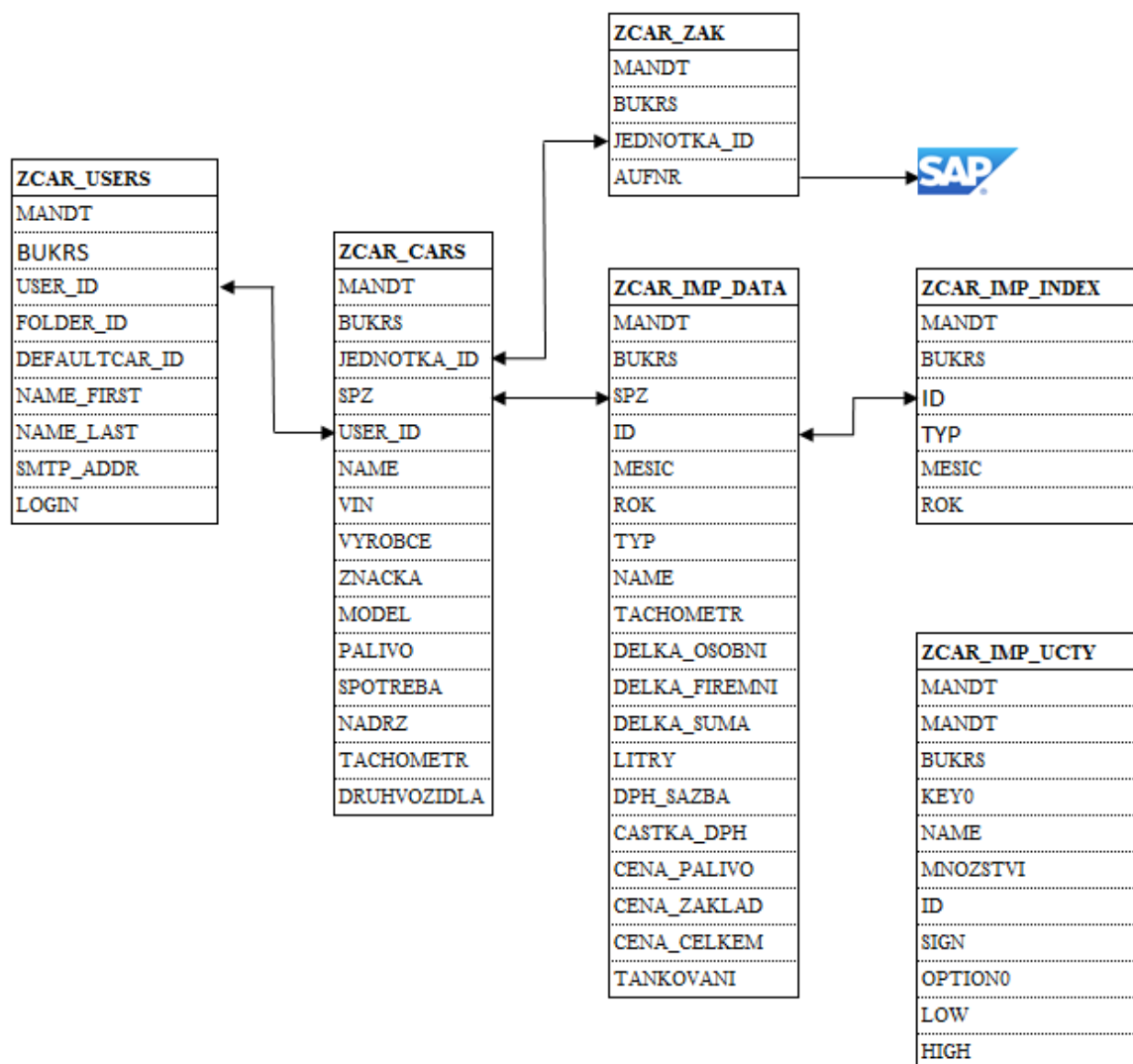
3.2 Vazby mezi tabulkami

Jedná se o následující vazby:

- Základní data o vozidlech jsou uložena v tabulce ZCAR_CARS. Primárním klíčem v tabulce je vazba MANDT-BUKRS-JEDNOTKA_ID-SPZ-USER_ID, tedy klient, účetní okruh, ID jednotky CarControl, Spz a ID řidiče.
- V tabulce ZCAR_USERS jsou definovány pole pro práci s uživateli systému. Tabulka má primární klíč MANDT-BUKRS-USER_ID, mezi další pole patří NAME_LAST, NAME_FIRST, LOGIN, atd. Přes pole USER_ID je tabulka propojena s polem USER_ID v tabulce ZCAR_CARS.
- Tabulka ZCAR_ZAK je propojovací tabulka, která obsahuje vazbu mezi polem JEDNOTKA_ID v tabulce ZCAR_CARS a SAPí tabulkou GLPCA a polem AUFNR. Primárním indexem v tabulce ZCAR_ZAK je vazba MANDT-BUKRS-JEDNOTKA-ID-AUFNR.

- Tabulky ZCAR_IMP_INDEX a ZCAR_IMP_DATA slouží pro uchování dat o provozu a nákladech vozidel. Tabulka ZCAR_IMP_DATA obsahuje vlastní data o měsíčním provozu a nákladech vozidel, tabulka ZCAR_IMP_INDEX obsahuje číselník měsíců, za které byl import proveden. Obě tabulky jsou provázány přes pole ID. Tabulka ZCAR_IMP_INDEX má definován primární index jako MANDT-
BUKRS-ID-TYP, tabulka ZCAR_IMP_DATA má primární index MANDT-
BUKRS-SPZ-ID-MESIC-ROK-TYP a obsahuje informace o SPZ, počtu soukromých kilometrů, firemních kilometrů, celkový počet ujetých kilometrů za daný měsíc a data o tankování za daný měsíc a vozidlo. Tato data se dále používají v reportu nákladů. Součástí obou tabulek je pole TYP, které definuje, jedná-li se o záznam za provoz vozidla nebo za náklad na vozidlo.
- V reportu pro vyčíslení nákladů jsou počítány náklady za pohonné hmoty, odpisy, ostatní náklady, atd. Aby tato data mohla být analyzována, je potřeba pro jednotlivé položky držet seznam příslušných účtů. Tyto účty nemůžeme definovat napevno, protože se můžou lišit v závislosti na účetním okruhu. Je proto definována tabulka ZCAR_UCTY, která nese informaci o těchto účtech.

Všechny tabulky obsahují pole MANDT, které definuje, že tabulky jsou klientově závislé.

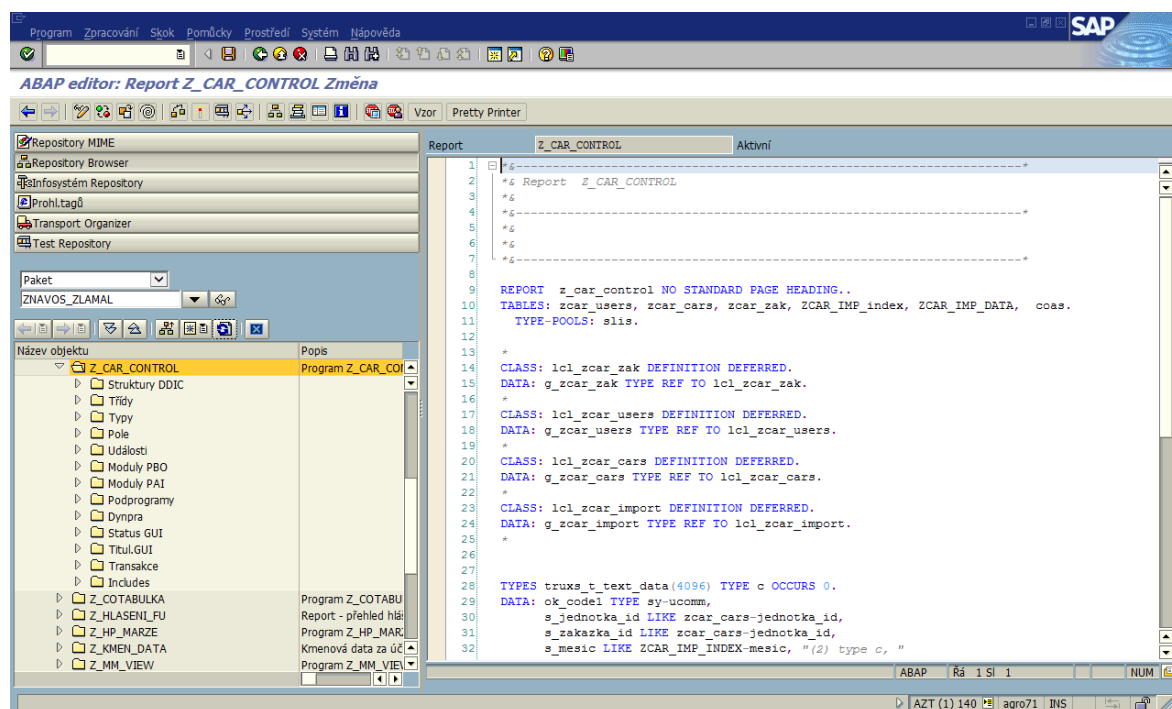


Obr. 5. Vazby mezi databázovými tabulkami transakce ZCAR

4 REALIZACE UŽIVATELSKÉHO MODULU

4.1 Uspořádání vstupních souborů

Vlastní kód reportu je psán přes transakci SE80 v ABAP DEVELOPMENT WORKBENCH.



Obr. 6. Okno vývojového prostředí ABAP DEVELOPMENT WORKBENCH v SAP

Pro přehlednost jsou jednotlivé objekty reportu organizovány v následujících kontejnerech:

- Struktury DDIC: seznam tabulek, které report využívá.
- Třídy: seznam tříd včetně jejich definicí a implementací.
- Typy: datové typy použité v reportu
- Pole: přehled proměnných
- Události: události reportu
- Moduly PBO: *process before output*. Moduly u dialogových oken-dynper, definují se zde vstupní parametry pro dynpra včetně PF-Statusů a titulku oken.
- Moduly PAI: *process after input*. Moduly, které zpracovávají události dynpra
- Podprogramy: podprogramy v rámci reportu
- Dynpra: výběrová dialogová okna

- Status GUI: ovládací ikony reportu včetně dynper
- Status GUI: titulky reportu včetně dynper
- Transakce: definovaná transakce reportu
- Includes: vložené soubory v reportu

Vstupní data aplikace jsou uložena v souboru *Z_CAR_CONTROL*, který dále obsahuje deklarace na jednotlivé třídy a globální proměnné. Mezi hlavní deklarace patří:

CLASS: lcl_zcar_zak DEFINITION DEFERRED.

DATA: g_zcar_zak TYPE REF TO lcl_zcar_zak.

CLASS: lcl_zcar_users DEFINITION DEFERRED.

DATA: g_zcar_users TYPE REF TO lcl_zcar_users.

CLASS: lcl_zcar_cars DEFINITION DEFERRED.

DATA: g_zcar_cars TYPE REF TO lcl_zcar_cars.

CLASS: lcl_zcar_import DEFINITION DEFERRED.

DATA: g_zcar_import TYPE REF TO lcl_zcar_import.

CLASS: lcl_zcar_coast DEFINITION DEFERRED.

DATA: g_zcar_coast TYPE REF TO lcl_zcar_coast.

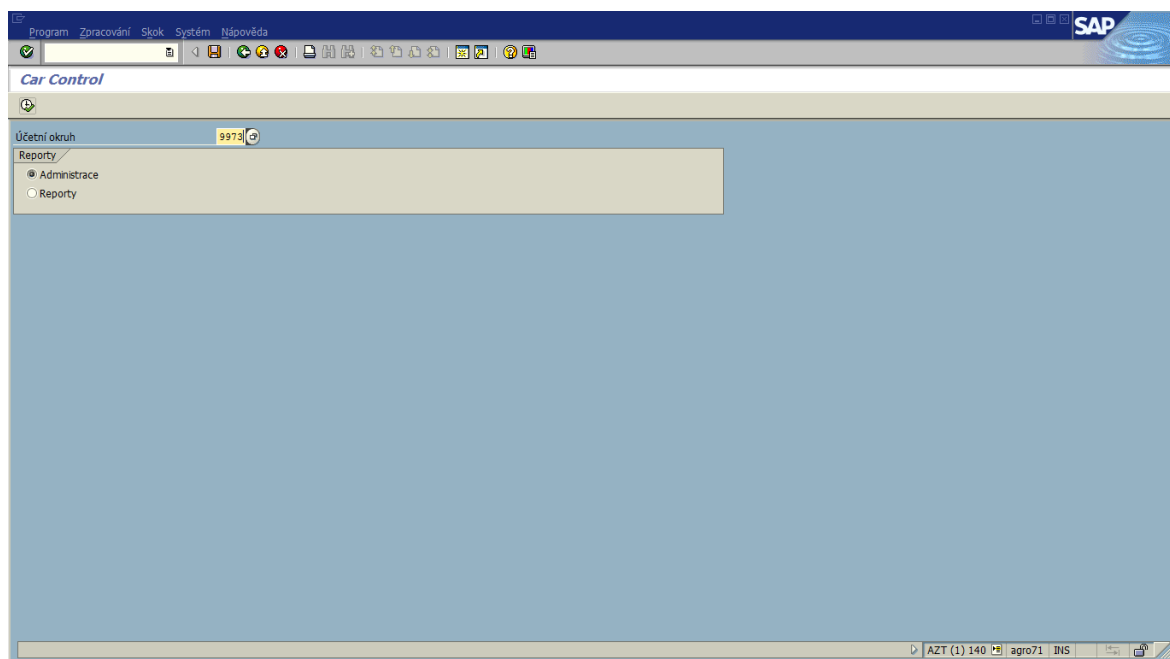
Pro přehlednost jsou jednotlivé oblasti reportu organizovány do jednotlivých include.

- *INCLUDE z_car_user* – definice a implementace třídy *lcl_zcar_users*, která zpracovává uživatele systému
- *INCLUDE z_car_car* – definice a implementace třídy *lcl_zcar_cars*, která zpracovává vozidla systému
- *INCLUDE z_car_zakazka* - definice a implementace třídy *lcl_zcar_zak*, která propojuje controllingové zakázky s vozidly
- *INCLUDE z_car_coast*- definice a implementace třídy *lcl_zcar_coast*, která zpracovává účty, za které se potom následně počítají reporty
- *INCLUDE z_car_import* - definice a implementace třídy *lcl_zcar_import*, která zpracovává import provozních dat
- *INCLUDE z_car_control_alv* - definice a implementace třídy *lcl_alv*, která zpracovává zobrazování dat v administraci přes ALV

- *INCLUDE Z_CAR_REP_ONLINE* – report online nahlížení do provozních dat přes webovou službu
- *INCLUDE z_car_control_scr* – výběrová obrazovka reportu
- *INCLUDE z_car_control_status_1010o01* – PBO a PAI moduly jednotlivých dynper
- *INCLUDE z_car_control_user_command_i01.*
- *INCLUDE z_car_control_status_1020o01.*
- *INCLUDE z_car_control_status_1030o01.*
- *INCLUDE z_car_control_status_1040o01.*
- *INCLUDE Z_CAR_CONTROL_STATUS_1050O01.*
- *INCLUDE z_car_rep1* – report statistiky vozidel
- *INCLUDE z_car_rep2* – report nákladů
- *INCLUDE z_car_rep_online* – ON-LINE pohled na provoz
- *INCLUDE Z_CAR_REP_ONLINE_NAKL* – ON-LINE pohled na náklady

4.2 Události reportu a definice polí výběrových obrazovek

V reportu ZCAR jsou pole výběrové obrazovky definovány příkazem PARAMETERS i SELECT-OPTIONS. Příkazem PARAMETR je definována globální proměnná *g_bukrs*, která si nese číslo účetního okruhu. Parametr OBLIGATORY znamená, že pole je povinné. V sekci SELECTION-SCREEN jsou definovány objekty typu checkbox, kterými je řízena logika reportu. Dle volby checkboxů dojde k otestování oprávnění, jestliže uživatel má právo, spustí se administrace nebo reporty.



Obr. 7. Transakce ZCAR - úvodní okno reportu

4.3 Administrace

Po spuštění administrace se objeví ovládací okno, kde je v toolbaru uveden seznam ovládacích tlačítek. V závislosti na volbě se potom zobrazuje seznam příslušných dat.

4.3.1 Ovládání administrace

Logika ovládání je definována v metodách lokální třídy *LCL_ALV*. Veškeré ovládání je řešeno přes obecnou třídu *CL_SALV_TABLE*. Pro vlastní ovládání jsou registrovány jednotlivé obsluhy událostí:

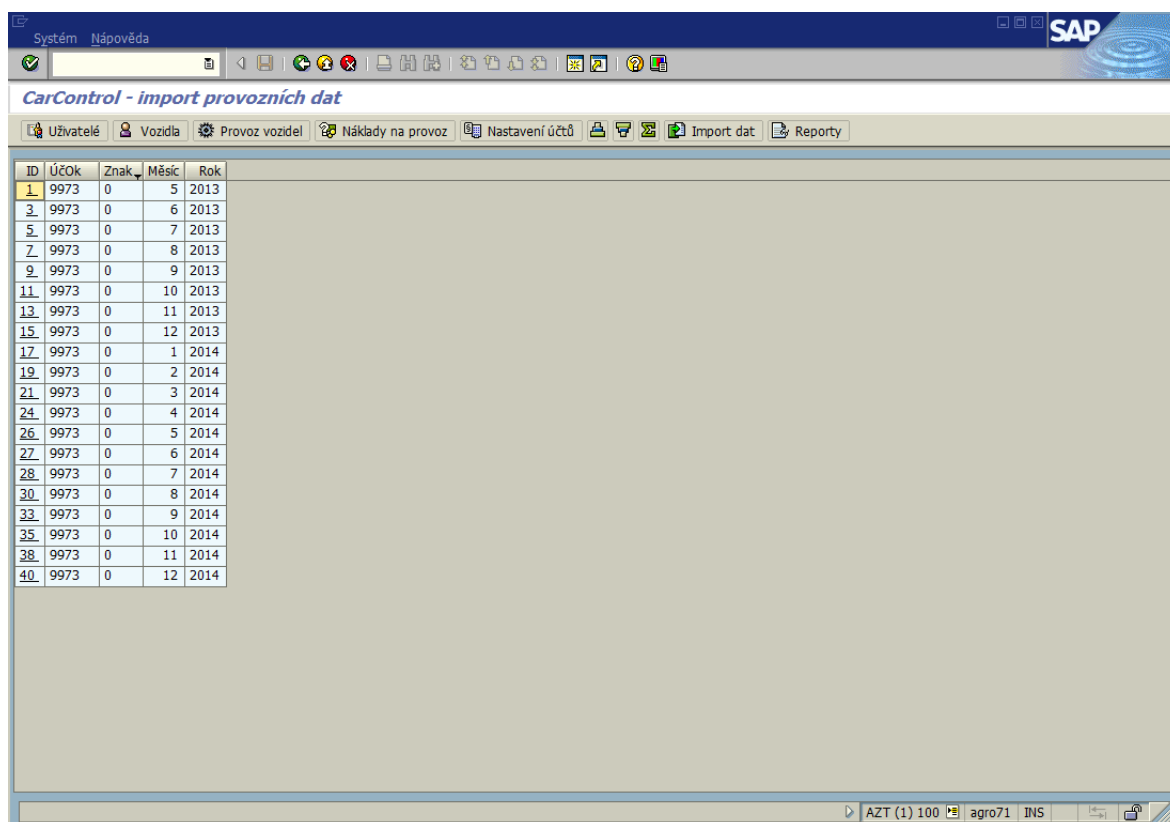
```
o_events = o_table->get_event( ).  
    SET HANDLER me->on_link_click FOR o_events.  
    SET HANDLER me->on_double_click FOR o_events.  
    SET HANDLER me->on_function_click FOR o_events.
```

Metodou *me->on_function_click* je řízeno ovládání administrace přes funkční tlačítka. Každé tlačítko má definované své ID, které je při kliku předáno v parametru *e_salv_function* metodě *ON_FUCTION_CLICK*. Jedná-li např. o překlik ze seznamu uživatelů do seznamu vozidel, metoda zabezpečí naplnění proměnné *ID_VIEW*, kterou si přebírá metoda *DISPLAY_ALV*, která zabezpečuje vlastní zobrazování dat. V případě, že se jedná o klik např. do importu, metoda zabezpečí zobrazení importního dialogu a předá ří-

zení reportu metodě `g_zcar_import->create_import_dialog`, která zpracuje vlastní import dat.

Metoda `DISPLAY_ALV` volá funkční modul `cl_salv_table=>factory`, který dle parametru `id_view` připojí na vstup interní tabulku uživatelů, vozidel nebo indexu provozních dat. Dále metoda vrací objekt `o_table`, který je instancí třídy `cl_salv_table`, který obsahuje metody a vlastnosti pro vlastní zobrazení dat.

Metoda `me->on_link_click` zabezpečuje obsluhu jednoduchého kliku, který je použit u náhledu na provoz vozidel a nákladů za vozidla. Kvůli přehlednosti jsou tyto dva pohledy řešeny tak, že se nejprve objeví seznam měsíců, za které byly provedeny importy provozu a nákladů.



ID	ÚČOK	Znak	Měsíc	Rok
1	9973	0	5	2013
2	9973	0	6	2013
5	9973	0	7	2013
7	9973	0	8	2013
9	9973	0	9	2013
11	9973	0	10	2013
13	9973	0	11	2013
15	9973	0	12	2013
17	9973	0	1	2014
19	9973	0	2	2014
21	9973	0	3	2014
24	9973	0	4	2014
26	9973	0	5	2014
27	9973	0	6	2014
28	9973	0	7	2014
30	9973	0	8	2014
33	9973	0	9	2014
35	9973	0	10	2014
38	9973	0	11	2014
40	9973	0	12	2014

Obr. 8. Transakce ZCAR - okno importu provozních dat

Teprve po kliku na ID u zvoleného měsíce se objeví seznam importovaných dat za daný měsíc.

Obsluha metody si dle čísla řádku *row* zjistí ID záznamu a ten jako parametr předá metodě *g_zcar_import->read_car_data*. Tato metoda provede načtení dat a předá řízení metodě *DISPLAY_ALV*, která zabezpečí zobrazení dat.

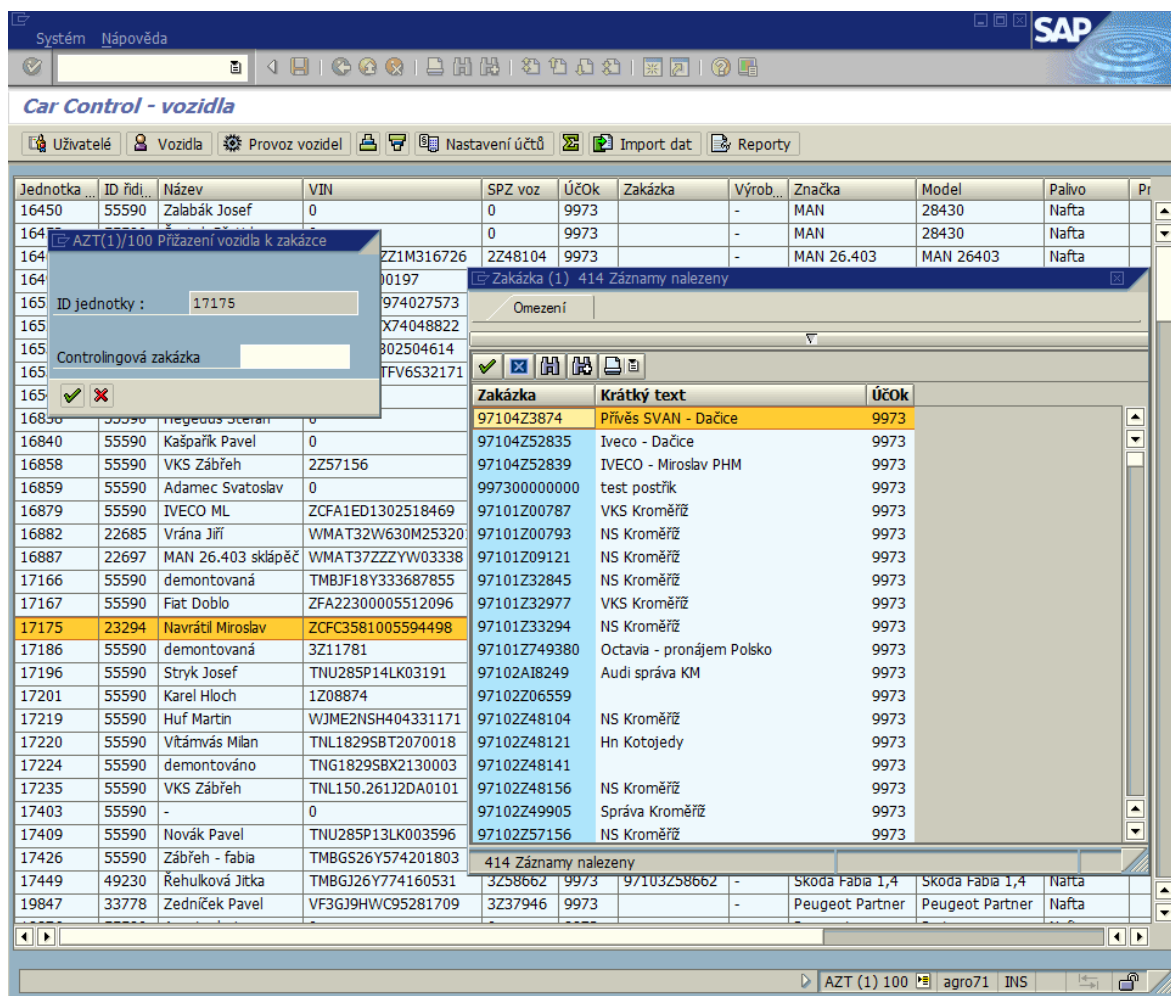
```
READ TABLE g_zcar_import->lt_zcar_import  
            INTO g_zcar_import->ll_zcar_import INDEX row.  
g_zcar_import->read_car_data( g_zcar_import->ll_zcar_import-id ).
```

Metoda *me->on_double_click* obsluhuje událost dvojitého kliku. Toto je využito při volání dialogových oken pro definici controllingové zakázky nebo importu dat. V obou případech dochází k inicializaci vstupních parametrů a vlastní volání dialogového okna příkazem

```
CALL SCREEN 'XXXX' STARTING AT XX X.
```

4.3.2 Propojení na SAP přes controllingovou zakázku

Propojení vozidla s příslušnou controllingovou zakázkou je řešeno v administraci na úrovni zobrazení seznamu vozidel. Při dvojitém kliku na daném vozidle se otevře dialogové okno, kde je uvedeno ID komunikační jednotky a okno pro definici controllingové zakázky. Pro výběr zakázky lze použít matchcode, který zobrazí seznam controllingových zakázek za daný účetní okruh. Po potvrzení výběru dojde k uložení dat.



Obr. 9. Transakce ZCAR - přiřazení controllingové zakázky k vozidlu

Dialogové okno – dynpro je voláno příkazem *CALL SCREEN '1010'*, kde 1010 je ID daného dynpra. Každé volané dynpro má sekce PBO a PAI pro své ovládání, a to:

PROCESS BEFORE OUTPUT.

MODULE STATUS_1010.

PROCESS AFTER INPUT.

MODULE USER_COMMAND_1010.

PROCESS ON VALUE-REQUEST.

FIELD S_ZAKAZKA_ID MODULE s_zakazka_f4.

V případě propojení zakázek se při otevření dynpra volá obsluha STATUS_1010, která nastaví vlastnosti dynpra ještě před jeho zobrazením, tj. přes PF-STATUS zobrazí ovládací tlačítka dynpra, přes TITLEBAR nastaví titulek a dále provede nastavení okna s ID jednotky na READ-ONLY.

```
MODULE STATUS_1010 OUTPUT.  
  SET PF-STATUS 'Z_CAR_CONTROL_ZAL'.  
  SET TITLEBAR 'Z_CAR_CONTROL_ZAK_TI'.  
  DATA screen_wa TYPE screen.  
  LOOP AT SCREEN INTO screen_wa.  
    IF screen_wa-name = 'S_JEDNOTKA_ID'.  
      screen_wa-input = '0'.  
      MODIFY SCREEN FROM screen_wa.  
    ENDIF.  
  ENDLOOP.  
ENDMODULE.
```

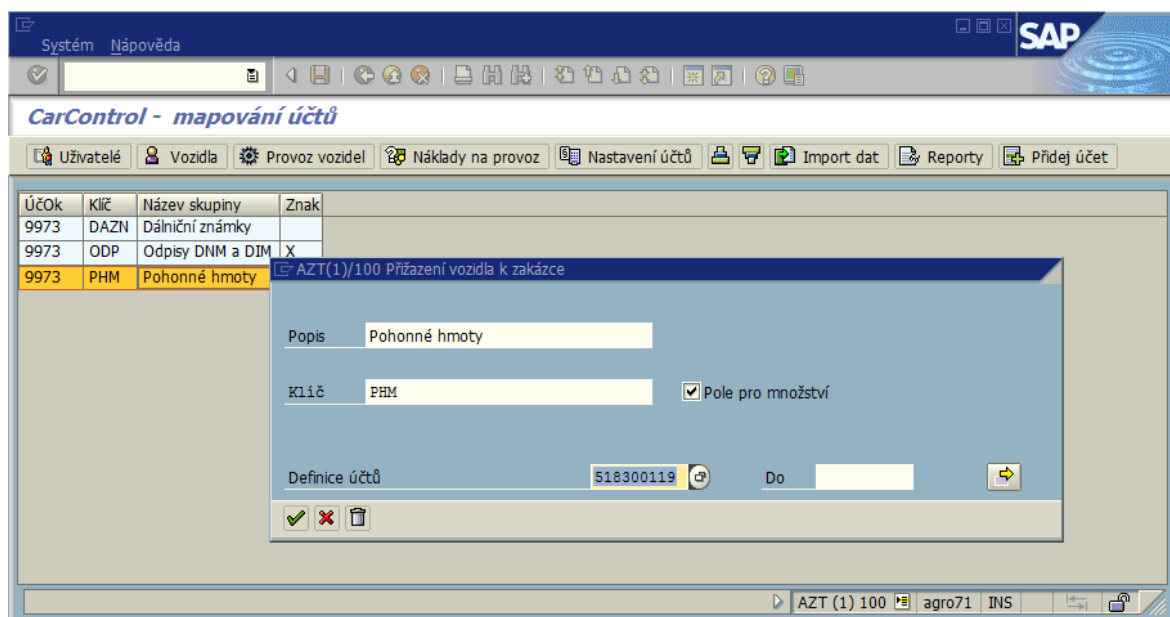
Modul `USER_COMMAND_1010` zabezpečí obsluhu dynpra při jeho ukončení. Jestliže uživatel zvolí tlačítko `SAVE`, bude metodou `CHECK_EXIST` otestováno, jestliže vozidlo má záznam v interní tabulce `LT_ZCAR_ZAK`. Jestliže ne, jedná se o prvotní propojení vozidla a zakázky. Jestliže ano, znamená to, že vozidlo má již zakázku definovanou a jedná se o změnu. Dle toho je volána metoda `SAVE_DATA`, která provede příslušný insert nebo update v databázové tabulce `ZCAR_ZAK`. V posledním kroku se dle indexu načte příkazem `READ TABLE` do struktury `LL_ZCAR_CARS` zvolené vozidlo a v položce `AUFNR` je dopsáno číslo zvolené zakázky. Příkazem `LEAVE TO SCREEN 0` dojde k uzavření dynpra.

4.3.3 Mapování účtů

Mapování účtů je důležitou součástí aplikace. Aby měl nákladový report vypovídající hodnotu, je potřeba definovat, za jaké kategorie účtů se mají data načítat. Protože report bude využívat více společností, kdy každá ze společností může mít jiné požadavky na účty nebo skupiny účtů, které mají být reportovány, je logika mapování účtů řešena následovně:

- Za každý účetní okruh je definována skupina účtů nebo několik skupin účtů
- V rámci každé z těchto skupin jsou definovány množiny účtů, které mají vazbu k dané skupině. Množina může být tvořena jednak seznamem jednotlivých účtů nebo rozsahem účtů od-do, které se mají započítávat do reportu.
- Počet těchto skupin není pevně dán, vlastní výpočty i zobrazování jsou prováděny dynamicky.

Klikne-li uživatel na ikonu Mapování účtů, objeví se seznam již vytvořených skupin účtů za daný účetní okruh.



Obr. 10. Transakce ZCAR - mapování účtů

Při definici je potřeba zadat:

- Název skupiny: označení skupiny účtů, tento název je předáván do nákladového reportu do názvu daného sloupce
- Klíč: Třídící klíč, jeho volba záleží na uživateli, dle tohoto klíče jsou prováděny výpočty v reportu nákladů
- Pole pro množství: Při zúčtování některých nákladů, např. při zúčtování spotřeby na pohonné hmoty do vlastní spotřeby, se do SAPu nezadáva pouze údaj o finanční částce, ale i údaj o množství. Je-li toto pole označeno, tak v reportu nákladů je k dané skupině účtů zobrazen další sloupec, kde je uvedena suma za množství za danou skupinu účtů
- Definice účtů: Zde se definuje seznam účtů k dané skupině účtů.

Po kliku na tlačítko Uložit dojde k uložení skupiny účtů do databáze. Data skupin účtů je možno dále editovat nebo mazat.

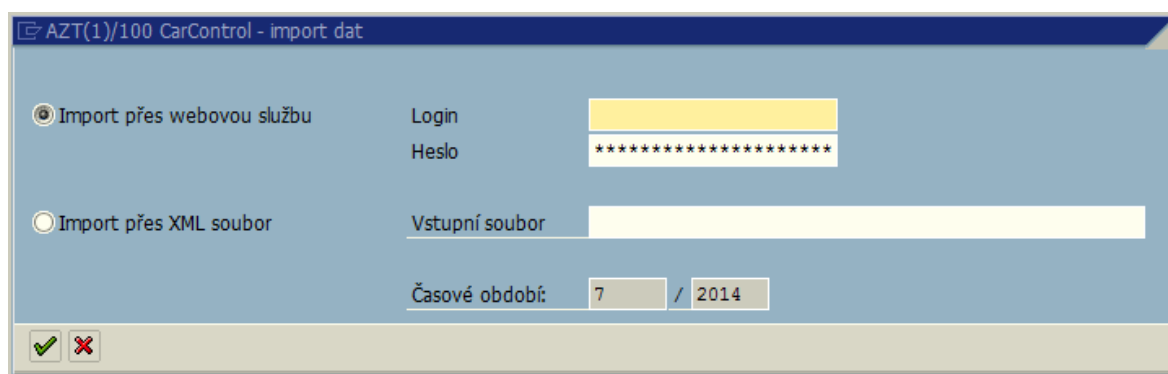
4.3.4 Import dat

Import dat je důležitou součástí administrace. Na straně CarControlu není sjednoceno, že všechny exporty lze realizovat přes webovou službu. Toto má za následek, že pro import některých dat je potřeba se přihlásit jako administrátor na portál CarControl a export provést ručně do formátu XML, který lze do aplikace importovat.

Možnosti importu jsou následující:

- Uživatelé systému: lze pouze přes XML soubor
- Vozidla systému: lze pouze přes XML soubor
- Provozní data: lze přes webovou službu i XML soubor
- Data o nákladech: lze pouze přes webovou službu

Při vlastním volání importu se uživateli zobrazí vstupní dialog pro spuštění importu.



Obr. 11. Transakce ZCAR - dialogové okno pro ověření k import dat

Tento dialog je totožný pro všechny druhy importů. Dialog je definován jako dynpro 1020, jehož vlastnosti se nastavují při jeho volání v *MODULE status_1020*, a to následovně:

- Pomocí metody *o_alv->get_alv()* je zjištěno, o jaký typ importu se jedná, a podle toho jsou nastaveny vlastnosti READ-ONLY na příslušných polích
- Jedná-li se o import provozních dat nebo nákladů, je potřeba definovat, za jaké časové období je potřeba import provést. V obou případech aplikace zjistí, jestli se nejedná o úplně první import provozních dat nebo nákladů za daný účetní okruh. Jestliže ano, pole pro definici měsíce a roku budou aktivní, a předvolí aktuální měsíc a rok. Uživatel má možnost tyto hodnoty změnit a určit tak měsíční období prvního importu jak za náklady, tak za provozní data. Po potvrzení volby dojde k importu. Při dalším importu budou již pole pro definici měsíce a roku READ-ONLY a

system nabídne vždy další měsíc a rok v pořadí, čili data jsou sehrávána chronologicky za sebou.

Logika importů je postavena tak, že v tabulce *zcar_imp_data* jsou data definována sumárně za SPZ a měsíc. Při exportu z CarControlu je ale parametr na plné datum od-do, kdy je možno provést export i za několik měsíců. Je-li proveden import přes webovou službu, aplikace zjistí, za jaký měsíc a rok je potřeba provést import. Na základě těchto dat si zjistí první a poslední den v měsíci daného roku a tyto parametry předá webové službě. Je tedy zabezpečeno, že se vrátí skutečně data, která byla požadována. Jedná-li se ale o import přes XML soubor, je definice dat plně v kompetenci uživatele a ten zodpovídá za to, že bude importovat XML soubor, jehož datové parametry odpovídají časovému období, které očekává import.

Po volbě typu importu a po definici přihlašovacích údajů do webové služby nebo importního XML souboru, dojde k volání `MODULE user_command_1020 INPUT`. Klikl-li uživatel na tlačítko `SAVE`, je volána příslušná metoda pro import.

4.3.4.1 Import přes XML soubor

Při importu přes XML soubor je nutno definovat vstupní soubor. Toto zabezpečuje metoda *cl_gui_frontend_services=>file_open_dialog*, která otevře standardní dialogové okno, kde uživatel vybere vstupní XML soubor. Je-li vše v pořádku, jméno souboru je uloženo v proměnné *s_input_file*.

V další fázi dojde voláním metod *gcl_xml->import_from_file* a *gcl_xml->render_2_xstring* k vlastnímu načtení a zpracování vstupních dat.

```
CALL METHOD gcl_xml->render_2_xstring
IMPORTING
    retcode = gv_subrc
    stream  = gv_xml_string
    size    = gv_size.

CALL METHOD gcl_xml->render_2_xstring
IMPORTING
    retcode = gv_subrc
    stream  = gv_xml_string
    size    = gv_size.
IF gv_subrc = 0.
```

Posledním funkčním modulem, který je volán, je modul *SMUM_XML_PARSE*, který provede rozparsování vstupního souboru dle formátu XML do interní tabulky *gt_xml_data*.

```
CALL FUNCTION 'SMUM_XML_PARSE'
EXPORTING
    xml_input = gv_xml_string
TABLES
    xml_table = gt_xml_data
return      = gt_return.
```

Proběhlo-li vše v pořádku, v proměnné *gt_xml_data* jsou data připravena pro vlastní import do SAPu.

4.3.4.2 Import přes webovou službu

Výhoda komunikace přes webovou službu spočívá v tom, že aplikace se ze SAPu přímo webovou službou předá vstupní parametry CarControlu, který na základě těchto parametrů vrátí požadovaná data.

Pro vlastní komunikaci je potřeba v SAPu na úrovni systému přes transakci SM59 definovat http spojení k externímu serveru. Toto spojení je nezávislé na reportu ZCAR, je použitelné v rámci systému pro různé aplikace, které by toto připojení chtěli využít. Pro případ reportu ZCAR bylo toto připojení vytvořeno následovně:

- Typ spojení: G
- Cílový počítač: carcontrol.cz.o2.com
- Prefix: /ssl/Export.asmx
- Číslo služby: 80

Po definici lze připojení otestovat tlačítkem Test Spojení. Aby bylo možno toto připojení v reportu používat, musí být v reportu definováno, což je provedeno následovně:

```
CONSTANTS: co_dest_o2 TYPE rfcdes-rfcdest VALUE 'CAR01'.
```

Pro vlastní volání webové služby je potřeba do proměnné *lv_qstring* vložit strukturu SOAP s parametry pro přístup včetně dat, za která chceme data získat.

Voláním metody *c1_http_client=>create_by_destination* je vytvořen http komunikační klient *l_client*, kterému jsou předávány vstupní parametry spojení.

```

CALL METHOD cl_http_client=>create_by_destination
EXPORTING
    destination                = co_dest_o2
IMPORTING
    client                     = l_client
l_client->request->set_cdata( data = link ).
l_client->request->set_header_field( name  = 'Content-Type'
                                   value = 'text/xml; charset=utf-8' ).

```

Metodou `l_client->send` dojde k odeslání dat na server. Je-li komunikace v pořádku, jsou data ze strany CarControlu přes metodu `l_client->get_data()` uložena do stringu `l_content`.

```

CALL METHOD l_client->send
EXPORTING
    timeout                    = 0
EXCEPTIONS
    http_communication_failure = 1
    http_invalid_state         = 2
    http_processing_failed     = 3
    http_invalid_timeout       = 4.

```

```
l_content = l_client->response->get_cdata( ).
```

Nakonec je opět volán modul `SMUM_XML_PARSE`, který provede rozparsování vstupního souboru dle formátu XML do interní tabulky `gt_xml_data`.

4.3.4.3 Zpracování importních dat

Jak v případě importu XML souboru, tak v případě webové služby, jsou vstupní data pro import připravena v interní tabulce `GT_XML_DATA`. Princip zpracování je totožný pro vozidla, uživatele i provozní data.

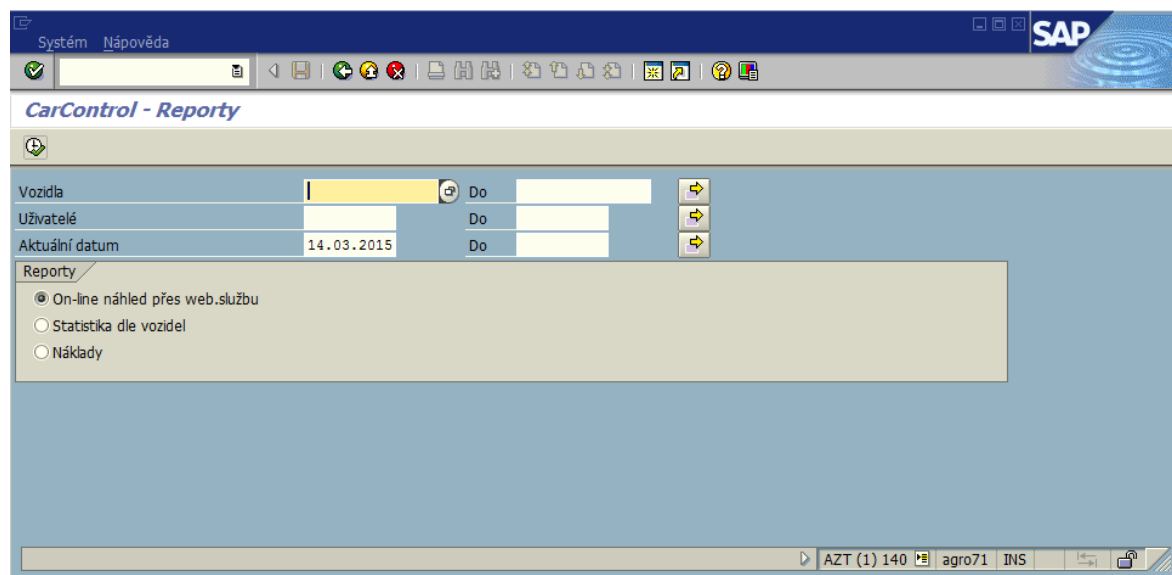
V případě uživatelů jsou jednotlivé řádky interní tabulky čteny v cyklu `LOOP – ENDLOOP` s tím, že je testován sloupec `<fs>-cname`. Jestliže název tohoto pole je totožný s názvem některého z polí struktury `LL_ZCAR_USERS`, je toto pole ve struktuře `LL_ZCAR_USERS` naplněno hodnotou `<fs>-cvalue`. Po naplnění všech polí je volána metoda `add_user()`, která v případě uživatelů provede vlastní insert nebo update dat do databáze.

Jelikož se testují veškeré vstupní pole, je v případě importu přes XML soubor zabezpečeno, že příslušný vstupní soubor má požadovanou strukturu a že dojde k načtení korektních dat.

Totožným způsobem jsou do SAPu ukládána data za vozidla, provozní data i data nákladů.

5 REPORTY

Nedílnou součástí všech aplikací jsou reporty, které data zobrazují a prezentují. Jelikož aplikace ZCAR je nová, má v současnosti 3 reporty, ale je počítáno s vytvářením dalších, dle požadavků z praxe.



Obr. 12. Transakce ZCAR – menu pro výběr reportů

Při spuštění reportů jsou uživateli zobrazeny vstupní parametry a seznam dostupných reportů. Výstupy z reportů se dají omezit na:

- Vozidla: výběr dle SPZ
- Uživatelé: výběr dle ID uživatele
- Aktuální datum: report za dané datum nebo datum od-do

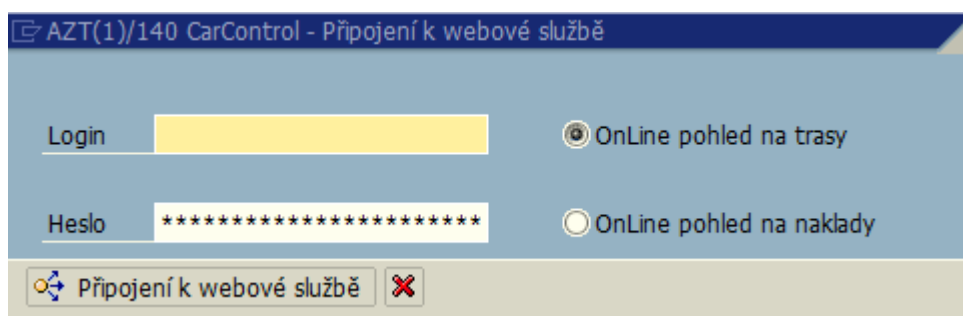
Reporty jsou psány standardně jako běžné reporty v SAPu, tedy obsahují:

- Deklarace vstupních dat: parametry, struktury, int. tabulky
- Výběr dat: dle typu reportu probíhá výběr a zpracování vstupních dat, výpočty nad interními tabulkami, apod.
- Zobrazení: je realizováno technologií ALV, včetně definice příslušných FIELD katalogů.

5.1.1 Report On-line náhled přes webovou službu

Z důvodu bezpečnosti v tabulkách SAPu nejsou uloženy detailní informace o využívání referenčních vozidel jednotlivými zaměstnanci, kde by byly vidět jednotlivé jízdy včetně informací o místech výjezdů a dojezdů včetně časů. Pouze se evidují údaje o najetých kilometrech, které se dále používají v reportu nákladů. Je-li potřeba získat detailní data o provozu, je možno využít reportů pro on-line náhled. Tato přes webovou službu získaná data nejsou nikde ukládána, jsou pouze zobrazována v relaci daného uživatele v SAPu

Při spuštění reportu je zobrazen dialog, kde je potřeba definovat přihlašovací jméno a heslo do CarControlu a dále zvolit, je-li potřeba report provozu nebo nákladu.



Obr. 13. Transakce ZCAR - ověřovací dialog pro on-line čtení dat

Po zvolení vstupních dat a volbě druhu reportu dojde k volání metody `g_zcar_import->go_to_web(lv_qstring)`, kde v proměnné `lv_qstring` je definován vstupní WSDL parametr. Je-li připojení v pořádku, metoda vrátí proměnnou `xml_data`, ve které jsou uložena požadovaná data. Posledním krokem je zpracování těchto dat a jejich uložení do jednotlivých polí interní tabulky `LT_REP_ONLINE`. Nakonec pomocí funkčního modulu `REUSE_ALV_GRID_DISPLAY` je zobrazen vlastní ALV grid s daty.

```
CALL FUNCTION 'REUSE_ALV_GRID_DISPLAY'
  EXPORTING
    is_layout          = ls_layout
    it_fieldcat         = lt_fieldcat
    i_callback_user_command = 'USER_COMMAND'

  TABLES
    t_outtab           = lt_rep_online.
```

SPZ	Datum výjezd	Datum příjezd	Místa	Účel	Řidič	Délka osob	Délka fre	Suma KM
4Z59583	2015-03-01T16:17:17+01:00	2015-03-01T17:01:12+01:00	[CZ] Hovorany, síh. II. tř. 380 > [CZ] Hovorany	-	Zlámal Radim	35,290	0,000	35
4Z59583	2015-03-02T06:59:11+01:00	2015-03-02T07:27:22+01:00	[CZ] Hovorany > [CZ] Hustopeče, síh. III. tř. 4203	Služební jízda		0,000	28,530	29
4Z59583	2015-03-02T09:48:32+01:00	2015-03-02T09:51:29+01:00	[CZ] Hustopeče, síh. III. tř. 4203 > [CZ] Hustopeče, síh. III. tř. 4203	Služební jízda		0,000	0,110	0
4Z59583	2015-03-02T14:49:37+01:00	2015-03-02T15:17:40+01:00	[CZ] Hustopeče, síh. III. tř. 4203 > [CZ] Hovorany	Služební jízda		0,000	28,300	28
4Z59583	2015-03-03T05:04:47+01:00	2015-03-03T06:06:48+01:00	[CZ] Hovorany > [CZ] Kroměříž, Čelakovského	Služební jízda		0,000	55,500	56
4Z59583	2015-03-03T11:19:42+01:00	2015-03-03T11:23:07+01:00	[CZ] Kroměříž, Blánská > [CZ] Kroměříž, Čelakovského	Služební jízda		0,000	1,000	1
4Z59583	2015-03-03T14:34:31+01:00	2015-03-03T15:41:23+01:00	[CZ] Kroměříž, Hulinská > [CZ] Hovorany	Služební jízda		0,000	59,300	59
4Z59583	2015-03-04T05:11:56+01:00	2015-03-04T06:12:10+01:00	[CZ] Hovorany > [CZ] Kroměříž, Čelakovského	Služební jízda		0,000	57,500	58
4Z59583	2015-03-04T11:19:38+01:00	2015-03-04T11:22:27+01:00	[CZ] Kroměříž, Čelakovského > [CZ] Kroměříž, Čelakovského	Služební jízda		0,000	0,300	0
4Z59583	2015-03-04T11:35:25+01:00	2015-03-04T11:56:58+01:00	[CZ] Kroměříž, Čelakovského > [CZ] Přerov, Polní	Služební jízda		0,000	21,600	22
4Z59583	2015-03-04T14:27:17+01:00	2015-03-04T14:50:54+01:00	[CZ] Přerov, Polní > [CZ] Kroměříž, Čelakovského	Služební jízda		0,000	19,300	19
4Z59583	2015-03-04T15:03:39+01:00	2015-03-04T16:13:51+01:00	[CZ] Kroměříž, Čelakovského > [CZ] Hovorany	Služební jízda		0,000	52,000	52
4Z59583	2015-03-05T06:45:33+01:00	2015-03-05T07:04:00+01:00	[CZ] Hovorany > [CZ] Ratíškovice, Za mýnem	Služební jízda		0,000	14,600	15
4Z59583	2015-03-05T07:52:07+01:00	2015-03-05T08:34:00+01:00	[CZ] Ratíškovice, Za mýnem > [CZ] Hustopeče, Janáčkova	Služební jízda		0,000	49,200	49
4Z59583	2015-03-05T08:39:18+01:00	2015-03-05T08:44:36+01:00	[CZ] Hustopeče, Janáčkova > [CZ] Hustopeče, síh. III. tř. 4203	Služební jízda		0,000	4,000	4
4Z59583	2015-03-05T13:46:04+01:00	2015-03-05T14:11:30+01:00	[CZ] Hustopeče, síh. III. tř. 4203 > [CZ] Čejč, síh. II. tř. 380	Služební jízda		0,000	24,600	25

Obr.14. Transakce ZCAR - report on-line náhled

5.1.2 Report Statistika dle vozidel

Report Statistika dle vozidel slouží pro zobrazení statistických dat o vozidlech.

Je definována struktura *ls_rep1*, nad kterou je implementována interní tabulka *lt_rep1* a pracovní oblastí *ll_rep1*.

Jelikož jsou všechna data uložena v interních tabulkách, není potřeba žádný select do databáze. V první fázi je provedeno seřazení dat v interních tabulkách dle vyhledávacích klíčů.

```
SORT g_zcar_zak->lt_zcar_zak BY jednotka_id.
```

```
SORT g_zcar_users->lt_zcar_users BY user_id.
```

V další fázi dochází ve smyčce LOOP.. ENDLOOP ke čtení interní tabulky *LT_ZCAR_CARS* s daty vozidel a uložení dat do pracovní oblasti *ll_rep1*.

Dále probíhá čtení tabulky zakázek *LT_ZCAR_ZAK*. Jestliže v této tabulce existuje záznam s totožným polem *jednotka_id*, znamená to, že vozidlo má definovanou controllingovou zakázku a tato je doplněna do struktury *ll_rep1*.

V dalším kroku je čtena tabulka *LT_ZCAR_USERS*. Existuje-li záznam s totožným *user_id*, jsou příslušná data doplněna do struktury *ll_rep1*.

Příkazem APPEND je struktura *ll_rep1* vložena do interní tabulky *lt_rep1*.

Pro vlastní zobrazení dat reportu je opět zabezpečeno voláním funkčního modulu *REUSE_ALV_GRID_DISPLAY*, kterému je jako vstupní parametr předána interní tabulka *lt_rep1*.

Jednot.	Uživatel	Označení	VIN	Druh vozidla	Spz	Zakázka	Výrob.	Značka	Model	Palivo	Spotřeba	Velkost n	Tachometr	Složka	Jměr
16524	27988	Hrabal Pavel	TMBGK26YX74048822	Osobní vozidlo	2272528	97105210954	-	Škoda Fabia	Škoda Fabia	Nafta	4,90	45,00	189.093,900	6269	Pave
31066	22975	Ležák Josef	TMBEH61Z882022682	Osobní vozidlo	3211564	97105210392	-	Škoda Octavia R	Škoda Octavia R	Nafta	6,00	55,00	227.654,900	6269	Josef
31101	35305	Šustr Jiří	TMBAE61Z392048373	Osobní vozidlo	3278311	97104293766	-	ŠKODA Octavia	Škoda Octavia	Nafta	6,27	55,00	116.447,000	6243	Jiří
40961	56815	Bužtěk Petr	TMBGT61Z6C2103003	Osobní vozidlo	4259967	97104259967	-	ŠKODA Octavia 2	Octavia 2	Nafta	4,70	55,00	123.705,600	6270	Petr
37923	58679	Sterzák Daniel	TMBNU26Y774119419	Osobní vozidlo	4259718	97104259718	-	Škoda Fabia	Škoda Fabia	Nafta	4,90	45,00	213.166,100	6246	Dani
37931	56876	Zlámal Radim	TMBBE612682070670	Osobní vozidlo	4259583	97104259583	-	ŠKODA Octavia 2	ŠKODA Octavia 2	Nafta	6,00	55,00	155.272,300	6270	Radim
37910	56841	Kotulek Miloš	TMBCS21Z282111102	Osobní vozidlo	4253541	97104253541	-	Škoda Octavia	Octavia	Nafta	6,00	55,00	142.706,400	6244	Miloš
37922	56854	Otek Jaroslav	TMBCS21Z182111043	Osobní vozidlo	4253526	97104253526	-	Škoda Octavia	Škoda Octavia	Nafta	6,00	55,00	208.004,000	6247	Jaros
37920	56868	Široký Jiří	TMBCS46Y523387593	Osobní vozidlo	4253458	97104253458	-	Škoda Fabia	Škoda Fabia	Nafta	5,50	45,00	221.339,800	6258	Jiří
31004	55590	demontovaná	VF18807CF267274404	Osobní vozidlo	2248121	97104253279	-	Renault Clio	Renault Clio	Nafta	4,47	50,00	112.696,000	3946	0
17449	49230	Řehulová Jitka	TMBGJ26Y774160531	Osobní vozidlo	3258662	97103258662	-	Škoda Fabia 1,4	Škoda Fabia 1,4	Nafta	4,90	45,00	164.959,800	6268	Jitka
31019	56830	Gryc Jaroslav	TMBGK26Y474011474	Osobní vozidlo	2264421	97102264421	-	Škoda Fabia	Škoda Fabia	Nafta	4,90	50,00	184.037,360	6246	Jaros
17201	55590	Karel Hloch	1Z08874	Nákladní vozidlo	1208874	97101208874	-	Avia	-	Nafta	20,00	120,00	55.487,400	3946	0
37911	68223	Haisler Zdeněk	TMBMG25J175038232	Osobní vozidlo	4259458	*9458	-	Škoda Roomster	Roomster	Nafta	6,00	55,00	186.877,100	6257	Zden
16450	55590	Zalabák Josef	0	Nákladní vozidlo	0	-	-	MAN	28430	Nafta	46,00	340,00	198.318,600	3946	0
16453	55590	Šustek Břetislav	0	Nákladní vozidlo	0	-	-	MAN	28430	Nafta	46,00	340,00	202.374,500	3946	0
16466	22874	Palčka Pavel	WMAT37ZZZ1M316726	Nákladní vozidlo	2248104	-	-	MAN 26.403	MAN 26403	Nafta	46,00	360,00	806.151,700	6269	Pave
16491	55590	prodáno	1000100000197	Nákladní vozidlo	KMA0604	-	-	LIAZ	-	Nafta	44,00	340,00	266.438,600	3946	0
16522	55590	Řidič Podvín	TMBGK26Y974027573	Osobní vozidlo	2264576	-	-	Škoda Fabia com	Škoda Fabia com	Nafta	4,90	55,00	260.527,300	3946	0
16532	23416	Pospíšil Michal	ZCFA1ED1302504614	Nákladní vozidlo	2290513	-	-	Iveco Eurocargo	Iveco Eurocargo	Nafta	20,00	200,00	212.791,500	6269	Mich
16534	35206	Večkovský Pavel	WFOVXX1TFV6S32171	Nákladní vozidlo	2257622	-	-	Ford Transit	Ford Transit	Nafta	7,27	80,00	145.842,800	6269	Pave
16542	55590	Polydor Petr	0	Osobní vozidlo	0	-	-	Škoda	Roomster	Nafta	6,00	50,00	154.569,100	3946	0
16838	55590	Hegedus Stefan	0	Nákladní vozidlo	0	-	-	LIAZ	-	Nafta	44,00	170,00	358.059,700	3946	0
16840	55590	Kašpařík Pavel	0	Nákladní vozidlo	0	-	-	MAN	26430	Nafta	45,00	380,00	271.963,200	3946	0
16858	55590	VKS Zábřeh	2257156	Nákladní vozidlo	2257156	-	-	LIAZ	-	Nafta	45,00	170,00	336.662,200	3946	0
16859	55590	Adamec Svatoslav	0	Nákladní vozidlo	0	-	-	MAN	26430	Nafta	45,00	380,00	184.108,400	3946	0
16879	55590	Iveco ML	ZCFA1ED1302518469	Nákladní vozidlo	3225807	-	-	Iveco Eurocargo	Eurocargo	SMN 30	20,00	120,00	186.143,300	3946	0
16882	22685	Vrána Jiří	WMAT32W630M253201	Nákladní vozidlo	2248156	-	-	MAN 18.463	MAN 18.463	Nafta	46,00	400,00	822.503,000	6269	Jiří
16887	22697	MAN 26.403 sklápěč	WMAT37ZZZYV03338	Nákladní vozidlo	2248157	-	-	MAN 26.403	MAN 26.403	SMN 30	46,00	500,00	1.013.909,200	6243	Rom

Obr. 15. Transakce ZCAR - report Statistika dle vozidel

5.1.3 Report Náklady

Report nákladů je stěžejní částí celé aplikace, protože provádí vlastní zpracování a vyhodnocování dat. Celý proces generování reportu je složen z několika cyklů, a to:

- Definice vstupních dat
- Výběr dat z tabulek SAPu
- Vytvoření struktury a načtení dat
- Dle počtu namapovaných skupin účtů dynamické vytvoření příslušného počtu sloupců do reportu včetně dopočtu hodnot

Při tvorbě reportu je brán zřetel na to, aby nedocházelo k zbytečnému zatížení systému, proto množina zpracovávaných dat je omezena na vstupní parametry, vždy se tedy pracuje jen s těmi controllingovými zakázkami, které jsou přiřazeny příslušným SPZ za zvolený časový interval.

Vlastní data jsou uložena v interní tabulce *lt_rep2*, pracovní oblast na tabulku je označena jako *ll_rep2*. V první fázi je definován do interní tabulky *lt_zakazka* seznam controllingových zakázek, nad kterými bude report proveden. Prvním vstupním parametrem je seznam SPZ uložen v parametru *g_cars*. Jestli žádné SPZ není definováno, *g_cars* je prázdné a

v tom případě do *lt_zakazka* jsou vloženy všechny definované controllingové zakázky. Je-li definován seznam SPZ, systém zjistí, jestli daná SPZ má definovanou controllingovou zakázku. Jestliže ano, vloží tuto zakázku do *lt_zakazka*.

Druhým vstupním parametrem je datum, za které má být report vytvořen. Může být definováno pouze jedno datum nebo datum od-do. Systém vždy dle volby data určí první a poslední den pro vytvoření reportu. Tento interval vloží do proměnné *lt_datum*.

5.1.3.1 Výběr dat z tabulek SAPu

Ze SAPu je potřeba načíst controllingová data, provozní a nákladová data z CarControlu. Controllingová data jsou ze SAPu čtena z tabulky GLPCA a jsou uložena do interní tabulky *lt_glpca*.

Vlastní select je omezen pouze na zakázky a datum definované v předešlém kroku.

Druhý select je prováděn nad tabulkou *ZCAR_IMP_DATA*, ve které jsou uložena provozní a nákladová data importovaná z CarControlu. Tato data jsou importována dle SPZ, proto select je omezen na proměnnou *g_cars*, ve které je tento seznam SPZ definován na vstupu. Je-li *g_cars* prázdné, select čte data za všechna vozidla. Dále je také select omezen datem definovaným na vstupu. Výsledná data jsou připravena pro generování vlastního reportu v interní tabulce *LT_ZCAR_IMP_DATA*.

5.1.3.2 Vytvoření struktury a načtení dat

Report o nákladech je měsíční, tedy pro každou SPZ, resp. controllingovou zakázku je v závislosti na vstupním časovém intervalu pro každou položku pro každý měsíc vytvořen záznam, na který jsou dopočítávány další údaje.

Vlastní tvorba struktury pro report probíhá v cyklu WHILE. Proměnné *mesic* a *rok* si nejsou časové podmínky, za které budou pro každou zakázku vytvořené záznamy pro vlastní zobrazení.

V rámci každého kroku cyklu je v LOOPu je plněna struktura *ll_rep2*. V první fázi jsou definována pole, která jsou vždy totožná pro danou controllingovou zakázku, nezávisle na měsíci a roku. Jedná se o pole:

- AUFNR: číslo controllingové zakázky
- POPER: měsíc
- RYEAR: rok

V další fázi příkazem jsou READ TABLE do struktury *LL_ZCAR_CARS* doimportována data vozidla, které má přiřazenu příslušnou controllingovou zakázku v poli AUFNR. Data z této struktury jsou přidána také do *ll_rep2*:

- User_id: ID uživatele vozidla
- Jednotka_id: ID jednotky ve vozidle
- Name: jméno řidiče
- Palivo: druh paliva vozidla
- Nadrz: velikost nádrže
- SPZ: spz vozidla

Dalším krokem je načtení provozních a nákladových dat importovaných z CarControlu. Příkazem READ TABLE je čten z interní tabulky *LL_ZCAR_IMP_DATA* záznam, který má totožné SPZ, rok a měsíc. Podmínka *TYP = 0* znamená, že budou čteny jen provozní data. Po nalezení záznamu jsou doplněny do *ll_rep2* další data:

- DELKA_OSOBNI: počet ujetých kilometrů soukromě za měsíc
- DELKA_FIREMNI: počet ujetých kilometrů firemně za měsíc
- DELKA_SUMA: počet ujetých kilometrů celkem
- DELKA_OSOBNI: počet ujetých kilometrů soukromě za měsíc
- TACHOMETR: konečný stav tachometru za měsíc

Pro čtení provozních dat byl použit příkaz READ TABLE, protože provozní data jsou v SAPu uložena za měsíc pouze jedním záznamem, který lze dle SPZ, měsíce a roku jednoznačně identifikovat. Data nákladů jsou ovšem za měsíc uložena položkově, navíc jsou ještě dělena na náklady za tankování a ostatní náklady, čili v tomto případě READ TABLE nelze použít, data musí být čtena v cyklu LOOP AT. Záznam je opět testován na SPZ, rok a měsíc *TYP = 1*, znamená, že se jedná o náklad. Jestliže pole *tankovani* je rovno hodnotě A, jedná se o náklad za tankování. Je-li takovýto záznam nalezen, jsou za daný měsíc inkrementovány pole:

- LITRY: počet načerpaných litrů za měsíc
- CENA_PALIVO: cena za palivo za měsíc
- CENA_ZAKLAD: základ za palivo za měsíc
- CENA_CELKEM: cena za palivo za měsíc včetně DPH

Jestliže pole *tankovani* není rovno hodnotě A, nejedná se o náklad za tankování, ale o náklad typu pořízení oleje, mytí, apod. V tom případě je za daný měsíc inkrementováno pole:

- DALŠÍ_NAKLADY: cena za další náklady za měsíc

V tomto okamžiku je struktura *ll_rep2* naplněna a příkazem APPEND je přidána do interní tabulky *lt_rep2*. Je ukončen cyklus příkazem ENDLOOP, čili za aktuální měsíc jsou vytvořeny záznamy pro všechny controllingové zakázky, může tedy pokračovat hlavní cyklus WHILE, kdy se budou zpracovávat opět všechny controllingové zakázky pro následující měsíc.

Po ukončení cyklu WHILE je vytvořena kompletní struktura dat pro zobrazení reportu, v níž jsou dotažené data z CarConrolu. Posledním krokem je dopočet dat z controllingových tabulek ze SAPu.

5.1.3.3 Dotažení dat z controllingových zakázek ze SAPu

Počet zobrazených sloupců je závislý na počtu mapovaných skupin účtů. Protože report počítá s tím, že počet těchto sloupců může být libovolný, je interní tabulka *lt_rep2* převedena na dynamickou a dále je zpracovává pomocí FIELD symbolů.

Report prochází v LOOPu interní tabulku *g_zcar_coast->lt_zcar_coast*, kde jsou mapovány skupiny účtů za příslušný účetní okruh. Pro každou skupinu účtů je dynamicky vytvořen sloupec, jehož pole *wa_comp_fld-name* je roven klíči v položce skupiny účtů. Dále je definován typ pole a případem APPEND je sloupec dynamicky přidán do interní tabulky *i_comp_tot_tab*. Po vytvoření požadované struktury jsou do dynamické interní tabulky doplněny data z *lt_rep2*.

V posledním kroku dochází k přičítání dat z controllingových zakázek ze SAPu. V LOOPu je procházena interní tabulka *lt_glpca* s načtenými daty controllingových zakázek a dále je čtena nově vytvořená dynamická tabulka *<fs_it_output>*. V okamžiku nalezení záznamu s totožným číslem controllingové zakázky, měsícem a rokem, dojde k testu, jestli účet definovaný v controllingové zakázce je ve shodě s účty v některé z mapovaných skupin účtů. Jestliže ano, je ze skupiny účtů zjištěn klíč, který je totožný s názvem z některých dynamicky vytvořených sloupců. Dle tohoto názvu je nalezeno příslušné pole v *<fs_it_output>* a do něj je přičten příslušný záznam.

V tomto okamžiku je vytvořen report nákladů a je připraven k zobrazení.

ZÁVĚR

Cílem mé bakalářské práce bylo zjednodušit formu vykazování ekonomických ukazatelů na provoz firemních vozidel. Velké ulehčení práce znamená pro pracovníka odpovědného za výkaznictví. Ten nemusí exportovat ručně data z CarControlu, dále ze SAPu, a pak tato data ručně spojovat a vyhodnocovat. Díky reportu ZCAR má tato data okamžitě k dispozici. Vypovídající hodnota reportu je omezena na data, která jsme schopni z CarControlu získat. Cíl projektu není v tom nahradit funkcionalitu CarControlu, ale propojit nabízená data a ta dále pro potřeby firmy automaticky zpracovávat.

Při tvorbě reportu jsem se snažil plně využívat možnosti, které SAP a jazyk ABAP nabízí včetně implementace funkcí pro zpracování dat z externích systémů pomocí webových služeb.

Celá aplikace byla v průběhu vývoje provozována v testovacím systému SAP AZT na dvou testovacích účetních okruzích včetně propojení na účetní data ze SAPu, takže veškeré postupy byly řádně testovány a prověřeny.

V současné době je aplikace provozována na ostrém systému SAP AZP. Ostrý provoz ovšem netrvá tak dlouho, aby bylo možno objektivně zhodnotit přínos aplikace pro celý systém AZP, protože ostatní účetní okruhy se s aplikací postupně seznamují. Firma Navos, a.s. již s aplikací pracuje, kdy jsou za daný účetní okruh naimportována provozní a nákladová data do aktuálního období a jsou definovány jednotlivé nákladové účty. Na základě těchto dat probíhají první analýzy nákladů na provoz, kdy pomocí reportu Náklady manažer dopravy získal měsíční informace o provozu vozidel a jejich nákladech. Na základě těchto informací již ve firmě Navos, a.s. proběhla reorganizace dodávkových vozidel, kdy dle vytíženosti, počtu najetých kilometrů za měsíc a nákladů na vozidlo, došlo k přesunu vozidel mezi středisky. Přínos aplikace ZCAR vidím v tom, že tato data manažer dopravy získal okamžitě ze systému SAP ve formě jednotného reportu, nemusel tedy data ručně dopočítávat a vyhodnocovat.

Jelikož téma dopravy a nákladů na dopravu je v naší firmě aktuální, je možný další vývoj aplikace včetně dalších požadavků a připomínek, které můžou přijít z praxe. Osobně vidím možnost dalšího vývoje v účetním zpracování dat, kdy v reportu ZCAR jsou data o tanko-

vání za aktuální měsíc na firemních čerpacích stanicích. Tato data by se dala využít např. pro zautomatizování vytváření účetních dokladů, které odepisují danou spotřebu ze skladových zásob, apod.

SEZNAM POUŽITÉ LITERATURY

- [1] MAASSEN, A., SCHOENEN, M., GADATSCH. A. SAP R/3: kompletní průvodce. Computer Press, 2007. ISBN 978-80-251-1750-7
- [2] KUHNHAUSER, K. ABAP: výukový kurz. Computer Press, 2009. ISBN 978-80-251-2117-7
- [3] GULSEN, A. ABAP Workbench -100 things you should know about. Galileo Press, 2012. ISBN 978-1-59229-427-5
- [4] ERL, T. SOA, servisně orientovaná architektura: kompletní průvodce. Computer Press, 2009. ISBN 978-80-251-1886-3
- [5] SAP Help Portal <http://help.sap.com>
- [6] KCT DATA, authorized SAP professionals – podklady ze školení ABAP

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SAP R/3 Systém, Applications and Products.

ABAP Advanced Business Application Programming.

DDIC Data Dictionary

ZCAR Transakce ZCAR

SEZNAM OBRÁZKŮ

<i>Obr.1. Úvodní obrazovka webového portálu CarControl</i>	<i>14</i>
<i>Obr. 2. Kniha jízd ve webovém portálu CarControl.....</i>	<i>15</i>
<i>Obr. 3. SAP report z transakce KOB1 – Jednotlivé položky skutečných nákladů za zakázky.....</i>	<i>18</i>
<i>Obr. 4. Transakce ZCAR - okno administrace.....</i>	<i>29</i>
<i>Obr. 5. Vazby mezi databázovými tabulkami transakce ZCAR</i>	<i>34</i>
<i>Obr. 6. Okno vývojového prostředí ABAP DEVELOPMENT WORKBENCH v SAP</i>	<i>35</i>
<i>Obr. 7. Transakce ZCAR - úvodní okno reportu.....</i>	<i>38</i>
<i>Obr. 8. Transakce ZCAR - okno importu provozních dat</i>	<i>39</i>
<i>Obr. 9. Transakce ZCAR - přiřazení controllingové zakázky k vozidlu</i>	<i>41</i>
<i>Obr. 10. Transakce ZCAR - mapování účtů.....</i>	<i>43</i>
<i>Obr. 11. Transakce ZCAR - dialogové okno pro ověření k import dat.....</i>	<i>44</i>
<i>Obr. 12. Transakce ZCAR – menu pro výběr reportů</i>	<i>49</i>
<i>Obr. 13. Transakce ZCAR - ověřovací dialog pro on-line čtení dat</i>	<i>50</i>
<i>Obr.14. Transakce ZCAR - report on-line náhled</i>	<i>51</i>
<i>Obr. 15. Transakce ZCAR - report Statistika dle vozidel</i>	<i>52</i>

SEZNAM TABULEK

<i>Tab. 1. ZCAR_USERS: uživatelé systému CarControl</i>	30
<i>Tab. 2. ZCAR_CARS: vozidla v systému CarControl</i>	30
<i>Tab. 3. ZCAR_ZAK: vazba mezi vozidlem v CarControlu a controllingovou tabulkou v SAPu</i>	31
<i>Tab. 4. ZCAR_IMP_INDEX: časový index pro import provozních dat</i>	31
<i>Tab. 5. ZCAR_IMP_DATA: importovaná data o provozu a nákladech vozidel</i>	31
<i>Tab. 6. ZCAR_IMP_UCTY: tabulka pro mapování účtů</i>	32

SEZNAM PŘÍLOH

- P I Události reportu
- P II Metoda DISPLAY_ALV
- P III Modul USER_COMMAND pro obsluhu propojení vozidlo-SAP
- P IV Modul USER_COMMAND pro obsluhu volání importní metody
- P V Volání XML souboru pro import dat
- P VI SOAP struktura pro webovou službu
- P VII Zpracování importních dat
- P VIII Report statistika dle vozidel
- P IX Report Náklady
- P X Zobrazení reportu nákladů

PŘÍLOHA P I: UDÁLOSTI REPORTU

```
PARAMETERS: g_bukrs TYPE t001-bukrs OBLIGATORY.
SELECTION-SCREEN BEGIN OF BLOCK vyber1 WITH FRAME TITLE text-s08.
PARAMETERS: r1 RADIOBUTTON GROUP rad1,
             r5 RADIOBUTTON GROUP rad1.
SELECTION-SCREEN END OF BLOCK vyber1.
DATA o_alv TYPE REF TO lcl_alv.
TABLES: makt.
```

```
START-OF-SELECTION.
    gt_ucokruh = g_bukrs.
    CREATE OBJECT g_zcar_zak.
    CREATE OBJECT g_zcar_users.
    CREATE OBJECT g_zcar_cars.
    CREATE OBJECT g_zcar_import.
    IF r1 EQ 'X'.
        PERFORM administrace.
    ENDIF.
    IF r5 EQ 'X'.
        PERFORM reporty.
    ENDIF.
```

PŘÍLOHA P II: METODA DISPLAY_ALV

```
IF id_view = 0. "USER
    status_pf = 0.
    cl_salv_table=>factory(
        IMPORTING
            r_salv_table = o_table
        CHANGING
            t_table      = g_zcar_users->lt_zcar_users ).
ENDIF.

IF id_view = 1. "CAR
    status_pf = 0.
    cl_salv_table=>factory(
        IMPORTING
            r_salv_table = o_table
        CHANGING
            t_table      = g_zcar_cars->lt_zcar_cars  ).
ENDIF.

IF id_view = 2. "IMPORT
    status_pf = 0.
    cl_salv_table=>factory(
        IMPORTING
            r_salv_table = o_table
        CHANGING
            t_table      = g_zcar_import->lt_zcar_import ).
ENDIF.

    IF id_view = 3. "provozni data
        status_pf = 1.
        cl_salv_table=>factory(
            IMPORTING
                r_salv_table = o_table
            CHANGING
                t_table      = g_zcar_import->lt_zcar_imp_data ).
        set_alv( 2 ).
    ENDIF.
```


PŘÍLOHA P III: MODUL USER_COMMAND PRO OBSLUHU PROPOJENÍ VOZIDLO-SAP

```
MODULE user_command_1010 INPUT.
  DATA l_state TYPE i.
  CASE ok_code1.
    WHEN 'SAVE'.
      DATA pom_aufnr LIKE coas-aufnr.
      pom_aufnr = s_zakazka_id.
      IF g_zcar_zak->check_exist( s_jednotka_id ) = 0.
        l_state = g_zcar_zak->save_data(
          id = s_jednotka_id key = pom_aufnr type = 0 ).
      ELSE.
        l_state = g_zcar_zak->save_data(
          id = s_jednotka_id key = pom_aufnr type = 1 ).
      ENDIF.
      READ TABLE g_zcar_cars->lt_zcar_cars INDEX o_alv->row_id
        INTO g_zcar_cars->ll_zcar_cars.
      g_zcar_cars->ll_zcar_cars-aufnr = s_zakazka_id.
      MODIFY g_zcar_cars->lt_zcar_cars FROM g_zcar_cars-
        >ll_zcar_cars INDEX o_alv->row_id.
      LEAVE TO SCREEN 0.
    WHEN 'EXIT'.
      LEAVE TO SCREEN 0.
    ENDCASE.
ENDMODULE.
```

PŘÍLOHA P IV: MODUL USER_COMMAND PRO OBSLUHU VOLÁNÍ IMPORTNÍ MEDODY

```
MODULE user_command_1020 INPUT.  
  CASE ok_code1.  
    WHEN 'SAVE'.  
      CASE l_ind.  
        WHEN 0.  
          CALL METHOD g_zcar_users->import_users( ).  
        WHEN 1.  
          CALL METHOD g_zcar_cars->import_cars( ).  
        WHEN 2.  
          CALL METHOD g_zcar_import->save_import( ).  
        WHEN 6.  
          CALL METHOD g_zcar_import->save_import( ).  
      ENDCASE.  
    LEAVE TO SCREEN 0.  
  WHEN 'EXIT'.  
    LEAVE TO SCREEN 0.  
  ENDCASE.  
ENDMODULE.
```

PŘÍLOHA P V: VOLÁNÍ XML SOUBORU PRO IMPORT DAT

```
DATA: file0 TYPE filetable,
      l_rc0 TYPE i.
CLEAR file0.
CALL METHOD cl_gui_frontend_services=>file_open_dialog
EXPORTING
    window_title          = 'c_title'
    default_extension     = 'xml'
    initial_directory     = 'c:'
    multiselection        = abap_false
CHANGING
    file_table            = file0
    rc                    = l_rc0
EXCEPTIONS
    file_open_dialog_failed = 1
    cntl_error              = 2
    error_no_gui            = 3
    not_supported_by_gui    = 4
    OTHERS                  = 5.
READ TABLE file0 INTO s_input_file INDEX 1.
```

PŘÍLOHA P VI: SOAP STRUKTURA PRO WEBOVOU SLUŽBU

CONCATENATE

```
'<?xml version="1.0" encoding="utf-8"?>'
'<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instan-
ce" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schem
as.xmlsoap.org/soap/envelope/">'
  '<soap:Body>'
    '<GenerateExport xmlns="http://tempuri.org/">'
      '<login>' s_login '</login>'
      '<password>' s_password '</password>'
      '<language>cs</language>'
      '<exportID>export_kniha_jizd_naklady</exportID>'
      '<paramKeys>'
        '<string>datum_od</string>'
        '<string>datum_do</string>'
      '</paramKeys>'
      '<paramValues>'
        '<string>' dat_od '</string>'
        '<string>' dat_do '</string>'
      '</paramValues>'
    '</GenerateExport>'
  '</soap:Body>'
'</soap:Envelope>'
INTO lv_qstring.
```

PŘÍLOHA P VII: ZPRACOVÁNÍ IMPORTNÍCH DAT

```
CLEAR ll_zcar_users.  
FIELD-SYMBOLS:<fs> TYPE smum_xmltb.  
LOOP AT gt_xml_data ASSIGNING <fs>.  
    CASE <fs>-cname.  
        WHEN 'ID'.  
            ll_zcar_users-user_id = <fs>-cvalue.  
        WHEN 'SlozkaID'.  
            ll_zcar_users-folder_id = <fs>-cvalue.  
        WHEN 'VychoziVozidloID'.  
            ll_zcar_users-defaultcar_id = <fs>-cvalue.  
        WHEN 'Jmeno'.  
            ll_zcar_users-name_first = <fs>-cvalue.  
        WHEN 'Prijmeni'.  
            ll_zcar_users-name_last = <fs>-cvalue.  
        WHEN 'Email'.  
            ll_zcar_users-smtp_addr = <fs>-cvalue.  
        WHEN 'PrihlasovaciJmeno'.  
            ll_zcar_users-login = <fs>-cvalue.  
        WHEN 'Uzivatel'.  
            ll_zcar_users-bukrs = gt_ucokruh.  
            l_status = me->add_users( ).  
    ENDCASE.  
ENDLOOP.
```

PŘÍLOHA P VIII: REPORT STATISTIKA DLE VOZIDEL

```
DATA: BEGIN OF ls_rep1,
    jednotka_id LIKE zcar_cars-jednotka_id,
    user_id LIKE zcar_cars-user_id,
    name LIKE zcar_cars-name,
    vin LIKE zcar_cars-vin,
    druhvozidla LIKE zcar_cars-druhvozidla,
    druhvozidla_text(25) TYPE c,
    spz LIKE zcar_cars-spz,
    aufnr TYPE aufnr,
    vyrobce LIKE zcar_cars-vyrobce,
    znacka LIKE zcar_cars-znacka,
    model LIKE zcar_cars-model,
    palivo LIKE zcar_cars-palivo,
    spotreba TYPE p DECIMALS 2, "zcar_cars-spotreba,
    nadrz LIKE zcar_cars-nadrz,
    tachometr LIKE zcar_cars-tachometr,
    user_id0 LIKE zcar_users-user_id,
    folder_id LIKE zcar_users-folder_id,
    name_first LIKE zcar_users-name_first,
    name_last LIKE zcar_users-name_last,
END OF ls_rep1.
DATA: lt_rep1 LIKE TABLE OF ls_rep1,
    ll_rep1 LIKE LINE OF lt_rep1.
FORM rep1.
CLEAR lt_rep1.
FIELD-SYMBOLS:<fs> TYPE g_zcar_cars->ls_zcar_cars.
SORT g_zcar_zak->lt_zcar_zak BY jednotka_id.
SORT g_zcar_users->lt_zcar_users BY user_id.
LOOP AT g_zcar_cars->lt_zcar_cars ASSIGNING <fs>.
    IF <fs>-spz IN g_cars.
        CLEAR ll_rep1.
        ll_rep1-jednotka_id = <fs>-jednotka_id.
        ll_rep1-user_id = <fs>-user_id.
        ll_rep1-name = <fs>-name.
        ll_rep1-vin = <fs>-vin.
        ll_rep1-spz = <fs>-spz.
        ll_rep1-aufnr = <fs>-aufnr.
        ll_rep1-vyrobce = <fs>-vyrobce.
        ll_rep1-znacka = <fs>-znacka.
```

```

ll_rep1-model = <fs>-model.
ll_rep1-palivo = <fs>-palivo.
ll_rep1-spotreba = <fs>-spotreba.
ll_rep1-nadrz = <fs>-nadrz.
ll_rep1-tachometr = <fs>-tachometr.
*   najdi zakazku
READ TABLE g_zcar_zak->lt_zcar_zak INTO g_zcar_zak->ll_zcar_zak
      WITH KEY jednotka_id = <fs>-jednotka_id
      BINARY SEARCH.
IF sy-subrc = 0.
    ll_rep1-aufnr = g_zcar_zak->ll_zcar_zak-aufnr.
ENDIF.
*   najdi druh voyidla
CASE <fs>-druhvozidla.
    WHEN 0.
        ll_rep1-druhvozidla_text = text-s12.
    WHEN 1.
        ll_rep1-druhvozidla_text = text-s13.
    WHEN 2.
        ll_rep1-druhvozidla_text = text-s14.
ENDCASE.
*   najdi uzivatete
READ TABLE g_zcar_users->lt_zcar_users INTO g_zcar_users->ll_zcar_users
      WITH KEY user_id = <fs>-user_id
      BINARY SEARCH.
IF sy-subrc = 0.
    ll_rep1-user_id0 = g_zcar_users->ll_zcar_users-user_id.
    ll_rep1-folder_id = g_zcar_users->ll_zcar_users-folder_id.
    ll_rep1-name_first = g_zcar_users->ll_zcar_users-name_first.
    ll_rep1-name_last = g_zcar_users->ll_zcar_users-name_last.
ENDIF.
APPEND ll_rep1 TO lt_rep1.
ENDIF.
ENDLOOP.
PERFORM set_layout.
PERFORM create_fiedcat_alv TABLES lt_fieldcat.
PERFORM show_data.
ENDFORM.                                "rep1

```

PŘÍLOHA P IX: REPORT NÁKLADY

FORM rep2.

CLEAR lt_zakazka.

IF LINES(g_cars) = 0.

FIELD-SYMBOLS:<fs> TYPE g_zcar_cars->ls_zcar_cars.

LOOP AT g_zcar_cars->lt_zcar_cars ASSIGNING <fs>.

 CLEAR ll_zakazka.

 IF <fs>-aufnr <> ''.

 ll_zakazka-sign = 'I'.

 ll_zakazka-option = 'EQ'.

 ll_zakazka-low = <fs>-aufnr.

 APPEND ll_zakazka TO lt_zakazka.

 ENDIF.

 ENDLOOP.

ELSE. " vezmi auta z filtru

 CLEAR l_cars.

 LOOP AT g_cars INTO l_cars.

 CLEAR ll_zakazka.

 ll_zakazka-low = g_zcar_cars->get_zak(l_cars-low).

 IF ll_zakazka-low <> ''.

 ll_zakazka-sign = l_cars-sign.

 ll_zakazka-option = l_cars-option.

 ll_zakazka-high = l_cars-high.

 APPEND ll_zakazka TO lt_zakazka.

 ENDIF.

 ENDLOOP.

ENDIF.

CLEAR lt_glpca.

DATA:

 mesic_od(2) TYPE c, " LIKE glpca-poper,

 mesic_do(2) TYPE c, "LIKE glpca-poper,

 rok_od(4) TYPE c, " LIKE glpca-ryear,

 rok_do(4) TYPE c, " glpca-ryear,

 den(2) TYPE c,

 datum_od LIKE sy-datum,

 datum_do LIKE sy-datum,

 lt_datum LIKE TABLE OF ls_zakazka WITH HEADER LINE,

 index TYPE i.

**vyber data ze zakazek v SAPu*

```
SELECT aufnr poper ryear budat racct msl hsl FROM glpca INTO CORRESPOND  
ING FIELDS OF TABLE lt_glpca
```

```
WHERE
```

```
aufnr IN lt_zakazka
```

```
AND budat IN lt_datum.
```

```
SORT g_zcar_cars->lt_zcar_cars BY aufnr.
```

```
PERFORM read_car_data_rep.
```

```
WHILE l_konec <> 0.
```

```
IF ( ( mesic = mesic_do ) AND ( rok = rok_do ) ).
```

```
l_konec = 0.
```

```
ENDIF.
```

```
CLEAR ll_zakazka.
```

```
CLEAR ll_rep2.
```

```
LOOP AT lt_zakazka INTO ll_zakazka.
```

```
ll_rep2-aufnr = ll_zakazka-low.
```

```
ll_rep2-poper = mesic.
```

```
ll_rep2-ryear = rok.
```

** dotahni data o vozidlech*

```
CLEAR g_zcar_cars->ll_zcar_cars.
```

```
READ TABLE g_zcar_cars->lt_zcar_cars INTO g_zcar_cars->ll_zcar_cars
```

```
WITH KEY aufnr = ll_rep2-aufnr
```

```
BINARY SEARCH.
```

```
IF sy-subrc = 0.
```

```
ll_rep2-user_id = g_zcar_cars->ll_zcar_cars-user_id.
```

```
ll_rep2-jednotka_id = g_zcar_cars->ll_zcar_cars-jednotka_id.
```

```
ll_rep2-name = g_zcar_cars->ll_zcar_cars-name.
```

```
ll_rep2-palivo = g_zcar_cars->ll_zcar_cars-palivo.
```

```
ll_rep2-nadrz = g_zcar_cars->ll_zcar_cars-nadrz.
```

```
ll_rep2-spz = g_zcar_cars->ll_zcar_cars-spz.
```

```
CLEAR g_zcar_import->ll_zcar_imp_data.
```

```
READ TABLE g_zcar_import->lt_zcar_imp_data INTO g_zcar_import-  
>ll_zcar_imp_data
```

```
WITH KEY spz = ll_rep2-spz mesic = ll_rep2-  
poper rok = ll_rep2-ryear typ = 0
```

```
BINARY SEARCH.
```

```

        IF sy-subrc = 0.
            ll_rep2-delka_osobni = g_zcar_import->ll_zcar_imp_data-
delka_osobni.
            ll_rep2-delka_firemni = g_zcar_import->ll_zcar_imp_data-
delka_firemni.
            ll_rep2-delka_suma = g_zcar_import->ll_zcar_imp_data-
delka_suma.
            ll_rep2-tachometr = g_zcar_import->ll_zcar_imp_data-tachometr.
        ENDIF.

*      dotahni data o nakladech
FIELD-SYMBOLS:<fsn> TYPE g_zcar_import->ls_zcar_imp_data.
ll_rep2-litry = 0.
LOOP AT g_zcar_import->lt_zcar_imp_data ASSIGNING <fsn>.
    IF <fsn>-spz = ll_rep2-spz AND <fsn>-mesic = ll_rep2-poper
        AND <fsn>-rok = ll_rep2-ryear AND <fsn>-typ = 1.
        IF <fsn>-tankovani = 'A'.    " naklad za tankovani
            ll_rep2-litry = ll_rep2-litry + <fsn>-litry.
            ll_rep2-cena_palivo = ll_rep2-cena_palivo + <fsn>-
cena_palivo.
            ll_rep2-cena_zaklad = ll_rep2-cena_zaklad + <fsn>-
cena_zaklad.
            ll_rep2-cena_celkem = ll_rep2-cena_celkem + <fsn>-
cena_celkem.
            IF ll_rep2-litry <> 0.
                ll_rep2-cena_l_phm = ll_rep2-cena_celkem / ll_rep2-litry.
            ENDIF.
        ELSE.
*      dolnit soucet za ostatni naklady dalsi_naklady.
            ll_rep2-dalsi_naklady = ll_rep2-dalsi_naklady + <fsn>-
cena_celkem.
*      <itab>-phm = '10'.
        ENDIF.
    ENDIF.
ENDLOOP.

ENDIF.
APPEND ll_rep2 TO lt_rep2.
ENDLOOP.

```

```

    mesic = mesic + 1.
    IF mesic = 13.
        mesic = 1.
        rok = rok + 1.
    ENDIF.
ENDWHILE.

CALL METHOD casline->krok.

wa_struct_type ?= cl_abap_typedescr=>describe_by_name( 'LS_REP2' ).
i_comp_tab = wa_struct_type->get_components( ).
APPEND LINES OF i_comp_tab TO i_comp_tot_tab.

LOOP AT g_zcar_coast->lt_zcar_coast INTO g_zcar_coast->ll_zcar_coast.
    wa_elem_type ?= cl_abap_elemdescr=>describe_by_name( 'VLCUR_PCA' ).
    wa_comp_fld-name = g_zcar_coast->ll_zcar_coast-key0.
    wa_comp_fld-type = cl_abap_elemdescr=>get_p(
        p_length = wa_elem_type->length
        p_decimals = wa_elem_type->decimals ).
    APPEND wa_comp_fld TO i_comp_tot_tab.
    IF g_zcar_coast->ll_zcar_coast-mnozstvi = 'X'.
        CONCATENATE 'MN_' g_zcar_coast->ll_zcar_coast-
key0 INTO wa_comp_fld-name.
        wa_comp_fld-type = cl_abap_elemdescr=>get_p(
            p_length = wa_elem_type->length
            p_decimals = wa_elem_type->decimals ).
        APPEND wa_comp_fld TO i_comp_tot_tab.
    ENDIF.
ENDLOOP.

wa_struct_type = cl_abap_structdescr=>create( i_comp_tot_tab ).
CREATE DATA wa_data_ref TYPE HANDLE wa_struct_type.
ASSIGN wa_data_ref->* TO <fs_wa_output>. .
CREATE DATA wa_data_ref LIKE STANDARD TABLE OF <fs_wa_output>.
ASSIGN wa_data_ref->* TO <fs_it_output>.

LOOP AT lt_rep2 INTO ll_rep2.
    APPEND ll_rep2 TO <fs_it_output>.
ENDLOOP.

```

```

SELECT *
    INTO CORRESPONDING FIELDS OF TABLE lt_zcar_coast_full FROM zcar_ucty
        WHERE bukr = gt_ucokruh.

FIELD-SYMBOLS: <lv_aufnr> TYPE ANY.
FIELD-SYMBOLS: <lv_poper> TYPE ANY.
FIELD-SYMBOLS: <lv_ryear> TYPE ANY.

FIELD-SYMBOLS: <lv_fieldval> TYPE ANY.
FIELD-SYMBOLS: <lv_fieldval_mn> TYPE ANY.

FIELD-SYMBOLS:<fs-uc> TYPE ls_glpca.
SORT lt_rep2 BY aufnr ryear poper.

DATA aa TYPE string.

LOOP AT lt_glpca ASSIGNING <fs-uc>. " cti tabulku zakazek
    LOOP AT <fs_it_output> ASSIGNING <fs_wa_output>. " cti data v reportu

        ASSIGN COMPONENT 'AUFNR' OF STRUCTURE <fs_wa_output> TO <lv_aufnr>.
        ASSIGN COMPONENT 'POPER' OF STRUCTURE <fs_wa_output> TO <lv_poper>.
        ASSIGN COMPONENT 'RYEAR' OF STRUCTURE <fs_wa_output> TO <lv_ryear>.

        IF <fs-uc>-aufnr = <lv_aufnr> AND <fs-uc>-
poper = <lv_poper> AND <fs-uc>-ryear = <lv_ryear>.
            *          CLEAR lt_coastkey0.
            LOOP AT g_zcar_coast->lt_zcar_coast INTO g_zcar_coast-
>ll_zcar_coast." nacti seynam uctu
                CLEAR lt_coastkey0[].
                LOOP AT lt_zcar_coast_full INTO ll_zcar_coast_full.
                    IF ll_zcar_coast_full-key0 = g_zcar_coast->ll_zcar_coast-
key0.
                        *          clear
                        lt_coastkey0-sign = ll_zcar_coast_full-sign.
                        lt_coastkey0-option = ll_zcar_coast_full-option0.
                        lt_coastkey0-high = ll_zcar_coast_full-high.
                        lt_coastkey0-low = ll_zcar_coast_full-low.

```

```

        APPEND lt_coastkey0.
    ENDIF.
ENDLOOP.

    IF <fs-uc>-
racct IN lt_coastkey0. " jesltize je shodnz ucet, pricti dan sloupec
        ASSIGN COMPONENT g_zcar_coast->ll_zcar_coast-
key0 OF STRUCTURE <fs_wa_output> TO <lv_fieldval>.
        <lv_fieldval> = <lv_fieldval> + <fs-uc>-hsl.

        IF g_zcar_coast->ll_zcar_coast-mnozstvi = 'X'. " je-
li pot5eba nac9tat i mnoystvi
            CONCATENATE 'MN_' g_zcar_coast->ll_zcar_coast-
key0 INTO fieldname.
            ASSIGN COMPONENT fieldname OF STRUCTURE <fs_wa_output> TO <
lv_fieldval_mn>.
            <lv_fieldval_mn> = <lv_fieldval_mn> + <fs-uc>-msl.
*            UNASSIGN <lv_fieldval_mn>.
        ENDIF.

    ENDIF.

ENDLOOP.

    ENDIF.    ENDLOOP.

ENDLOOP.

FREE casline.
PERFORM set_layout_rep2.
PERFORM create_fiedcat_alv_rep2 TABLES lt_fieldcat.
PERFORM show_data_rep2.

ENDFORM.                "rep2

```

PŘÍLOHA P X: ZOBRAZENÍ REPORTU NÁKLADŮ

Jednotka		ID IdU	ID IdUce	Zadání	Učební odd.	Fiskal.	SPZ	Pařto	Nadrz.	Soukromé K.	Fremm K.	Cekem KM	Stav	radho	CC-pole	CC-cena pa	CC-cena za	CC-cena ce	CC-dřel.	Cena PHM	Cena za KM	Dahmž z	Odpoř DN	Ostani sř NM	Ostani Pohomě hm	
37331	56676	Zajna Radm	97104-25953	3	2013	4259583	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7718.00	0.00	6118.38	
37331	56676		97104-25953	4	2013	4259583	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7718.00	260.00	0.00	3903.04
37331	56676		97104-25953	5	2013	4259583	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7718.00	111.00	0.00	4110.22
37331	56676		97104-25953	6	2013	4259583	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7718.00	0.00	0.00	5496.77
37331	56676		97104-25953	7	2013	4259583	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7718.00	102.40	0.00	5269.57
37331	56676		97104-25953	8	2013	4259583	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7718.00	0.00	0.00	3986.91
37331	56676		97104-25953	9	2013	4259583	Nafra	55.00	445.560	2.103.90	2.549	102.048.30	123.640	337.720	9.267.34	11.179.0	0.000	90.548	0.000	0.00	0.00	0.00	7718.00	0.00	0.00	6772.21
37331	56676		97104-25953	10	2013	4259583	Nafra	55.00	1.240.40	2.807.01	4.067	106.005.45	208.190	545.670	13.964.4	18.394.2	0.000	88.353	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	11	2013	4259583	Nafra	55.00	975.40	2.826.38	3.802	109.987.57	169.980	463.460	13.529.6	19.433.4	0.000	114.339	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	12	2013	4259583	Nafra	55.00	494.20	2.015.88	2.510	112.407.68	154.670	443.120	11.525.6	15.678.8	0.000	102.656	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	1	2014	4259583	Nafra	55.00	335.010	2.488.00	3.023	115.430.69	170.120	1.039.16	13.536.0	17.881.6	0.000	105.112	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	2	2014	4259583	Nafra	55.00	349.420	2.036.86	2.368	117.816.92	146.920	368.200	10.000.3	13.659.5	0.000	92.867	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	3	2014	4259583	Nafra	55.00	615.660	2.067.37	3.283	121.100.00	133.070	232.970	6.304.05	9.161.89	0.000	68.850	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	4	2014	4259583	Nafra	55.00	618.980	2.346.01	2.965	124.064.99	215.550	263.660	7.520.35	11.980.3	0.000	55.580	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	5	2014	4259583	Nafra	55.00	981.270	2.017.77	2.999	127.064.03	167.110	431.928	12.336.2	16.520.5	0.000	98.680	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	6	2014	4259583	Nafra	55.00	1.227.920	2.672.80	3.901	130.964.75	455.160	1.360.58	36.621.9	48.732.6	0.000	107.054	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	7	2014	4259583	Nafra	55.00	107.400	2.599.30	2.707	133.671.45	130.590	482.640	14.371.2	18.732.6	0.000	143.690	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	8	2014	4259583	Nafra	55.00	1.0052.250	3.707.01	4.739	138.430.71	324.390	593.660	16.841.6	21.797.2	0.000	67.132	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	9	2014	4259583	Nafra	55.00	286.470	3.874.75	4.161	142.591.93	192.900	556.890	15.388.1	18.565.5	0.000	96.245	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	10	2014	4259583	Nafra	55.00	716.070	3.237.44	3.954	146.545.44	263.250	673.850	18.688.3	22.540.6	0.000	85.297	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	3	2013	3278311	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00
37331	56676		97104-25953	4	2013	3278311	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00
37331	56676		97104-25953	5	2013	3278311	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00
37331	56676		97104-25953	6	2013	3278311	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00
37331	56676		97104-25953	7	2013	3278311	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00
37331	56676		97104-25953	8	2013	3278311	Nafra	55.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00
37331	56676		97104-25953	9	2013	3278311	Nafra	55.00	186.610	1.946.36	2.133	149.126.00	0.000	68.240	2.284.07	2.763.72	0.000	34.120	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	10	2013	3278311	Nafra	55.00	460.460	2.056.32	2.517	151.642.78	0.000	81.480	2.446.44	2.960.44	0.000	27.160	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	11	2013	3278311	Nafra	55.00	511.560	1.149.99	1.662	153.304.73	0.000	66.660	1.956.00	2.366.76	0.000	33.320	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	12	2013	3278311	Nafra	55.00	374.150	1.525.29	1.899	155.204.17	0.000	102.810	2.275.92	3.338.46	0.000	34.270	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	1	2014	3278311	Nafra	55.00	62.490	1.099.99	1.162	156.366.50	0.000	179.560	1.901.76	2.301.12	0.000	33.440	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	2	2014	3278311	Nafra	55.00	116.900	1.559.60	1.677	158.043.00	0.000	67.220	2.138.82	2.587.97	0.000	33.610	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	3	2014	3278311	Nafra	55.00	272.500	2.323.00	2.596	160.638.50	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00	
37331	56676		97104-25953	4	2014	3278311	Nafra	55.00	357.580	1.922.60	2.280	162.918.40	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00	
37331	56676		97104-25953	5	2014	3278311	Nafra	55.00	507.280	2.565.32	3.073	165.991.00	0.000	99.780	3.106.10	3.758.38	0.000	33.260	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	6	2014	3278311	Nafra	55.00	4751.00	2.332.70	2.808	168.798.90	0.000	316.760	9.689.64	11.724.4	0.000	31.688	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	7	2014	3278311	Nafra	55.00	574.100	2.751.30	3.325	172.124.20	0.000	125.840	4.264.00	5.159.44	0.000	31.460	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	8	2014	3278311	Nafra	55.00	181.230	1.440.00	1.621	173.745.43	0.000	63.400	1.886.28	2.282.40	0.000	31.700	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	9	2014	3278311	Nafra	55.00	287.320	2.539.34	2.827	176.572.09	0.000	95.100	3.093.40	3.740.60	0.000	31.700	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37331	56676		97104-25953	10	2014	3278311	Nafra	55.00	323.630	2.026.79	2.350	178.922.51	0.000	92.100	3.146.12	3.806.80	0.000	30.700	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37320	56668	Šumák Jitř	97104-253458	3	2013	4253458	Nafra	45.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00
37320	56668		97104-253458	4	2013	4253458	Nafra	45.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00
37320	56668		97104-253458	5	2013	4253458	Nafra	45.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00
37320	56668		97104-253458	6	2013	4253458	Nafra	45.00	0.000	0.000	0.000	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.00	0.00	0.00
37320	56668		97104-253458	7	2013	4253458	Nafra	45.00	0.000	0.000																