

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Diplomová práce**

# **Návrh a implementace datového skladu pro studentskou skupinu UWB Racing Team Pilsen**

## Prohlášení

Prohlašuji, že jsem diplomovou práci vypracovala samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne .....

.....

Kristýna Šimová

## **Abstrakt:**

### **Návrh a implementace datového skladu pro studentskou skupinu UWB Racing Team Pilsen**

Cílem této práce je návrh a implementace datového skladu spolu s aplikací, sloužící pro práci s daty, které se týkají nákladů vynaložených na vývoj vozidla vytvořeného studentským týmem UWB Racing Team Pilsen. Práce se zabývá výběrem a implementací dílčích komponent ze sady nástrojů Business Intelligence. Mezi vybrané komponenty patří především již ze zadání datový sklad, poté ETL proces, kterým je datový sklad plněn daty, a OLAP technologie zajišťující průzkum a analýzu dat v datovém skladu. Další část práce se věnuje návrhu a implementaci aplikace, skrze kterou členové studentské skupiny přistupují k datům v datovém skladu a která poskytuje analytická data v podobě grafických výstupů. Vytvořené řešení slouží pro podporu strategického rozhodování týmu UWB Racing Team Pilsen v oblasti snižování nákladů vytvářené vozidla. Součástí práce je i tzv. Proof of concept, jehož účelem je vytvoření řešení pro analýzu dat všech studentských týmů, které se soutěže účastní.

***Klíčová slova:** datový sklad, analýza dat, UWB Racing Team Pilsen, Business Intelligence, ETL, OLAP*

## **Abstract:**

### **Design and implementation of data warehouse for students group UWB Racing Team Pilsen**

The aim of this thesis is design and implementation of data warehouse with an application for working with data related to the cost of a vehicle developed by the UWB Racing Team Pilsen. The thesis deals with the selection of Business Intelligence components. The selected components include the data warehouse, ETL process, by which the data warehouse is filled, and OLAP technology for data warehouse data analysis. The next part of the thesis deals with the design and implementation of the application through which the members of the student group access data in the data warehouse and which provides analytical data in the form of graphical outputs. The solution is designed to support the strategic decision-making of the UWB Racing Team Pilsen in area of reducing costs. The thesis also contains Proof of concept, the purpose of which is to create a data analysis solution for data of all student teams participating in the competition.

***Key words:** data warehouse, data analysis, UWB Racing Team Pilsen, Business Intelligence, ETL, OLAP*

# Obsah

<b>1</b>	<b>Úvod</b> .....	<b>4</b>
<b>2</b>	<b>Požadavky</b> .....	<b>5</b>
2.1	Oblast aplikace pro aktuální data .....	6
2.1.1	Import.....	6
2.1.2	Datová vrstva .....	10
2.1.3	Prezentační vrstva .....	13
2.1.4	Export zprávy o nákladech .....	17
2.2	Oblast aplikace pro historická data .....	18
2.2.1	Porovnání v průběhu let.....	18
2.2.2	Porovnání v rámci konkrétní oblasti komodit.....	18
2.2.3	Informace konkrétních sestav .....	20
2.2.4	Porovnání v rámci konkrétního roku .....	21
<b>3</b>	<b>Business Intelligence</b> .....	<b>24</b>
3.1	Vrstva extrakce, transformace a nahrávání dat .....	24
3.1.1	ETL .....	24
3.1.2	EAI.....	25
3.2	Vrstva pro ukládání dat .....	25
3.2.1	Datový sklad .....	25
3.2.2	Datové tržiště .....	26
3.2.3	Operační datové úložiště.....	26
3.2.4	Dočasné úložiště dat .....	26
3.3	Vrstva pro analýzu dat .....	26
3.3.1	OLAP .....	27
3.4	Prezentační vrstva .....	34
3.4.1	EIS .....	34
<b>4</b>	<b>Návrh datové vrstvy</b> .....	<b>35</b>
4.1	Operační databáze .....	35
4.2	Datový sklad.....	36
4.2.1	Výběr dimenzí.....	37
4.2.2	Výběr faktů .....	38
4.2.3	OLAP .....	39
4.2.4	Logická struktura datového skladu.....	39

4.2.5	Rozšíření datového skladu .....	40
<b>5</b>	<b>Návrh aplikační vrstvy .....</b>	<b>45</b>
5.1	Oblast aplikace pro aktuální data .....	45
5.2	Oblast aplikace pro historická data .....	49
5.3	Rozšíření aplikační vrstvy .....	49
<b>6</b>	<b>Výběr technologie .....</b>	<b>53</b>
6.1	Systém řízení báze dat.....	53
6.2	Programovací jazyk.....	54
6.3	Technologie pro vývoj uživatelského rozhraní .....	55
6.4	Technologie pro manipulaci dat operační databáze .....	56
<b>7</b>	<b>Implementace .....</b>	<b>57</b>
7.1	Databázová vrstva .....	57
7.1.1	Uživatелеm definované procedury .....	57
7.1.2	Uživatелеm definované funkce .....	64
7.2	Aplikační vrstva .....	69
7.2.1	Architektura aplikace .....	69
7.2.2	ETL .....	72
7.2.3	OLAP .....	78
7.2.4	Použité knihovny .....	79
7.2.5	Uživatelské rozhraní .....	80
<b>8</b>	<b>Testování.....</b>	<b>83</b>
8.1	ETL .....	83
8.1.1	Extrakce .....	83
8.1.2	Transformace .....	84
8.1.3	Načtení dat .....	84
8.2	OLAP .....	84
8.3	Uživatelské rozhraní.....	85
<b>9</b>	<b>Závěr .....</b>	<b>86</b>
	<b>Seznam příloh.....</b>	<b>89</b>

# 1 Úvod

Studentská skupina UWB Racing Team Pilsen, která je zadavatelem této práce, se účastní mezinárodní soutěže Formula Student, která je každoročně pořádaná organizací SAE International [1]. Historie této soutěže sahá až do roku 1979 a účastní se jí více než 500 technických univerzit z celého světa. Cílem každého týmu, složeného ze studentů fakult dané univerzity, je postavit monopost formulového typu podle pravidel soutěže pro daný rok. Týmy tráví 8-12 měsíců při navrhování, stavbě a přípravě svých vozidel do soutěže.

Vytvářené vozy nejsou posuzovány pouze dle klasických rychlostních závodů, ale jsou hodnoceny hned v několika disciplínách. Mezi tyto disciplíny patří například technické kontroly, prezentace projektu, inženýrský design, konstrukční řešení a sólové výkonnostní zkoušky.

Tato soutěž dává týmům možnost prokázat jak svou kreativitu, tak technické dovednosti a porovnat je na světové úrovni. Studenti zde své znalosti a dovednosti, získané na univerzitě, převádějí do praxe. Jelikož nízká výše nákladů zůstává v praktickém životě všeobecně rozhodujícím kritériem, je nedílnou součástí této soutěže i posuzování vozu dle celkových nákladů. Právě na tuto disciplínu je tato práce zaměřena. V rámci této disciplíny je tým povinen sestavit podrobnou zprávu o všech nákladech, které byly na vozidlo vynaloženy.

Cílem práce je snížit čas a úsilí univerzitního týmu UWB Racing Team Pilsen nutného na sběr, formátování a čištění informací ze surových dat. Takto získané informace jsou pak klíčové pro strategické rozhodování týmu v oblasti snižování nákladů. Surovými daty jsou zde myšleny výše zmíněné zprávy o nákladech, jejichž rozsah se pohybuje zhruba kolem šesti set stránek. Získat z těchto dokumentů důležité ukazatele je proto netriviální záležitostí. Předmětem práce je navrhnout a implementovat datový sklad, který bude uchovávat historická data o nákladech na všechna vozidla konstruována tímto týmem. Aplikace, skrze kterou budou členové týmu k datům přistupovat, bude poskytovat pohledy na dřívější a současné nákladové položky. Nástroj bude dále umožňovat analýzu ve smyslu nacházení souvislostí, které nejsou z primárních dat na první pohled zřejmé. Účelem celkového řešení tedy bude zefektivnění a podpora rozhodování na strategické úrovni studentského týmu UWB Racing Team Pilsen.

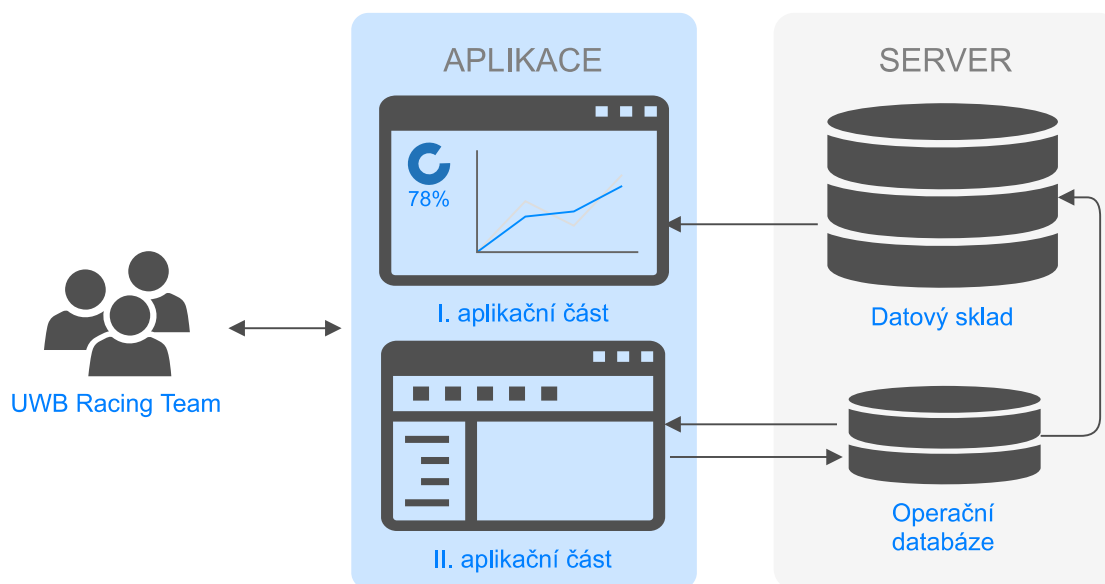
Práce bude rozšířena i o vývoj řešení, které bude sloužit pro práci s daty všech studentských týmů účastnících se soutěže. Toto rozšíření bude v rámci práce figurovat jako tzv. *Proof of Concept* a jeho účelem bude porovnání nákladů napříč všemi studentskými týmy. Vzniká nám zde tak možnost analýzy všech těchto dat a nalezení případných anomálií v rámci výše nákladů jednotlivých týmů.

## 2 Požadavky

V této kapitole budou postupně probrány všechny požadavky studentského týmu UWB Racing Team na aplikaci patřící do celkového řešení této práce.

Jedná se o klientskou aplikaci, kterou budou mít k dispozici jednotliví členové týmu. Aplikace se bude připojovat na server, kde budou uložena data. Aplikace dále umožní ukládat plnohodnotná data pro období, kdy se aktuálně vozidlo vytváří. Po finálním ocenění nákladů vozidla, budou tato data přesunuta do datového skladu. Z tohoto důvodu se aplikace bude dělit na dvě hlavní oblasti. První část aplikace bude sloužit především pro ukládání a ocenění všech součástí vozidla za aktuální rok a druhá část aplikace bude poskytovat celkový přehled ceny (či jiné ukazatele) všech vytvořených vozidel v průběhu let.

Na následujícím Obrázku č. 2.1 je zobrazen koncept aplikační a datové vrstvy. Je zde vidět výše vysvětlené rozdělení aplikace na dvě hlavní oblasti. Jak je z obrázku zřejmé, aplikace se bude připojovat na server, kde se bude nacházet jak datový sklad, tak i operační databáze.



Obrázek č. 2.1: Konceptuální model aplikace

Mezi další požadavky zadavatele patří, že aplikace musí být spustitelná minimálně na operačním systému Windows 7. Dále zadavatel výrazně upřednostňuje úsporu paměťového prostoru nad rychlostí a výkonem celkového řešení.

S rozdělením aplikace bude rozdělena i tato kapitola na dvě hlavní části, které budou oblastem aplikace odpovídat.

## 2.1 Oblast aplikace pro aktuální data

Hlavním cílem této části aplikace je umožnit členům vkládat všechny součástky vozidla do operační databáze. Tyto součástky budou v aplikaci uživateli oceněny a po dokončení zadávání bude aplikace schopna exportovat zprávu o nákladech, které byly na vozidlo vynaloženy (viz kapitola 2.1.4). Popis jakým způsobem budou všechny typy materiálu, spojovacího materiálu, procesů a opracování oceněny je popsán v souborech od zadavatelů (zdroj [2]), a tyto soubory bude aplikace importovat a dle nich i zmíněné položky oceňovat.

### 2.1.1 Import

Jak již bylo výše zmíněno, zadavatel poskytne soubory s popisem kategorií nákladů, které budou předmětem celkového ocenění vozidla. Pro ukázkou se v adresáři „*attachment*“ na přiloženém CD nacházejí zmíněné soubory z roku 2016:

- „*Fasteners.xlsx*“
- „*Materials.xlsx*“
- „*Processes.xlsx*“
- „*ProcessMultipliers.xlsx*“
- „*Tooling.xlsx*“

Soubor „*Fasteners.xlsx*“ obsahuje výčet všech typů **spojovacího materiálu**, kde pro každý typ jsou k dispozici informace o ceně spojovacího materiálu, případně informace nutné pro výpočet jeho ceny. V Tabulce č. 2.1 se nachází příklad dat, která se mohou v takovém souboru objevit.

Fastener Name	Unit Price	Unit1	Unit2	C1	C2	Table Price	Comments
Steel Loop Straps, Rubber-Cushioned	-	mm	-	0,01	0,01	$[C1]*[Size1]+[C2]$	[Size1]=final clamped diameter [mm]
Retaining Ring, Internal	-	mm	-	0,02	0,08	$[C1]*([Size1]^2)+[C2]$	[Size1]=shaft diameter [mm]
Hook and Loop, Hook Side	\$ 0,03	cm <sup>2</sup>	-	-	-	-	Only includes one side of joint (hook side)
Eyebolt, Threaded, Steel	-	mm	-	0,87	0,05	$[C1]*EXP([C2]*[Size1])$	[Size1]=Inner Diameter of Eye
Nut, Lug	\$ 0,04	unit	-	-	-	-	Any size
Pin, Safety, Retainer Snap	-	mm	mm	0,01	1	$[C1]*[Size1]*[Size2]+[C2]$	[Size1]=Pin Diameter (mm), [Size2]=Usable length (mm).

Tabulka č. 2.1: Ukázka dat souboru s typy spojovacího materiálu



U jednotlivých typů spojovacího materiálu se mohou objevit následující atributy:

<i>Fastener Name</i>	název typu spojovacího materiálu
<i>Unit Price</i>	jednotková cena typu spojovacího materiálu
<i>Size 1</i>	hodnota vstupující do vzorce pro výpočet ceny (zadáva uživatel)
<i>Size 2</i>	hodnota vstupující do vzorce pro výpočet ceny (zadáva uživatel)
<i>Unit 1</i>	měrná jednotka pro hodnotu <i>Size 1</i>
<i>Unit 2</i>	měrná jednotka pro hodnotu <i>Size 2</i>
<i>C1</i>	hodnota konstanty vstupující do vzorce pro výpočet ceny
<i>C2</i>	hodnota konstanty vstupující do vzorce pro výpočet ceny
<i>Table Price</i>	vzorec pro výpočet ceny spojovacího materiálu
<i>Comments</i>	komentář

Jak lze vidět v tomto souboru, v hodnotách *Table Price* se vyskytují atributy vstupující do výpočtu ceny *C1*, *C2*, což jsou konstanty pevně dané, a *Size1*, *Size2*, což jsou hodnoty, které zadává uživatel.

Soubor „*Materials.xlsx*“ obsahuje výčet všech typů **materiálu**, kde pro každý typ jsou k dispozici informace o ceně materiálu, či případně opět informace nutné pro výpočet jeho ceny. V Tabulce č. 2.2 se nachází příklad dat, která se mohou v tomto souboru objevit.

Material Name	Category	Unit Price	Unit1	Unit2	C1	C2	Table Price	Comments
Bearing Ball, Steel	Bearings	-	mm	-	0,03	0,05	$[C1]*[Size1]+[C2]$	[Size1]=Ball diameter (mm)
Datalogger, Aim MXL Pista	Control Module	\$ 1000	unt	-	-	-	-	Dash display data logger
ECU, Haltech, E8	Control Module	\$ 762	unit	-	-	-	-	4 c seq
Belt	Drivetrain	\$ 0,05	mm	-	-	-	-	-
Display, LCD	Electronics	-	cm	cm	1,25	0	$[C1]*[Size1]*[Size2]+[C2]$	[Size1]=Display surface area (cm <sup>2</sup> )
Engine and Transmission	Engine	\$ 2,00	cc	-	0,01	1	-	-
Rubber	Raw Material	\$ 3,30	kg	-	-	-	-	Density= 1100kg/m <sup>3</sup>

Tabulka č. 2.2: Ukázka dat souboru s typy materiálu

Pro jednotlivé typy materiálu mohou být zobrazeny následující atributy:

<i>Material Name</i>	název typu materiálu
<i>Category</i>	název kategorie, do které daný typ materiálu patří
<i>Unit Price</i>	jednotková cena typu materiálu
<i>Table Price</i>	vzorec pro výpočet ceny spojovacího materiálu
<i>Size 1</i>	hodnota vstupující do výpočtu ceny (zadáva uživatel při výpočtu)
<i>Size 2</i>	hodnota vstupující do výpočtu ceny (zadáva uživatel při výpočtu)
<i>Unit 1</i>	měrná jednotka pro hodnotu <i>Size 1</i>
<i>Unit 2</i>	měrná jednotka pro hodnotu <i>Size 2</i>
<i>C1</i>	hodnota konstanty vstupující do vzorce pro výpočet ceny
<i>C2</i>	hodnota konstanty vstupující do vzorce pro výpočet ceny
<i>Comments</i>	komentář

I zde v hodnotách *Table Price* se vyskytují atributy vstupující do výpočtu ceny *c1*, *c2*, což jsou konstanty opět pevně dané, a atributy *Size1*, *Size2*, které zadává uživatel.

V souboru „*Processes.xlsx*“ je obsažen výčet všech typů **procesů**. Pro ukázkou se v Tabulce č. 2.3 nachází příklad dat tohoto souboru.

Process Name	Category	Unit Price	Unit	Tooling Required	Multiplier Type Used	Comments
Die Casting	Basic Forming	\$ 4,00	kg	Yes	-	-
Adjustment - Misc.	Labor	\$ 5,00	unit	-	-	Chain tension, etc.
Assemble, >20 kg, Loose	Labor	\$ 1,875	unit	-	Assembly	-
Drilled hole < 50.8 mm dia.	Material Removal	\$ 0,7	hole	-	Drill & Tap	Holes >= 50.8 mm must be machined
Grind, Flat	Material Removal	\$ 0,15	cm <sup>2</sup>	-	Machining	0.25mm machine stock
Tube cut	Tubing	\$ 0,15	cm	-	-	Use diameter of tube

Tabulka 2.3: Ukázka dat souboru s typy procesu

Pro jednotlivé typy procesů se v souboru mohou nacházet následující atributy:

<i>Process Name</i>	název typu procesu
<i>Unit Cost</i>	jednotková cena typu procesu
<i>Unit</i>	měrná jednotka
<i>Category</i>	název kategorie, do které daný typ patří
<i>Tooling Required</i>	zda je vyžadováno použití opracování
<i>Multiplier Type Used</i>	název použitého multiplikátoru
<i>Comments</i>	komentář

V souboru „*ProcessMultipliers.xlsx*“ jsou doplňující informace pro kategorii procesů. Zde se nachází výčet multiplikátorů procesů, které jsou důležité pro výpočet konečné ceny vybraných typů procesu (viz kapitola Specifikace výpočtu celkových cen).

Process Multiplier	Multiplier	Use
Assemble - Length > 0.5m	1,25	Assembly
Fastener Engagement Length > 2D	1,25	Fastener Installation
Material - Plastic	0,5	Machining
Material - Titanium	3,65	Machining

Tabulka č. 2.4: Ukázka dat souboru s multiplikátory procesů

Pro každý konkrétní multiplikátor jsou v souboru obsaženy atributy:

<i>Process Multiplier</i>	název multiplikátoru procesu
<i>Multiplier Value</i>	hodnota multiplikátoru
<i>Use</i>	název druhu použití

V souboru „*Tooling.xlsx*“ se nachází výčet jednotlivých typů **opracování**. V následující Tabulce č. 2.5 je zobrazen příklad dat takového souboru.

Process	Tool	Cost	Unit	Comments
Die Casting	Die Casting - Die	\$ 10 000	die	Per die not die set. Minimum number of dies is 2 per
Plastic injection molding	Plastic injection molding - Die	\$ 10 000	die	Per die not die set.
Sand Casting	Sand Casting - Sand Core Package	\$ 5 000	core	Per core not core package.
Braze	Brazing Fixture	\$ 500	point	Each point is a pickup or support point.

Tabulka č. 2.5: Ukázka dat souboru s typy opracování

Pro každý typ opracování jsou k dispozici následující atributy:

<i>Process</i>	typ procesu, pro který se daný typ opracování používá
<i>Tool</i>	název nástroje, který je použit při opracování
<i>Cost</i>	cena daného typu
<i>Unit</i>	měrná jednotka
<i>Comments</i>	komentář k typu opracování

## 2.1.2 Datová vrstva

Databáze by měla udržovat informace o nákladech rozdělené do tzv. oblastí komodit. Vozidlo je členěno na celkem osm oblastí komodit, mezi které patří:

*Brake System*  
*Engine & Drivetrain*  
*Frame & Body*  
*Instruments & Wiring*  
*Miscellaneous, Fit & Finish*  
*Steering System*  
*Suspension & Shocks*  
*Wheels & Tires*

Tyto oblasti obsahují položky ve dvouúrovňové hierarchii. V první úrovni položek se nachází sestavy, které sdružují položky druhé úrovně tzv. části sestav. Do položek jak první tak druhé úrovně musí být možné zadávat materiál, proces, opracování a spojový materiál, který byl pro takovou položku použit. Způsob výpočtu ceny materiálu, procesu, opracování či spojovacího materiálu může být zcela rozdílný, o tom ale více v následujících dvou kapitolách.

### Specifikace výpočtu jednotkových cen

U tabulek s materiálem a spojovým materiálem může nastat případ, kdy jednotková cena není pevně daná, ale vypočítává se zadaným vzorcem. V takové situaci bude uživatel zadávat hodnoty atributů *Size 1* a *Size 2* (pokud se vyskytují ve vzorci), které vstupující do výpočtu jednotkové ceny, například:

$$UC = [C1] * [Size1]^2 + [C2]$$

kde, platí že:

*UC*    *jednotková cena*  
*C1*    *konstanta, pevně daná*  
*C2*    *konstanta, pevně daná*  
*Size1* *atribut zadaný uživatelem*

## Specifikace výpočtu celkových cen

### Spojový materiál

U spojovacího materiálu je jednotková cena buďto zadána nebo je vypočtena dle zadaného vzorce (viz předchozí kapitola Specifikace výpočtu jednotkových cen). Jinak se cena celková počítá stejně pro všechny druhy spojovacího materiálu. V Tabulce č. 2.6 jsou vyznačeny zeleně parametry zadávané uživatelem a modře hodnoty pevně dané. Způsoby výpočtu cen spojovacího materiálu:

Unit Cost	Sub Total
[Formula] => Calculated Unit Cost	[Unit Cost] * [Quantity]
[Unit Cost]	[Unit Cost] * [Quantity]

Tabulka č. 2.6: Druhy výpočtu ceny spojovacího materiálu

### Procesy

Jednotková cena je pro všechny typy procesů jednoznačně zadána. Jak je zřejmé z Tabulky č. 2.7, zde se výpočet celkové ceny procesu liší, což je dáno tím, zda je u konkrétního typu procesu definován multiplikátor či nikoli. Způsoby výpočtu cen procesu:

Multiplier	Sub Total
[Multiplier Name]	[Unit Cost] * [Quantity] * [Multiplier Value]
-	[Unit Cost] * [Quantity]

Tabulka č. 2.7: Druhy výpočtu ceny procesu

### Opracování

Jednotková cena je pro všechny typy opracování již zadána. V tomto případě existuje pouze jediný vzorec pro výpočet celkové ceny opracování.

Unit Cost	Sub Total
[Unit Cost]	([Unit Cost] * [Quantity]) / ([PVF] * [FracInId])

Tabulka č. 2.8: Způsob výpočtu ceny opracování

## Material

Na následující Tabulce č. 2.9 se nachází způsoby výpočtu ceny materiálu. Cena za jednotku materiálu je buďto zadaná nebo se musí dopočítat dle zadaného vzorce (viz kapitola Specifikace výpočtu jednotkových cen). Dále se rozlišuje způsob výpočtu ceny celkové, jenž závisí na měrné jednotce *Unit*, která se vztahuje k jednotkové ceně.

Unit	Unit Cost	Sub Total	Note
-	[Formula] => Calculated Unit Cost	[Unit Cost] * [Quantity]	
unit	[Unit Cost]	[Unit Cost] * [Quantity]	
kg	[Unit Cost]	[Unit Cost]*[Area]*[Length]*[Density]*10 <sup>-9</sup>	[Length] in mm [Area] in mm <sup>2</sup> [Density] in kg/m <sup>3</sup>
cm	[Unit Cost]	[Unit Cost] * [Length]	[Length] in cm
m	[Unit Cost]	[Unit Cost] * [Length]	[Length] in m
pin(s)	[Unit Cost]	[Unit Cost] * [Quantity]	[Quantity] in pin(s)
digit	[Unit Cost]	[Unit Cost] * [Quantity]	[Quantity] in digit
cm <sup>2</sup>	[Unit Cost]	[Unit Cost] * [Area]	[Area] in cm <sup>2</sup>
m <sup>2</sup>	[Unit Cost]	[Unit Cost] * [Area]	[Area] in m <sup>2</sup>
cc	[Unit Cost]	[Unit Cost] * [Quantity]	[Quantity] in cc
mm <sup>3</sup>	[Unit Cost]	[Unit Cost] * [Area] * [Length]	[Length] in mm [Area] in mm <sup>2</sup>
cm <sup>3</sup>	[Unit Cost]	[Unit Cost] * [Area] * [Length]	[Length] in cm [Area] in cm <sup>2</sup>
kw	[Unit Cost]	[Unit Cost] * [Quantity]	[Quantity] in kw
pack	[Unit Cost]	[Unit Cost] * [Quantity]	[Quantity] in pack
channel	[Unit Cost]	[Unit Cost] * [Quantity]	[Quantity] in channel
kwh	[Unit Cost]	[Unit Cost] * [Quantity]	[Quantity] in kwh
pin	[Unit Cost]	[Unit Cost] * [Quantity]	[Quantity] in pin
litr	[Unit Cost]	[Unit Cost] * [Quantity]	[Quantity] in liter

Tabulka č. 2.9: Druhy výpočtu ceny materiálu

## 2.1.3 Prezentační vrstva

### Okno položky

Z důvodu snadné orientace členů týmu, bude design okna položky podobný vzor exportování položky, který lze vidět v adresáři elektronické přílohy „*attachment*“ v souboru „*FCA Input.xls*“ na listu „*part*“. V tomto okně tedy bude umožněno zobrazit:

- I. Informace o položce
  - a. Název položky
  - b. V případě, že položka není sestavou ale pouze částí sestavy, bude uveden název sestavy, jejíž je součástí
  - c. Oblast komodit, do které položka spadá
  - d. Pořadí položky v rámci oblasti komodit (pokud se jedná o sestavu) nebo v rámci sestavy (pokud se jedná o část sestavy)
  - e. Verze položky
  - f. Poznámky k položce
- II. Tabulky
  - a. Tabulka procesů položky
  - b. Tabulka materiálů položky
  - c. Tabulka spojovacího materiálu položky
  - d. Tabulka opracování položky
- III. Sumační informace
  - a. Jednotková cena
  - b. Cena celková

V případě výše zmíněných tabulek materiálu, spojovacího materiálu, procesů a opracování položky bude okno umožňovat editaci, smazání či změnu pořadí jednotlivých řádek. Dále bude okno položky umožňovat přidávat či odebírat položce obrázek.

### Okna formulářů položky

Budou vytvořena okna s formuláři, prostřednictvím kterých bude uživatel moci položce přidat materiál, proces, spojový materiál či opracování. Se zadavatelem byla vytvořena předběžná podoba okna, kterou lze vidět na Obrázku č. 2.2.

Zde můžeme vidět, že okno je rozděleno do dvou stěžejních částí. První část (modře označena), slouží výhradně pro vybrání typu nově vytvářeného nákladu (materiál, proces, spoj. materiál či opracování), který chceme položce přidat. V této části se tedy bude vždy nacházet seznam všech typů a vyhledávací pole, na základě kterého se bude seznam filtrovat dle názvu typu. Jelikož u materiálů a procesů jsou jednotlivé typy řazeny navíc do kategorií, bude u nich v této části okna umožněna filtrace typů nejen dle názvu ale také dle kategorie typu.

Druhá část okna (zeleně vyznačena) slouží pro prezentaci zvoleného typu z části první. Jsou zde detailní informace o daném typu a dále jsou zde zadávací pole pro různé

atributy nově vytvářeného nákladu, které vyplní uživatel a mezi kterými jsou atributy nezbytné pro výpočet ceny. Jak je zřejmé, informace obsažené v této části budou zcela záviset na volbě typu v části druhé, z čehož vyplývá, že každá změna výběru typu v první části, automaticky vyvolá aktualizaci obsahu v části druhé.

The screenshot shows a software window titled "Add Cost Window". It is split into two panes. The left pane, labeled with a circled '1', features a search bar at the top and a list titled "List of Costs" containing four items: "Cost Type 1", "Cost Type 2", "Cost Type 3" (which is selected and highlighted in blue), and "Cost Type 4". The right pane, labeled with a circled '2', shows the details for the selected "Cost Type 3". It includes a "Name" field with the value "Cost Type 3" and a "Details" field with "...". Below this is a "Calculation of Cost" section with two rows: "Attribute: 10 units" and "Attribute: 2 units". At the bottom of this section, it shows "Unit Price: \$ 8.00" and "Sub Total Price: \$ 16.00". A blue "Add Cost" button is positioned at the bottom right of the right pane.

Obrázek č. 2.2: Okno pro zadávání nákladu položce

### Formulář pro zadávání materiálu

Formulář pro zadávání materiálu položce se tedy musí skládat ze dvou stěžejních komponent, kdy v tomto případě se bude v první části nacházet i filtrování typů materiálu dle kategorie. V této části okna se bude tedy zobrazovat:

- Přehled kategorií materiálu, umožňující výběr jednotlivých kategorií
- Zadávací pole pro filtrování typů materiálu dle jejich názvů
- Přehled jednotlivých typů materiálů patřící do vybraných kategorií a obsahující v názvu řetězec ze zadávacího pole

Druhá část, okna vyhrazená pro konkrétní, materiál bude obsahovat následující informace o vybraném typu materiálu:

- Název materiálu
- Kategorii materiálu
- Vzorec pro výpočet jednotkové ceny materiálu (pokud byl zadán)
- Komentář k materiálu

Dále se v této komponentě nachází zadávací pole pro vyplnění atributů materiálu, především těch, které slouží k výpočtu ceny materiálu. Oblast se zadávanými poli se musí aktualizovat při každé změně zvoleného typu materiálu. Ceny typů materiálu



se nepočítají vždy stejným způsobem (viz kapitola 2.1.2), tudíž každý typ materiálu může vyžadovat vyplnění jiných atributů pro výpočet jeho ceny.

Atributy, které se mohou zobrazit pro vyplnění informací o zadávaném materiálu:

<i>Use</i>	použití materiálu
<i>Size 1</i>	hodnota vstupující do výpočtu ceny + zobrazená měrná jednotka
<i>Unit 1</i>	
<i>Size 2</i>	hodnota vstupující do výpočtu ceny + zobrazená měrná jednotka
<i>Unit 2</i>	
<i>Quantity</i>	kvantita materiálu
<i>Area Name</i>	název plochy
<i>Area</i>	velikost plochy
<i>Length</i>	délka
<i>Density</i>	hustota

Po vyplnění všech potřebných atributů se automaticky zobrazí vypočtená jednotková i celková cena materiálu.

### **Formulář pro zadávání spojovacího materiálu**

V první části okna formuláře pro zadání materiálu bude zobrazeno následující:

- Přehled jednotlivých typů spojovacího materiálu
- Vyhledávací pole pro filtrování typů spojovacího materiálu dle názvu

Druhá část okna slouží pro zvolený typ spojovacího materiálu z první komponenty okna. Hodnoty zobrazené pro vybraný typ spojovacího materiálu:

- Název spojovacího materiálu
- Vzorec pro výpočet jednotkové ceny spojovacího materiálu (pokud byl zadán)
- Komentář ke spojovému materiálu

Dále se v této komponentě nachází zadávací pole pro vyplnění atributů spojovacího materiálu. Oblast se zadávanými poli se musí aktualizovat při každé změně zvoleného typu spojovacího materiálu. I zde každý typ spojovacího materiálu může vyžadovat k vyplnění jiné atributy pro výpočet jeho ceny (viz kapitola 2.1.2).

Atributy, které se mohou zobrazit pro vyplnění informací o zadávaném spojovacím materiálu:

<i>Use</i>	užití spojovacího materiálu
<i>Size 1</i>	hodnota vstupující do výpočtu ceny + zobrazená měrná jednotka <i>Unit 1</i>
<i>Size 2</i>	hodnota vstupující do výpočtu ceny + zobrazená měrná jednotka <i>Unit 2</i>
<i>Quantity</i>	kvantita spojovacího materiálu

## Formulář pro zadávání procesů

První komponenta formuláře pro zadávání procesů se skládá ze stejných částí jako u materiálu:

- Přehled kategorií typů procesu, umožňující výběr jednotlivých kategorií
- Zadávací pole pro filtrování typů procesů dle jejich názvů
- Přehled typů procesů filtrovaný dle vybraných kategorií a názvu

Hodnoty zobrazené pro vybraný typ procesu v druhé části okna:

- Název procesu
- Kategorie procesu
- Zda vybraný typ procesu vyžaduje zadat opracování
- Typ multiplikátoru, který lze pro tento typ procesu použít
- Komentář k procesu

Dále se v této komponentě nachází zadávací pole pro vyplnění atributů procesu, především těch, které slouží k výpočtu ceny. Atributy, které se mohou zobrazit pro vyplnění:

<i>Use</i>	užití procesu
<i>Quantity</i>	kvantita procesu
<i>Multiplier</i>	rozbalovací menu s výčtem multiplikátorů, který koresponduje s typem multiplikátoru, který lze pro typ procesu použít

Oblast se specifikací atributů procesu se musí aktualizovat s každou změnou typu procesu. Což se projevuje především viditelností rozbalovacího menu, která závisí na tom, zda je u typu procesu typ multiplikátoru definován či nikoli.

## Formulář pro zadávání opracování

V první části okna formuláře pro zadávání opracování bude zobrazeno následující:

- Přehled jednotlivých typů opracování
- Zadávací pole pro filtrování typů opracování dle názvu

Pro vybraný typ obrábění bude vždy v druhé části okna zobrazen název a komentář k typu obrábění.

Dále se v této komponentě nachází zadávací pole pro vyplnění atributů opracování, sloužící především k výpočtu ceny. Možné atributy zobrazující pro vyplnění informací o zadávaném opracování:

<i>Use</i>	užití opracování
<i>Quantity</i>	kvantita opracování
<i>PVF</i>	faktor objemu výroby (PVF - Production Volume Factor)
<i>FracIncl</i>	počet dílů, pro které se používá to samé opracování

## 2.1.4 Export zprávy o nákladech

Tato zpráva bude exportována v XLS souborech a to ve stanoveném formátu. Exportována bude nejprve sada souborů, kdy každý soubor obsahuje informace o jedné konkrétní sestavě a všech jejích částech. Tyto soubory jsou exportovány ve formátu, který lze vidět na přiloženém CD v souboru „*FCA\_Inputs.xls*“ (v adresáři „*attachment*“), kde na listu „*assembly*“ je zobrazen vzor listu pro sestavu, a na listu „*part*“ je zobrazen vzor listu pro dílčí část sestavy.

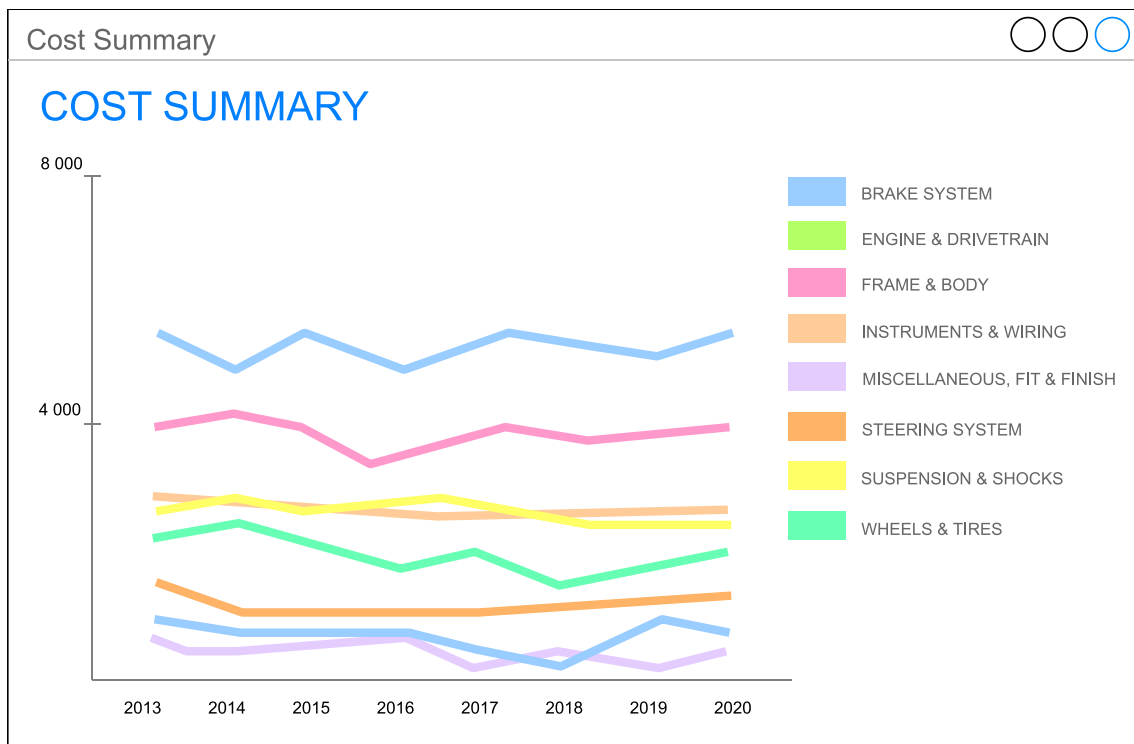
Na konec bude exportován soubor, ve kterém jsou obsaženy souhrnné informace za celé vozidlo. Formát tohoto souboru je zachycen na přiloženém CD v souboru „*FSAE\_eBOM.xls*“. Pro každou oblast komodit se zde nachází sumy nákladů za materiál, procesy, obrábění a spojový materiál. Součástí souboru je i detailní tabulka, která obsahuje výpis všech konkrétních sestav se sumačními částkami za materiál, procesy, obrábění a spojový materiál.

## 2.2 Oblast aplikace pro historická data

Tato část aplikace bude sloužit především pro vizualizaci nákladů vozidla v průběhu let a to v tabulkách a grafech, které si následovně probereme.

### 2.2.1 Porovnání v průběhu let

Aplikace bude graficky znázorňovat vývoj nákladů všech osmi hlavních oblastí komodit (viz kapitola 2.1.2). Z důvodu graficky zřetelného vývoje trendu (růst, pokles) a přehledné vizualizace byl pro vykreslení dat zvolen liniový graf, kdy dílčí linie grafu budou odpovídat nákladům jednotlivých oblastí komodit (viz Obrázek č. 2.3).



Obrázek č. 2.3: Okno pro porovnání oblastí vozidla v průběhu let

### 2.2.2 Porovnání v rámci konkrétní oblasti komodit

Na následujícím Obrázku č. 2.4 je návrh okna pro porovnání nákladů v oblasti komodit, který byl se zadavatelem vytvořen. V tomto konkrétním případě se jedná o data v rámci oblasti „Brake System“. Data této oblasti jsou zobrazena z dvou úhlů pohledu. Jednotlivé položky nákladů jsou v řádcích tříděny na základě kategorií či podrobněji dle typu. Dále v sloupcích jsou rozříděny podle let, tedy dle konkrétních vozidel, které byly v každém roce vytvořeny. Jak lze vidět na návrhu, konkrétní typy spadající do kategorie jsou viditelné po rozkliknutí dané kategorie. V takovéto kategorii budou dílčí typy dále řazeny dle velikosti ukazatele (od největšího).

Mezi základní ukazatele, které jsou zadavatelem požadovány, patří cena v dolarech a počet. Tyto ukazatele bude moci uživatel v rámci tohoto okna měnit (viz pravý horní roh návrhu).

Brake System Overview					
BRAKE SYSTEM					
unit of measure: \$					
	2015	2016	2017	2018	2019
<b>Material</b>	<b>730.25</b>	<b>686.86</b>	<b>713.58</b>	<b>690.80</b>	<b>705.25</b>
Washer, Grade 8.8 (SAE 5)	330.30	330.80	310.23	310.20	310.30
Steel, Stainless (per kg)	230.20	215.36	225.45	234.24	233.78
Brake Caliper, ISR, 22-048	120.80	88.95	123.54	95.93	111.46
Brake Caliper, ISR, 22-049	38.45	42.30	40.13	39.20	39.56
Brake Pad, Iron or Steel Rotor	10.50	9.45	14.23	11.23	10.15
<b>Fasteners</b>	<b>15.90</b>	<b>13.67</b>	<b>9.16</b>	<b>8.89</b>	<b>8.90</b>
<b>Processes</b>	<b>108.60</b>	<b>105.65</b>	<b>102.62</b>	<b>104.32</b>	<b>101.34</b>
Wrench <= 25.4 mm	41.90	40.80	41.42	43.11	22.43
Laser cut	37.50	36.40	35.85	34.65	4.50
Assemble, 1 kg, Interference	15.00	14.20	14.35	14.24	4.20
Heat Treatment	6.50	6.75	4.50	4.82	0.50
Machining Setup and Install	4.60	4.45	1.57	3.45	0.40
Safety Wire, Install	3.10	3.05	3.03	3.05	0.05
<b>Tooling</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>-</b>	<b>-</b>

Obrázek č. 2.4: Návrh okna pro porovnání nákladů v průběhu let u oblasti komodit

V datech bude uchována informace o konkrétních sestavách, aby uživatel mohl dohledat takové sestavy, které jsou nejvíce nákladné. Na základě takové informace pak uživatel zjistí, na kterou konkrétní část vozidla se zaměřit a hledat zde případnou redukci či záměnu materiálu za cílem snížení nákladů.

Na Obrázku č. 2.5 je ukázán další návrh okna, který byl opět vytvořen spolu se zadavatelem. Zde je okno řešeno podobným způsobem, jako to bylo v předchozím návrhu. Na tomto příkladu jsou data z oblasti „Brake System“, dělena v řádcích na základě kategorií, ale ve sloupcích jsou data rozdělena dle výše zmíněných sestav. Data v tomto okně se budou vztahovat vždy ke konkrétnímu roku (vozidlu), který bude moci uživatel regulovat v horní části okna. Zde uživateli bude opět umožněno měnit i aktuálně se zobrazující ukazatel.

Brake System Overview ○ ○ ○

**BRAKE SYSTEM** for year:  unit of measure:

	Brake Discs front	Brake Discs rear	Brake Caliper front	Brake Caliper rear	Brake lines
<b>Material</b>	5	4	5	5	1
Washer, Grade 8.8 (SAE 5)	2	2	2	2	1
Steel, Stainless (per kg)	3	-	-	-	-
Brake Caliper, ISR, 22-048	-	2	-	-	-
Brake Caliper, ISR, 22-049	-	-	2	2	-
Brake Pad, Iron or Steel Rotor	-	-	1	1	-
<b>Fasteners</b>	5	7	8	15	12
<b>Processes</b>	8	10	7	4	6
Wrench <= 25.4 mm	2	2	2	2	2
Laser cut	2	4	4	1	4
Assemble, 1 kg, Interference	2	-	1	-	-
Heat Treatment	-	4	-	-	-
Machining Setup and Install	2	-	-	-	-
Safety Wire, Install	-	-	-	1	-
<b>Tooling</b>	-	-	-	-	-

Obrázek č. 2.5: Okno pro porovnání nákladů mezi sestavami v rámci oblasti komodit

### 2.2.3 Informace konkrétních sestav

Aplikace bude umožňovat zobrazení detailu konkrétní sestavy. Budou zde informace o konkrétních nákladech, které byly v rámci sestavy vynaloženy. Tyto náklady budou rozlišeny dle dílčích částí sestavy.

Na Obrázku č. 2.6 je návrh okna konkrétní sestavy. Zde jsou pro příklad vypsány detaily o sestavě „*Brake Discs Front*“. Okno je řešeno obdobně jako na dvou předchozích návrzích. Jsou zde tedy opět data dělena v řádcích dle kategorií. První sloupec „*Total*“ obsahuje celkový součet za sestavu včetně jejich částí. Ve druhém sloupci „*Assembly*“ jsou obsaženy náklady, které byly vynaloženy pro spojení dílčích částí v celkovou sestavu. V dalších sloupcích jsou již pak náklady za konkrétní části sestavy.

Brake System - A1001 Overview

**BRAKE DISCS FRONT**  
A1001

for year: 2017 unit of measure: pcs

	TOTAL	ASSEMBLY
<b>Material</b>	7,99	6
Washer, Grade 8.8 (SAE 5)	6	6
Steel, Stainless (per kg)	1,97	-
Steel, Alloy (per kg)	0,02	-
<b>Fasteners</b>	6	6
<b>Processes</b>	425,4	38
Hand - Start Only	19	19
Laser cut	401	19
Assemble, 1 kg, Interference	2	-
Machining Setup and Install	2	-
Machining	1,4	-
<b>Tooling</b>	-	-

ASSEMBLY PARTS	
Disks	Flow washer
0,97	1,02
-	-
0,97	1
-	0,02
-	-
385	2,4
-	-
382	-
2	-
1	1
-	1,4
-	-

Obrázek č. 2.6: Okno s detaily konkrétní sestavy

## 2.2.4 Porovnání v rámci konkrétního roku

Aplikace musí umožňovat porovnání nákladů v rámci konkrétního roku. Bude se jednat o hlavní přehledovou tabulku pro daný rok. Náklady budou v takovém přehledu členěny dle hlavních oblastí komodit a dle nákladových kategorií.

2017 Cost Summary

**COST SUMMARY 2017**

for year: 2017

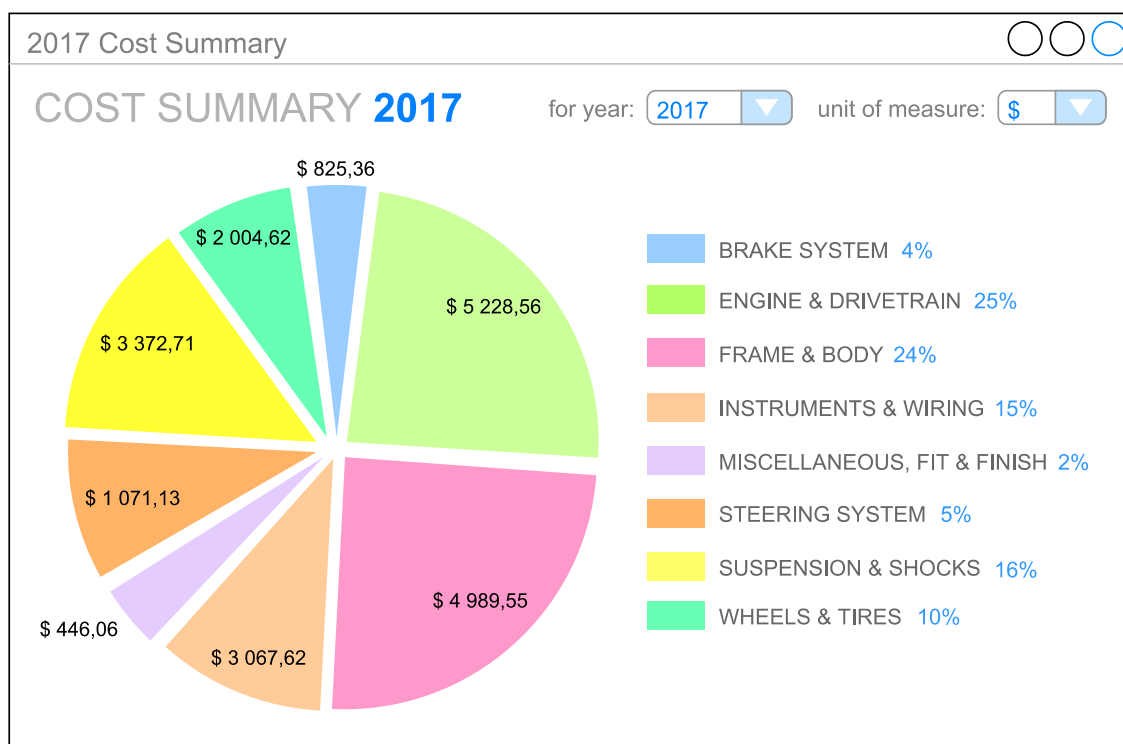
	MATERIALS	PROCESSES	FASTENERS	TOOLING	
<b>BRAKE SYSTEM</b>	\$ 713,58	\$ 102,62	\$ 9,16	-	\$ 825,36
<b>ENGINE &amp; DRIVETRAIN</b>	\$ 2 607,74	\$ 1 174,77	\$ 40,32	\$ 20,79	\$ 5 228,56
<b>FRAME &amp; BODY</b>	\$ 3 978,58	\$ 366,58	\$ 91,66	\$ 713,58	\$ 4 989,55
<b>INSTRUMENTS &amp; WIRING</b>	\$ 2 668,85	\$ 366,83	\$ 3,64	\$ 0,67	\$ 3 067,62
<b>MISCELLANEOUS, FIT &amp; FINISH</b>	\$ 309,58	\$ 132,44	\$ 2,11	\$ 2,34	\$ 446,06
<b>STEERING SYSTEM</b>	\$ 415,15	\$ 380,13	\$ 4,75	\$ 0,93	\$ 1 071,13
<b>SUSPENSION &amp; SHOCKS</b>	\$ 2 240,88	\$ 1 007,18	\$ 7,24	-	\$ 3 372,71
<b>WHEELS &amp; TIRES</b>	\$ 1 125,25	\$ 622,86	\$ 16,48	-	\$ 2 004,62
	\$ 14 079,98	\$ 5 634,80	\$ 175,36	\$ 96,70	\$ 21 005,59

Obrázek č. 2.7: Nástin okna - Přehled nákladů za konkrétní rok (tabulka)

Na Obrázku č. 2.7 je zobrazen nástin okna, které by sloužilo pro přehled nákladů konkrétního roku. Data jsou zde vizualizovaná pomocí tabulky, kde sloupce odpovídají

kategoriím nákladů a řádky odpovídají dílčím oblastem komodit vozidla. V horní části okna bude uživateli umožněno měnit rok, za který budou data zobrazována.

Dále na Obrázku č. 2.8 je okno, které zobrazuje stejná data jako na předchozím obrázku, ale tentokrát pomocí výsečového grafu, kdy každá výseč reprezentuje podíl jedné oblasti komodit vozidla.

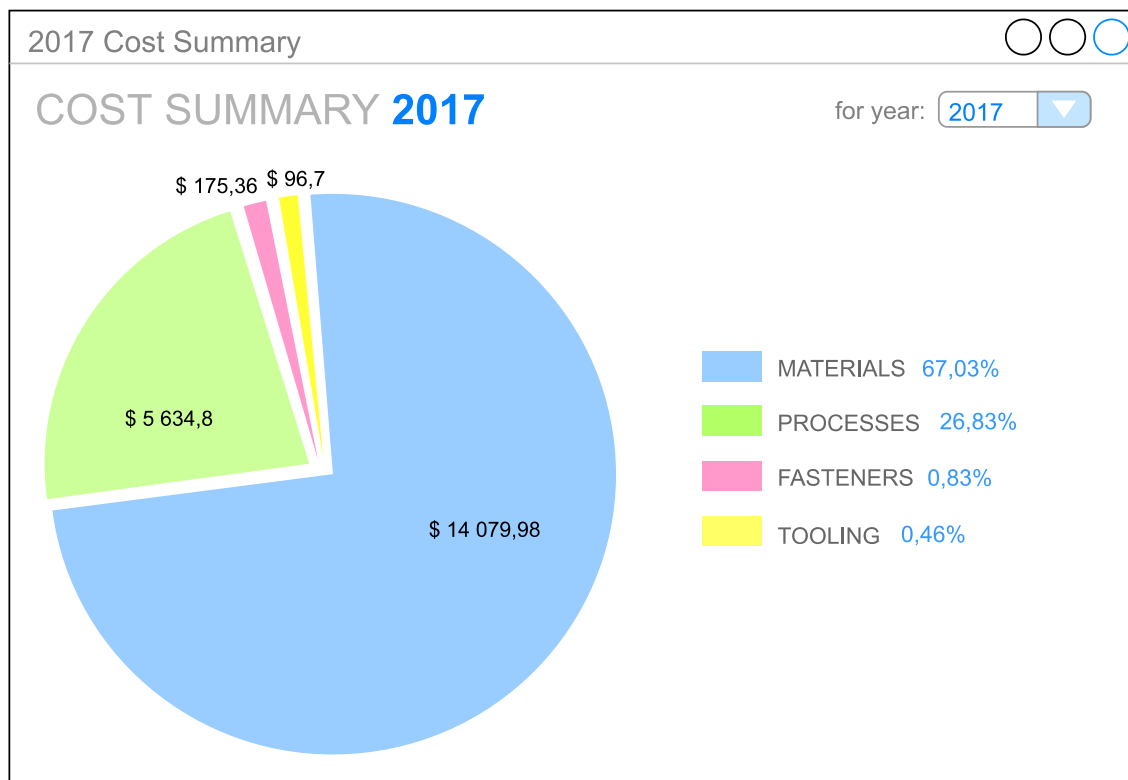


Obrázek č. 2.8: Nástin okna - Přehled nákladů za konkrétní rok (graf)

Eventuelně je zde možnost umožnit vyšší úroveň podrobnosti a vizualizovat i podíl jednotlivých sestav, na které se hlavní oblasti vozidla dále dělí. Nejedná se však o nezbytnou funkcionalitu aplikace.

Ve stejném duchu je i řešeno následující okno (Obrázek č. 2.9), kde jsou ta samá data vizualizovaná opět pomocí výsečového grafu. V tomto případě jsou náklady členěny nikoli dle oblasti komodit ale dle dílčích nákladových kategorií.





Obrázek č. 2.9: Nástin okna – Přehled nákladů, členění dle kategorie (graf)

## 3 Business Intelligence

Cílem této práce je podpora studentského týmu v rozhodování na strategické úrovni a právě Business Intelligence je nedílnou součástí této oblasti.

Business Intelligence (BI) je pojem, který označuje komplex činností, technologií a úloh, které tvoří běžnou složku řízení podniků a jejich informačních systémů. Tento komplex je využíván jako podpora rozhodovacích procesů v organizaci. [3]

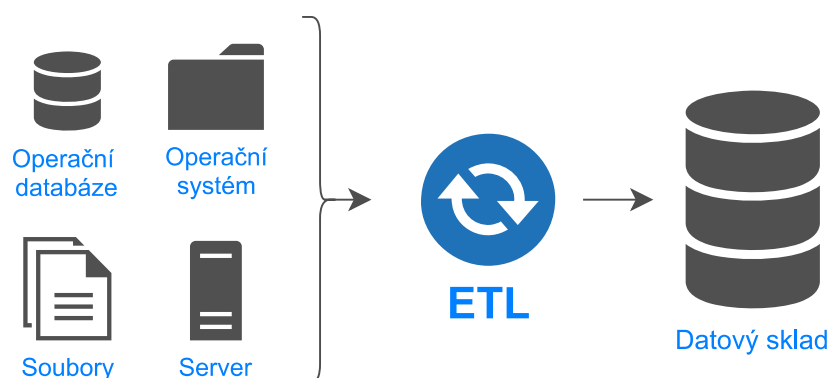
Architektura či obecné koncepce BI se neustále výrazně mění dle situace nebo potřeb daného zákazníka či podniku. Konkrétní řešení architektur BI se liší především díky rozmanitosti a kombinaci samotných nástrojů. Za dobu vývoje v této oblasti se však obecná koncepce BI ustálila do několika stěžejních vrstev [3].

### 3.1 Vrstva extrakce, transformace a nahrávání dat

Tato vrstva pokrývá sběr či přenos dat ze zdrojových systémů do vrstvy pro ukládání dat. Mezi komponenty této vrstvy patří ETL systémy, neboli systémy pro extrakci, transformaci a přenos dat, a EAI systémy, tedy systém pro integraci aplikací.

#### 3.1.1 ETL

ETL (Extraction, Transformation and Loading) je obecně proces získání (kopírování) dat z jednoho nebo více zdrojů do cílového systému, který pak na tyto data poskytuje jednotný pohled. Tento proces je běžně známý i jako datová pumpa. Úkolem tohoto procesu je data ze zdrojových systémů získat a vybrat (Extraction), vyčistit a upravit do požadované formy (Transformation) a nahrát je do datových schémat, jako je například datový sklad (Loading), což je naznačeno na Obrázku č. 3.1 [3].



Obrázek č. 3.1: Koncept ETL procesu

Metodika a úkoly ETL jsou již mnoho let dobře známy a nejsou nutně využívány pouze v oblasti datového skladu. Kromě již zmíněných stěžejních třech kroků ETL zahrnuje mnoho dalších podprocesů. Mezi ně patří například kontrola kvality dat (hledání anomálií a nekonzistentností v datech), auditování či bezpečnost, zálohování či obnova dat.

### 3.1.2 EAI

Cílem EAI (Enterprise Application Integration) je integrovat primární podnikové systémy a razantně redukovat počet jejich vzájemných rozhraní [3]. V Business Intelligence je EAI využito zejména pro přenos dat do datových úložišť. Oproti nástrojům ETL pracují EAI platformy v reálném čase a tím tak umožňují vznik nové generace datových skladů, tzv. Real-Time Data Warehouse.

## 3.2 Vrstva pro ukládání dat

Zajišťuje procesy ukládání, aktualizace a správy dat pro řešení BI. Do této vrstvy patří následující databázové komponenty:

- Datové sklady (Data Warehouse) – základní databázová komponenta
- Datová tržiště (Data Marts) – součást či nadstavba datového skladu
- Operační datová úložiště (Operational Data Store) – podpurné analytické databáze
- Dočasná datová úložiště (Data Staging Areas) – slouží pro dočasné uložení dat před jejich vlastním zpracováním

### 3.2.1 Datový sklad

Co je tedy datový sklad? Datový sklad je kolekcí sjednocených, předmětově orientovaných databází navržených za účelem poskytovat informace požadované pro rozhodování [4]. Na tyto vlastnosti se podíváme podrobněji.

Význam sjednocení se skrývá především ve spojení dat z mnoha různorodých provozních systémů, jedná se tedy například o sjednocení dat v oblasti doménových modelů. Dále umožňuje datový sklad uchovávat data na různých stupních podrobnosti, tedy jak data atomická, tak i data sumarizovaná. Datový sklad je koncipován jako „Read Only“, což znamená, že zde žádná data nevznikají ručním pořízením, a nelze je ani žádnými uživatelskými nástroji editovat [3]. Data jsou sem nahrávána z operačních databází či jiných externích zdrojů a existují zde po celou dobu života datového skladu.

Datový sklad je významným nástrojem pro podporu rozhodování, a to zejména na strategické úrovni podniku.

### Tvorba datového skladu

Pro budování datového skladu a všech souvisejících komponent existují dva základní přístupy. Prvním z nich je tzv. metoda velkého třesku, která spočívá v jednorázovém vybudování datového skladu a s ním i všech jeho potřebných komponent. V rámci tohoto přístupu je nejprve provedena celková analýza a dokumentace uživatelských potřeb, návrh celkového řešení a poté jeho implementace. [3]

Druhým přístupem je tzv. přírůstková metoda. Zde se jedná o postupné a průběžné budování datového skladu rozdělené do jednotlivých etap. Na začátku je sestaven návrh

celého řešení včetně harmonogramu jednotlivých časově a finančně omezených kroků (přírůstků), během kterých pak dochází k postupné implementaci řešení [3]. Největší výhodou tohoto přístupu na rozdíl od předchozího je především možnost reagovat na případné změny požadavků uživatelů v průběhu budování datového skladu. Na druhou stranu tato metoda bývá velmi časově náročná, a to zejména na začátku projektu při tvorbě celkové koncepce.

Všeobecně lze říct, že neexistuje univerzální nejvhodnější přístup pro budování datového skladu. Výběr dané metody bude vždy záviset jak na aktuální architektuře celopodnikového informačního systému, tak i na preferencích zákazníka.

### **3.2.2 Datové tržiště**

Na rozdíl od datového skladu, který uchovává velké množství dat, datové tržiště obvykle obsahuje podmnožinu dat, která může být v datovém skladu uložena [5]. Datová tržiště jsou vytvářena pro specifické cíle jednotlivých částí organizace. Například mohou existovat datová tržiště pro oddělení podniku, konkrétní divizi či geografickou lokaci části podniku. Datová tržiště jsou tedy problémově orientované datové sklady, určené pro pokrytí konkrétní problematiky daného okruhu uživatelů [3].

### **3.2.3 Operační datové úložiště**

Operační datové úložiště (ODS) neboli sklad provozních dat se v mnohém s datovým skladem neliší. Jedná se zde o sadu sjednocených databází navržených k podpoře sledování provozu. Oproti klasickým databázím, které jsou orientovány transakčně, ODS stejně jako datové sklady obsahují předmětově orientovaná celopodniková data. Nicméně na rozdíl od datových skladů, jsou data v ODS nestálá, aktuální a podrobná [5]. Sklady provozních dat tedy poskytují sjednocený pohled na data z primárních systémů podniku.

### **3.2.4 Dočasné úložiště dat**

Smyslem dočasného úložiště dat (DSA) je dočasné uložení extrahovaných dat z produkčních systémů a jeho hlavním úkolem je podpora rychlé a efektivní extrakce dat [3]. V DSA se nacházejí neagregovaná data, která neobsahují historii (přenášejí se pouze aktuální data) a jsou ve stejné struktuře, v jaké jsou uložena ve zdrojových systémech.

## **3.3 Vrstva pro analýzu dat**

Zde jsou pokryty činnosti spojené s vlastním zpřístupněním dat a analýzou dat. V této vrstvě se nacházejí následující analytické komponenty:

- Systémy OLAP
- Reporting
- Dolování dat (Data Mining)

### 3.3.1 OLAP

Nyní se podíváme blíže na výše jmenovaný OLAP systém. Aplikace pro podporu rozhodování neboli OLAP (Online Analytical Processing) aplikace poskytují manažerům pohledy na historická a aktuální podniková data v uživatelsky přívětivých formátech jako jsou například grafy a diagramy. Často jsou používány pro především rychlé vyjádření významných vztahů či souvislostí mezi velkým množstvím dat. [5]

Blíže si představíme OLAP technologii prostřednictvím tzv. „dvanáctera“ **původních pravidel OLAP** od Dr. E. F. Codda [6]:

#### 1. Multidimenzionální konceptuální pohled

OLAP by měl poskytovat multidimenzionální model pro podporu potřeb uživatele.

#### 2. Transparentnost

Technologie systému OLAP (architektura výpočtů, podřízená databáze) by měly být uživateli transparentní, aby mohl plně využít veškerý potenciál této technologie.

#### 3. Dostupnost

OLAP by měl přistupovat ke všem údajům, které jsou pro analýzu dat nezbytná.

#### 4. Konzistentní vykazování

I s postupným růstem počtu záznamů by uživatel neměl pocítit výrazné snížení výkonu systému.

#### 5. Architektura klient-server

Systém OLAP by měl odpovídat architektuře klient-server.

#### 6. Generická dimenzionalita

Každá jednotlivá dimenze, ze které je na údaje pohlíženo, musí být ekvivalentní ve struktuře i operačních schopnostech.

#### 7. Dynamické ošetření řídkých matic

OLAP by měl být optimalizován pro ošetření řídkých matic

#### 8. Podpora pro více uživatelů

#### 9. Neomezené křížové dimenzionální operace

OLAP by měl být schopen rozeznat hierarchie a automaticky vykonat asociované kumulované propočty jak v rámci dimenzí i mezi nimi.

#### 10. Intuitivní manipulace s údaji

#### 11. Flexibilní vykazování

Musí v těchto technologiích existovat schopnost uspořádání dat takovým způsobem, který umožní analýzu a intuitivní vizuální prezentaci analytických ukazatelů uživateli.

#### 12. Neomezené dimenze a úrovně agregace

OLAP by neměl zavádět žádné omezení co se počtu dimenzí či úrovní agregace týče.

Existuje několik typů OLAP systémů, mezi které například patří:

- **ROLAP** (Relational Online Analytical Processing)
- **MOLAP** (Multidimensional Online Analytical Processing)
- **HOLAP** (Hybrid Online Analytical Processing)
- **DOLAP** (Desktop Online Analytical Processing)
- **WOLAP** (Web Online Analytical Processing)
- **Mobile OLAP** (Mobile Online Analytical Processing)
- **SOLAP** (Spatial Online Analytical Processing)

Blíže si popíšeme tři nejčastější druhy této technologie, tedy ROLAP, MOLAP a HOLAP, které velice úzce souvisí s modelem datového skladu.

## **ROLAP**

Zde hovoříme o relačním OLAP, jelikož tato technologie pracuje s RDBMS (Relational Database Management System). Data jsou zde uložena v relační databázi, kde jsou vytvořeny tabulky pro uložení agregací. Ačkoli je zde výhodnou lepší škálovatelnost či efektivnější práce s neagregovanými údaji oproti MOLAP, analýza dat je zde náročná z důvodu nutnosti užívání SQL příkazů pro načtení dat přes vícero tabulek spojenými relacemi. Přesto je tento přístup jednodušší a snadnější na používání než MOLAP [5].

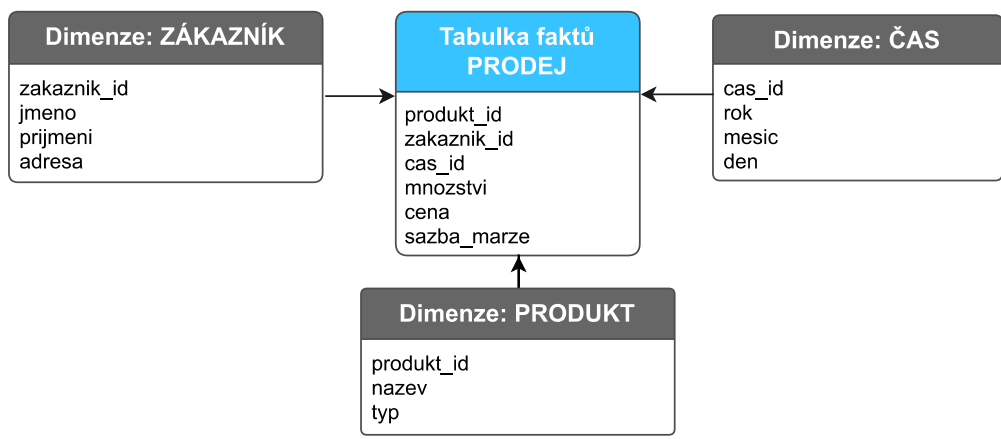
V tomto případě je datový sklad modelován pomocí tzv. tabulek faktů a dimenzí (viz Obrázek č. 3.2). V tabulce faktů se nacházejí logicky související hodnoty sledovaných ukazatelů [3]. Mohou se zde nacházet metriky jako je například prodejní cena v Kč, počet kusů či náklady v Kč. Dimenzionální tabulky pak uchovávají informace o hodnotách ukazatelů v dané tabulce faktů [7].

Tabulka faktů se pak prostřednictvím cizích klíčů odkazuje na odpovídající tabulky dimenzí. Všechny tyto cizí klíče pak slouží jako složený primární klíč. Řádky v takovéto tabulce odpovídají nejnižší úrovni detailu tzv. granularita tabulky faktů. Nízká granularita znemožňuje práci s detailními daty, vysoká granularita nabízí naopak možnost detailnějších analýz.

Ukazatele v tabulkách faktů se dle možnosti agregace dělí následovně [3]:

- Aditivní – data při agregaci mají smysl pro všechny dimenze datového skladu
- Neaditivní – data v případě agregace nemají smysl pro žádnou z dimenzí
- Semiaditivní – agregované hodnoty mají smysl pouze dle určitých dimenzí

Model datového skladu, který je uveden pro příklad na Obrázku č. 3.2, sleduje informace o prodaném zboží. Mezi ukazatele patří cena a počet kusů prodaného zboží a sazba marže daného prodeje v %. Dimenzemi datového skladu jsou čas, produkt a zákazník. Cenu a počet kusů prodaného zboží lze zde sledovat napříč všemi těmito dimenzemi, tudíž se jedná o aditivní ukazatele. Oproti tomu ukazatel udávající sazbu marže nemá smysl agregovat pro žádnou z uvedených dimenzí a jedná se proto o neaditivní ukazatel.



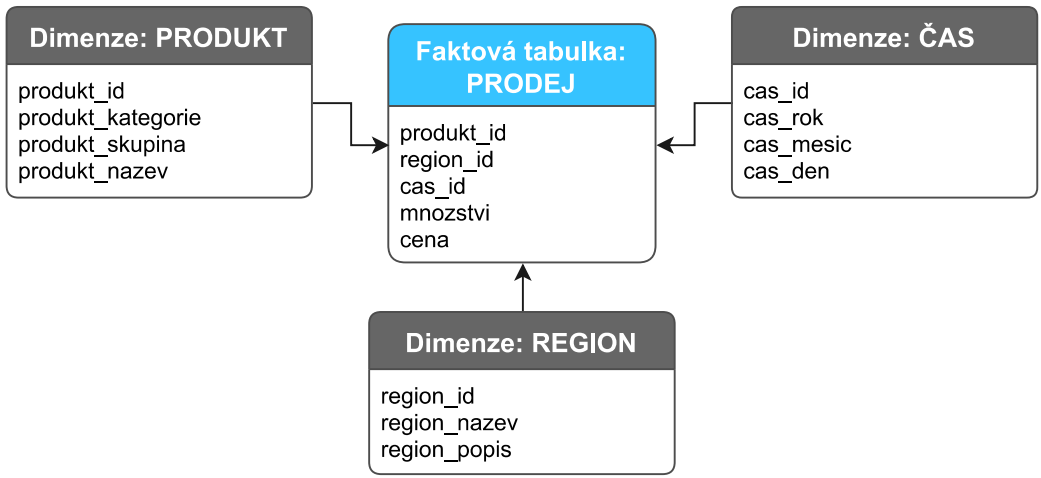
Obrázek č. 3.2: Model datového skladu (relační tabulky)

Mezi známá schémata relačních tabulek u modelování datových skladů patří Hvězdicové schéma (Star Schema), schéma Sněžné vločky (Snowflake Schema) a schéma Souhvězdí (Constellation Schema, Galaxy Schema).

**Hvězdicové schéma**

V tomto schématu se nachází jedna tabulka faktů s množinou tabulek dimenzí. Pokud v dimenzi existuje hierarchie, jsou veškeré úrovně detailu uloženy v jedné tabulce odpovídající dané dimenzi.

Tímto ve hvězdicovém schématu dochází k denormalizaci, kdy tabulka dimenzí musí obsahovat všechny možné kombinace všech úrovní hierarchie, a tím tak dochází k duplicitě dat. To má samozřejmě za následek i vyšší paměťovou náročnost datového skladu. Protože se v tomto schématu nachází pouze relace mezi tabulkou faktů a tabulkami dimenzí, je hvězdicové schéma rychlé v době odezvy na dotazování [3].



Obrázek č. 3.3: Schéma Hvězdy

Na Obrázku č. 3.3 je zobrazen model datového skladu, který sleduje cenu a počet kusů prodaného zboží ze tří dimenzí. V dimenzích „Product“ a „Time“ lze vidět různé úrovně podrobnosti. V časové dimenzi se mohou data (fakta) sledovat přes jednotlivé roky,

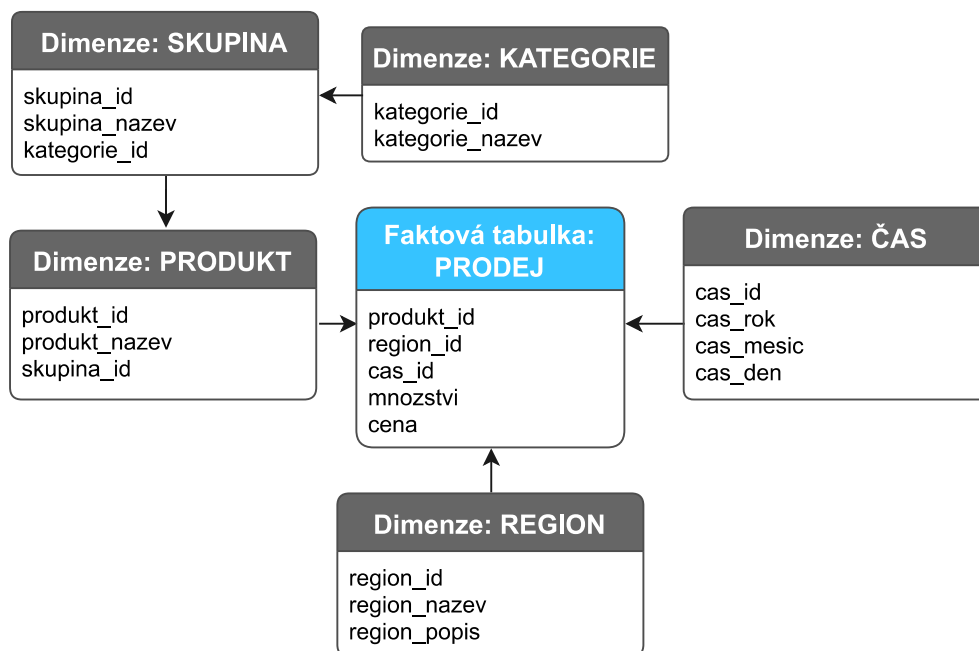
měsíce až po jednotlivé dny. Skrze dimenzi produktů lze data sledovat přes kategorie, skupiny až po konkrétní typy produktů.

### Schéma sněžné vločky

V tomto schématu se opět nachází jedna tabulka faktů s množinou tabulek dimenzí. Zde jsou však hierarchie dimenzí řešeny odlišně. Dimenze je v takovém případě rozdělena do několika tabulek dimenzí, kdy každá odpovídá dané úrovni detailu.

Zde tedy dochází naopak k normalizaci, konkrétně jsou vytvářeny dle třetí normální formy. Díky tomu je schéma Sněžné vločky vhodné pro časté změny v dimenzích a v hierarchické struktuře jejich prvků [3]. Toto samozřejmě vede i k úspoře místa v datovém skladu. Na rozdíl od Hvězdicového schématu je toto schéma složitější, komplexnější a tedy i časově náročnější při dotazování nad datovým skladem [6].

Na Obrázku č. 3.4 se nachází stejný model datového skladu jako na předchozím Obrázku č. 3.3. Zde je ale hierarchie u dimenze produktů řešena odlišně. Vznikly nám zde tři dimenzionální tabulky, které jsou na sebe napojeny od nejnižší úrovně podrobnosti po největší.



Obrázek č. 3.4: Schéma Sněžné vločky

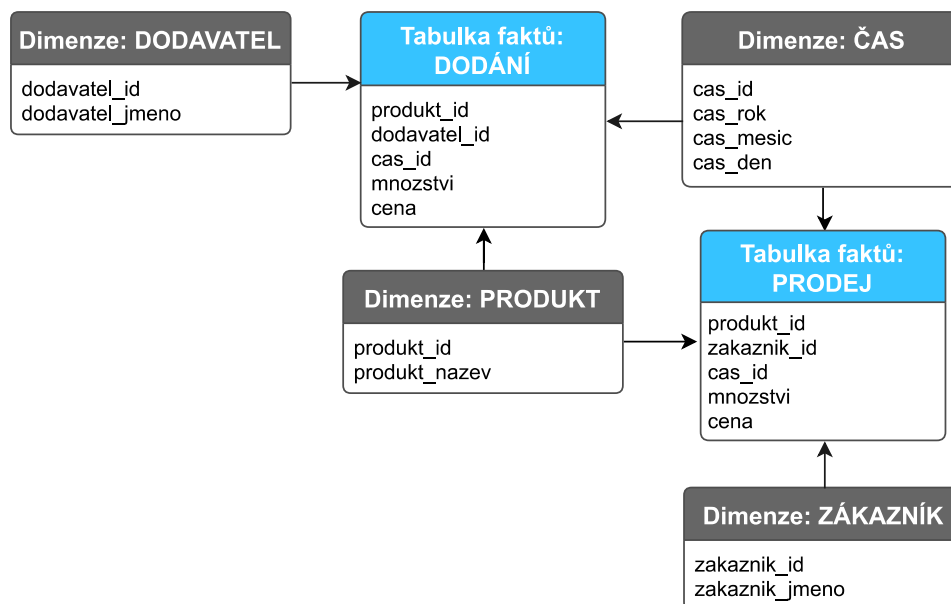
### Schéma souhvězdí

V tomto schématu je obsaženo několik hvězdicových schémat najednou, je zde tedy více než jedna tabulka faktů. Běžně je takové schéma nazýváno i jako Galaxie. Tabulky faktů jsou zde propojeny pouze skrz tabulky dimenzí, které mají společné.

Hlavním nedostatkem je komplikovanější návrh, protože je třeba zvážit a vybrat mnoho variant pro konkrétní druhy agregace.



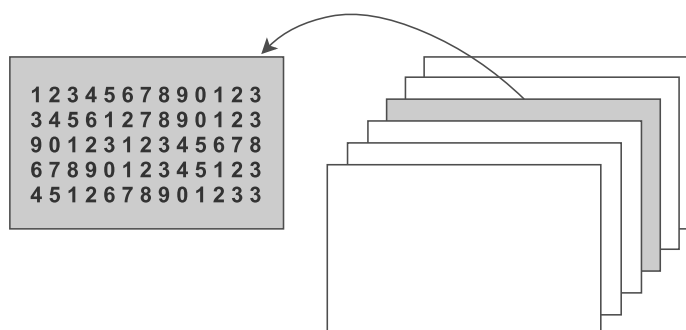
Na Obrázku č. 3.5 je příklad schématu Souhvězdí, kde je zobrazen datový sklad sledující dodání zboží od dodavatelů a prodej zboží zákazníkům. Je zde tedy tabulka faktů pro dodání zboží a tabulka faktů pro prodej zboží. Obě tyto tabulky mají společnou dimenzi času a dimenzi produktů, skrze které jsou také propojeny.



Obrázek č. 3.5: Schéma Souhvězdí

## MOLAP

V tomto případě se hovoří o multidimenzionální OLAP analýze. Místo modelování datového skladu pomocí tabulek se zde používají multidimenzionální databáze. Data se ukládají v tzv. hyperkostkách, které jsou dle potřeby stránkovány do a z paměti (viz Obrázek č. 3.6) [5]. Hyperkostku si lze představit, jako m-dimenzionální tabulku, kdy každá dimenze odpovídá jednomu rozměru kostky. V jednotlivých buňkách kostky jsou uloženy n-tice faktů, nad kterými budou probíhat různé agregace z daných dimenzí.



Obrázek č. 3.6: Datové struktury multidimenzionálních databází (zdroj [5])

Při tomto způsobu uložení dat není cílem minimalizovat redundanci, ale urychlit průběh analýzy nad daty, což je významnou výhodou MOLAP. Tento způsob ovšem přináší i nevýhody. Především v podobě zátěže paměťového prostoru a zvýšení časové náročnosti při plnění skladu, která je způsobena pracnější transformací dat.

## HOLAP

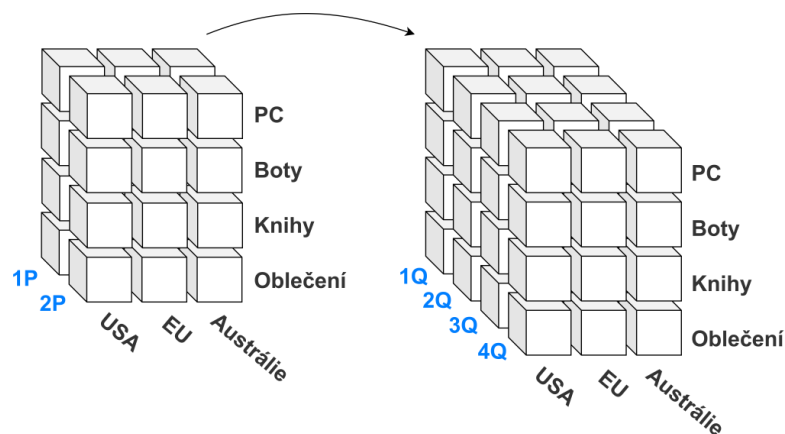
HOLAP je specifický přístup, který je výsledkem snahy o začlenění nejlepších vlastností MOLAP a ROLAP do jediné architektury. Agregované součty jsou ukládány do multidimenzionálních databází (MOLAP), zatímco podrobná data jsou uložena v relační databázi (ROLAP). To nabízí jak datovou efektivitu modelu ROLAP, tak výkon modelu MOLAP. Mezi výhody tohoto systému patří lepší škálovatelnost, rychlé zpracování dat a flexibilita v přístupu ke zdrojům dat. Na druhou stranu toto řešení vyžaduje uchovávat data na dvou místech najednou.

## Základní OLAP operace

Následující operace zajišťují zobrazení dat v různých agregacích, což je přímo závislé na zvolení dimenzí, případně na zvolení její úrovně detailu. Každá operace bude znázorněna na tzv. OLAP datových kostkách, kterými je znázorněn multidimenzionální pohled na data.

- **Drill-down**

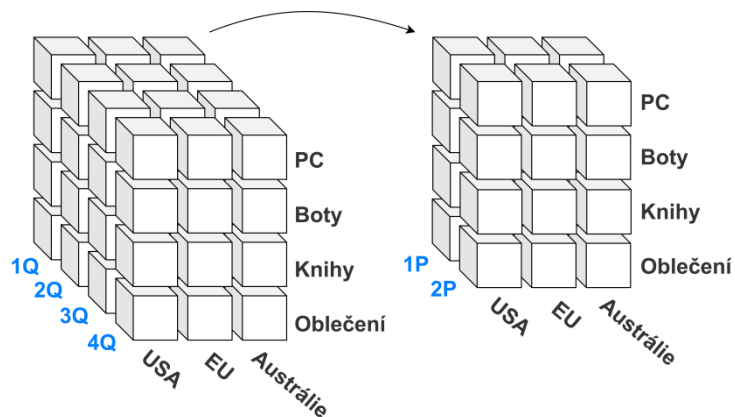
Data jsou zde fragmentována do menších (podrobnějších) částí. Tedy například při sledování zisku za jednotlivá pololetí zvýšíme úroveň detailu a tím vznikne pohled na data, kde budou zisky za jednotlivé kvartály (viz Obrázek č. 3.7).



Obrázek č. 3.7: OLAP kostka, Drill-down

- **Roll-up**

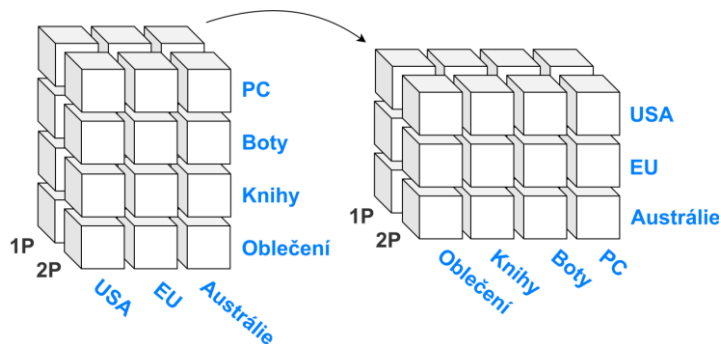
Je opakem předešlé operace, kdy data jsou zobrazena na vyšší agregační úrovni. Úroveň detailu tedy se tedy sníží (viz Obrázek č. 3.8).



Obrázek č. 3.8: OLAP kostka, Roll-up

- **Pivoting**

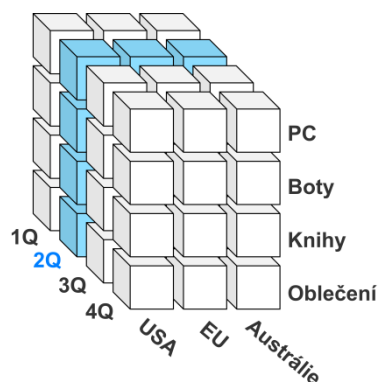
Jedná o zobrazení stejných dat ale skrze zaměněné dimenze (viz Obrázek č. 3.9).



Obrázek č. 3.9: OLAP kostka, Pivot

- **Slicing**

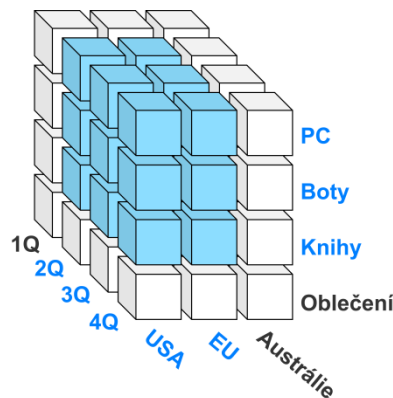
Tato operace aplikuje filtr na jednu z dimenzí. Například zobrazení zisků firmy pouze za konkrétní kvartál (viz Obrázek č. 3.10).



Obrázek č. 3.10: OLAP kostka, Slicing

- **Dicing**

Dicing operace je obdobou té předchozí, kdy je aplikován filtr na více dimenzí najednou (viz Obrázek č. 3.11).



Obrázek č. 3.11: OLAP kostka, Dicing

## 3.4 Prezentační vrstva

V prezentační vrstvě se nacházejí nástroje pro koncové uživatele, které zajišťují komunikaci koncových uživatelů s ostatními komponentami BI. Zde probíhá zejména sběr požadavků a následná prezentace výsledků. Mezi takovými nástroji patří systémy EIS (Executive Information Systems) a další analytické aplikace. Je zde možnost i portálové aplikace založené na WWW technologiích.

### 3.4.1 EIS

Manažerské aplikace EIS jsou vytvářeny pro podporu manažerských procesů, jako jsou například podnikové analýzy, plánování či rozhodování. Hlavním rysem těchto aplikací je poskytování uživatelům relevantní informace v účinné a přehledné formě. Umožňují přístup ke konkrétním datům stejně tak jako k jejich agregacím. Jsou mnohdy jednoduše ovladatelné a zajišťují vysokou vypovídající hodnotu pomocí grafických výstupů. [3]

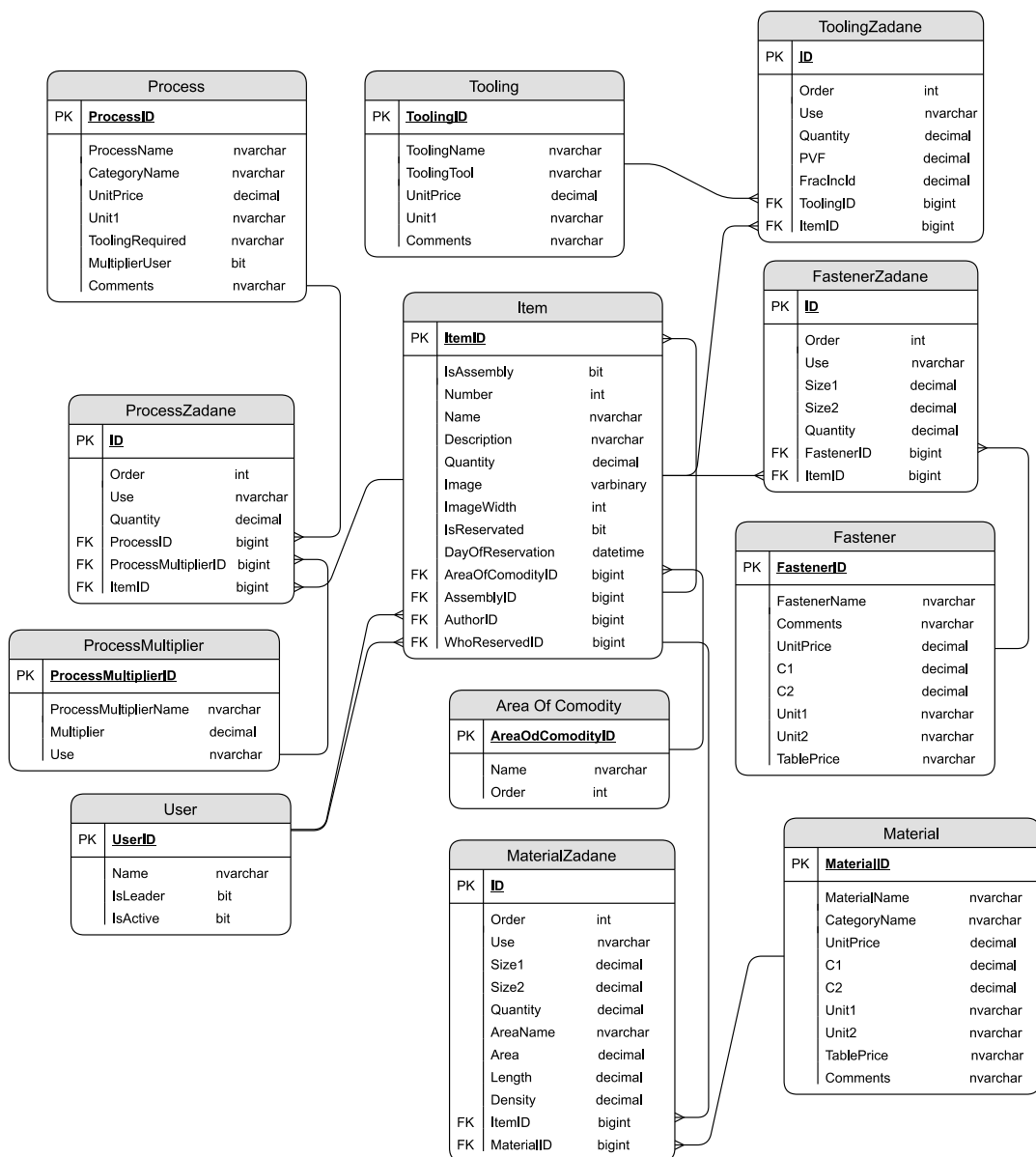
Takovéto aplikace jsou výsledkem snahy o odstínění „běžného“ uživatele od syntaxe SQL a od nutnosti znát strukturu databáze, se kterou pracují. To je ovšem zároveň i nevýhodou, jelikož uživatel má k dispozici pouze určitý soubor předpřipravených dotazů.

## 4 Návrh datové vrstvy

Dle požadavků zadavatele bude do datové vrstvy celkového řešení přidán jak datový sklad, tak i operační databáze, jejímž účelem bude uchovávat data za aktuální rok. Tato databáze bude sloužit jako zdrojový systém pro datový sklad, kdy se prostřednictvím ETL procesu získají data aktuálního roku, transformují se a následně jsou načtena do datového skladu.

### 4.1 Operační databáze

Na Obrázku č. 4.1 je zobrazen návrh logického modelu operační databáze (ERA model). Zde se tedy nachází tabulka pro již výše zmíněnou oblast komodit *Area Of Commodity* a tabulka pro sestavy a její dílčí části *Item*.



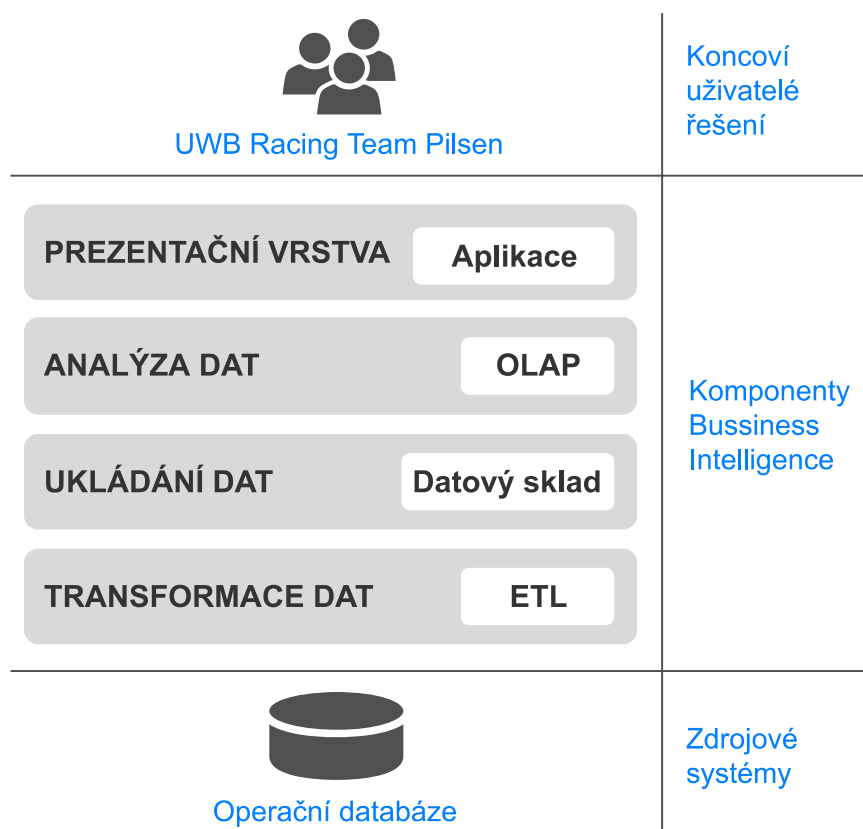
Obrázek č. 4.1: Návrh modelu operační databáze

Tabulka *Item* uchovává atributy, jako jsou název, číselné označení, popis, počet, případně i výkres sestavy či její části. Tabulka *User* uchovává data o autorech sestav a o uživateli, které mají případně aktuálně rezervované sestavy. Díky těmto informacím pak uživatelé nebudou moci jakkoli editovat tu samou sestavu zároveň.

Další tabulky se pak týkají pouze oceňovaného nákladu. Pro každou kategorii nákladu (materiál, spojový materiál, procesy a obrábění) existuje vždy tabulka, která uchovává typy nákladů, tedy data získaná z importovaných souborů (viz kapitola 2.1.1). Dále pak pro danou kategorii existuje tabulka, která uchovává náklady, které jsou uživatelem k daným položkám již přidány.

## 4.2 Datový sklad

Hlavní částí datové vrstvy celkového řešení bude centralizovaný datový sklad, jehož hlavním cílem bude sledování nákladů všech vytvořených vozidel v průběhu let. Na Obrázku č. 4.2 je zobrazen návrh struktury Business Intelligence pro tuto práci, kde se nalézá i navrhovaný datový sklad.



Obrázek č. 4.2: Komponenty Business Intelligence

## 4.2.1 Výběr dimenzí

V této kapitole se budeme zabývat výběrem dimenzí datového skladu, které budou nabízet uživateli různé pohledy na data. Mezi tyto dimenze byly vybrány následující:

### 1) Dimenze časová

Datový sklad bude plněn inkrementálně, přičemž perioda plnění bude odpovídat jednomu roku. Granularita datového skladu bude tedy na úrovni jednotlivých let. Jelikož se jedná o období, za které je vozidlo vytvořeno, nemá význam do této dimenze přidávat vyšší úroveň podrobnosti.

### 2) Dimenze oblasti vozidla

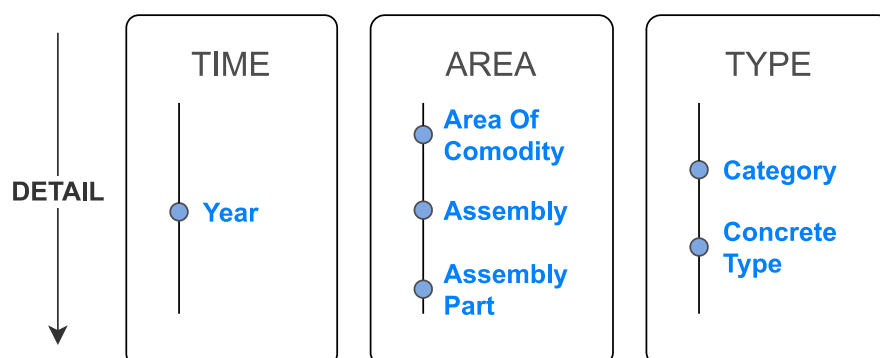
Tato dimenze bude sloužit pro třídění dat dle oblastí vozidla, kam daná položka nákladu spadá. Jak bylo již výše v této práci zmíněno, vozidlo je členěno do osmi hlavních oblastí komodit (viz kapitola 2.1.2).

V rámci těchto jednotlivých oblastí se dále vozidlo člení do tzv. sestav. Dle požadavků od zadavatele (viz kapitola 2.2) bude u dat, tedy u jednotlivých položek nákladů, uchována informace o tom, do které konkrétní sestavy spadají. Tyto druhy sestav se však v průběhu let mění. Některé jsou nově vytvořeny, některé přejmenovány či dokonce zrušeny. Z tohoto důvodu není možné provést porovnání jednotlivých sestav napříč léty. I přesto je nezbytné informaci o sestavách zachovat, aby uživatel mohl dohledat nejnákladnější sestavy. Dále zde budou uchovány informace o dílčích částech sestav. Bude se jednat o nejvyšší úroveň detailu v této dimenzi.

### 3) Dimenze typu nákladu

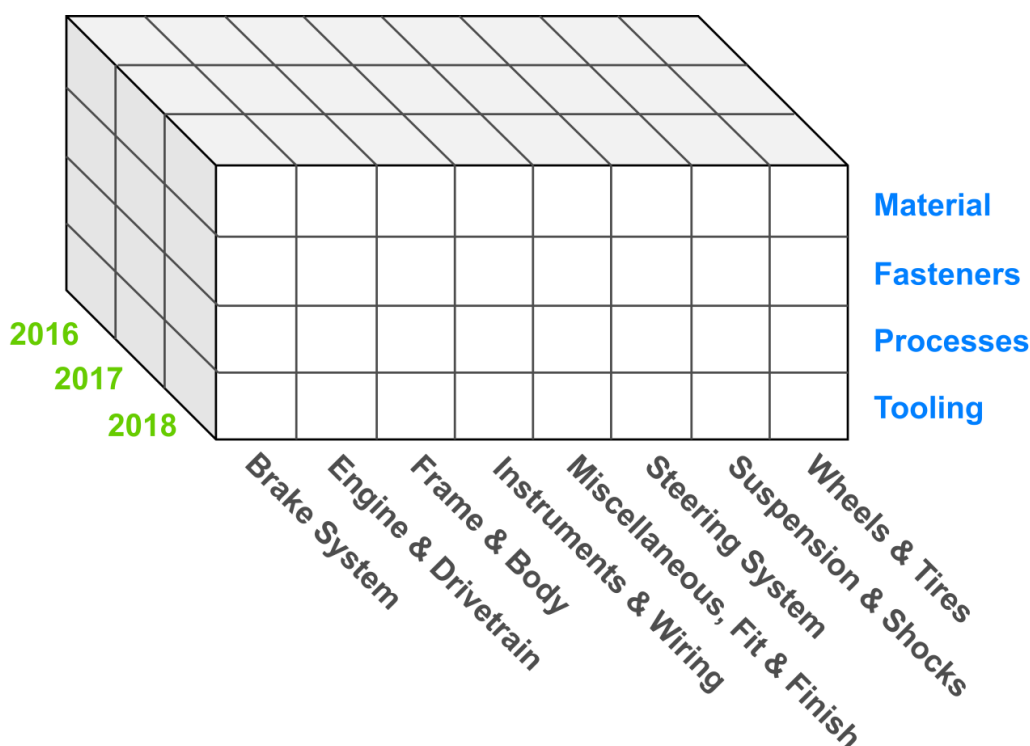
Tato dimenze se soustředí na členění nákladů vozidla dle již opět výše zmíněných kategorií (materiál, procesy, spojový materiál, obrábění), ke kterým daná položka nákladů náleží. Tyto kategorie jsou dále děleny dle typů (viz kapitola 2.1.1).

Na následujícím Obrázku č. 4.3 jsou shrnuty jednotlivé dimenze, kdy v rámci každé je znázorněno, do jaké úrovně podrobnosti lze data v datovém skladu pozorovat.



Obrázek č. 4.3: Návrh dimenzí datového skladu

Na Obrázku č. 4.4 je pro názornost zobrazena tzv. datová kostka (viz kapitola 3.3.1), kde je zřetelně vidět, jakým způsobem bude na data v datovém skladu nahlíženo z výše stanovených dimenzí. Zde je zeleně vyznačena dimenze časová, konkrétně roky 2016 až 2018. Na horizontální ose datové kostky se nachází dimenze oblasti vozidla, kde jsou šedě vypsány hlavní oblasti komodit vozidla (nejnižší úroveň podrobnosti). Na vertikální ose se nachází dimenze kategorií nákladů (nejnižší úroveň podrobnosti), které jsou zde modře vypsány.



Obrázek č. 4.4: Datová kostka – vizualizace multidimenzionálního pohledu na data

## 4.2.2 Výběr faktů

Na základě požadavků od zadavatele byl vytvořen následující seznam faktů, které lze dle stanovených dimenzí na datech sledovat:

- Cena nákladů v \$
- Kvantita nákladů
- Poměr k nákladům za oblast komodit v %

Ukazatel poměru nákladů vůči nákladům oblasti komodit vozidla bude sloužit pro snadnější dohledání nejnákladnější části vozidla. Hodnota tohoto ukazatele bude předpočítána pomocí procesu ETL (viz 3.1.1). Agregace ceny a kvantity nákladů mají smysl pro všechny vybrané dimenze datového skladu, jedná se tudíž o aditivní ukazatele (viz kapitola 3.3.1). Poslední ukazatel je ukazatelem semiaditivním, jelikož má jeho agregace smysl pouze v rámci dimenze oblasti vozidla.



### 4.2.3 OLAP

Další technologií, která bude použita, je OLAP. Ta se nalézá v rámci Business Intelligence v analytické vrstvě (viz Obrázek č. 4.2). Pro implementaci byla zvolena konkrétně technologie ROLAP (viz 3.3.1). Datový sklad bude tedy modelován prostřednictvím relačních tabulek faktů a tabulek dimenzí, který bude podrobněji popsán v následující kapitole.

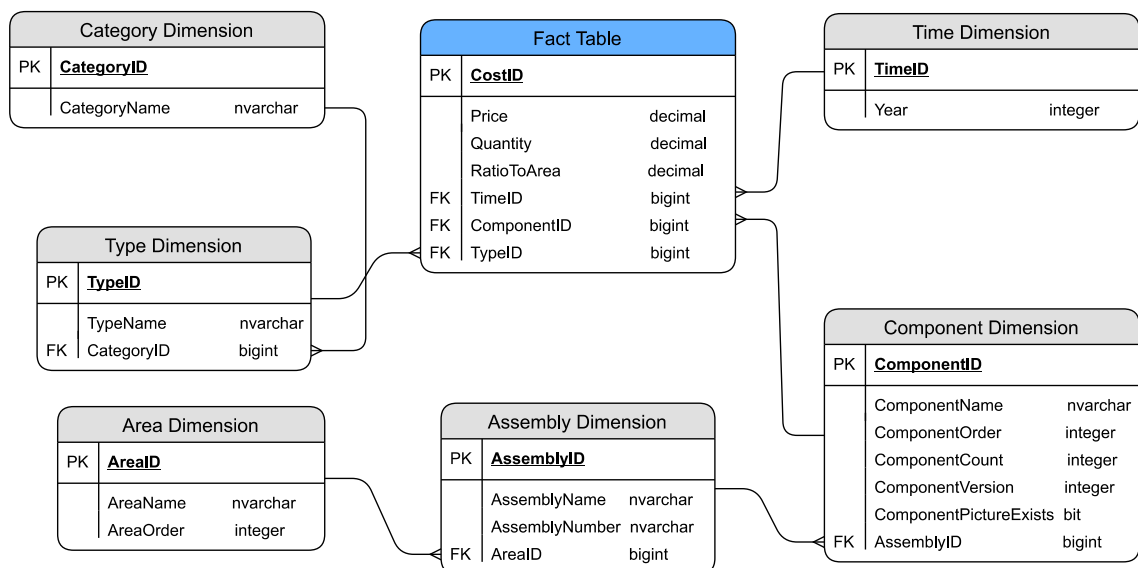
### 4.2.4 Logická struktura datového skladu

Jak již bylo řečeno, datový sklad realizujeme pomocí relačních tabulek, budeme s ním tedy pracovat pomocí relačního databázového systému (RDBMS).

Na základě zvolených dimenzí a zvolených faktů byl vytvořen model datového skladu (viz Obrázek č. 4.5). Jedná se konkrétně o schéma Sněhové vločky (viz 3.3.1), kdy kvůli úspoře dat došlo k částečné normalizaci dat u vybraných dimenzí. Za účelem úspory paměťového prostoru bude provedena normalizace databázové struktury rozdělením hlavních dimenzí na dílčí tabulky dimenzí, čímž se zabrání redundanci dat.

Jedná se o dimenzi typu nákladů, kde byla rozdělena hierarchie do dvou tabulek dimenzí „Category“ a „Type“. Rozdělena byla i dimenze oblasti vozidla, a to do jednotlivých tří tabulek dimenzí „Area“ (oblast komodit vozidla), „Assembly“ (konkrétní sestava) a „Part“ (část sestavy).

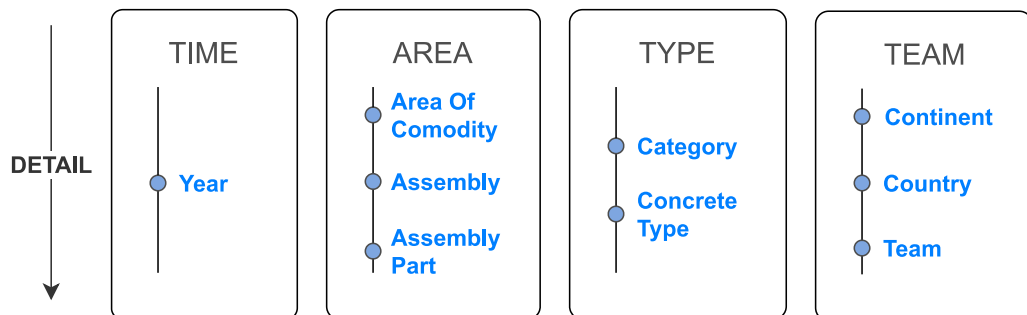
Tabulky „Part“ a „Assembly“ kromě primárních a cizích klíčů obsahují vždy název, verzi a informaci o tom, zda je u konkrétní části vozidla k dispozici obrázek. Tato informace pak slouží k průzkumu, u kolika částí vozidla chyběly dané výkresy. Díky informaci o verzi dané části, bude pak možno provádět analýzu napříč daty, kolik částí vozu bylo měněno (doděláváno). Tabulka faktů zde obsahuje tři vybraná fakta (viz 4.2.2) a cizí klíče k odpovídajícím třem hlavním dimenzím datového skladu.



Obrázek č. 4.5: Logický model datového skladu

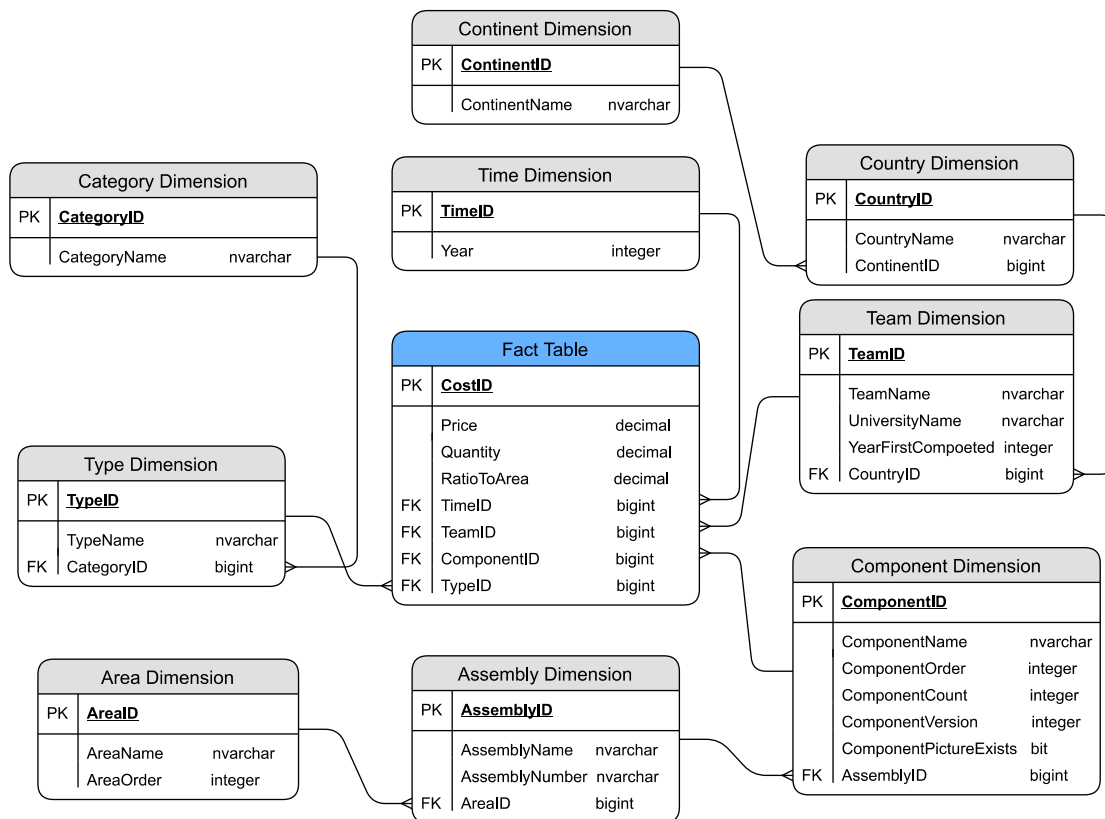
## 4.2.5 Rozšíření datového skladu

Dále tato práce bude figurovat také jako tzv. *Proof of concept*<sup>1</sup>, který bude spočívat především v rozšíření datového skladu. Účelem tohoto rozšíření je porovnání nákladů nikoli v rámci jednoho týmu, nýbrž napříč všemi týmy, které se soutěže účastní.



Obrázek č. 4.6: Dimenze rozšířeného datového skladu

Datový sklad, který byl popsán v předcházející kapitole č. 4.2.4 bude tedy rozšířen o další dimenzi, a to o dimenzi týmů, které vozidla vytváří. Jelikož soutěží více týmů i ze stejných zemí, nabízí se zde možnost porovnání mezi jednotlivými státy, dokonce i mezi kontinenty. Tím pak vzniká hierarchie v dimenzi týmu (viz Obrázek č. 4.6).



Obrázek č. 4.7: Model rozšířeného datového skladu

<sup>1</sup> Proof of concept je označení pro projekt, který vznikl za účelem ověření proveditelnosti řešení zadané problematiky.

Za účelem úspory paměťového prostoru, může být hierarchie přidané dimenze realizovaná opět pomocí tří dimenzionálních tabulek. Bude se jednat o tabulku „*Continent*“, která bude představovat nejnížší úroveň podrobnosti. Dále pak zde bude tabulka „*Country*“ a tabulka „*Team*“ (nejvyšší úroveň detailu). Tabulka „*Team*“ bude uchovávat informace o názvu týmu, názvu univerzity, roku, kdy tým poprvé soutěžil, a o počtu členů týmu. Počet členů bude moci sloužit k analýze, zda zde existuje závislost mezi výsledky týmu a počtem jeho členů. Model takového datového skladu je zobrazen na Obrázku č. 4.7.

## **Analýza dat rozšířeného datového skladu**

Takovéto rozšíření dává možnost analyzovat data za účelem odhalení různých anomálií v datech. Jelikož je cílem každého týmu vykazovat co nejnížší náklady, budou analýzy zaměřeny především na výkyvy ve směru nízkých nákladů.

Nejprve je nutné určit si v datech mez, která bude co nejlépe charakterizovat částky nákladů všech týmů. Od této meze bude možné sledovat nejruznější odchylky (anomálie). Mez lze určit použitím některého z nejnámějších a nejpoužívanějších statistických charakteristik. Tím všeobecně nejnámějším je aritmetický průměr, který zde získáme sumou částek nákladů za všechny týmy dělenou počtem týmů.

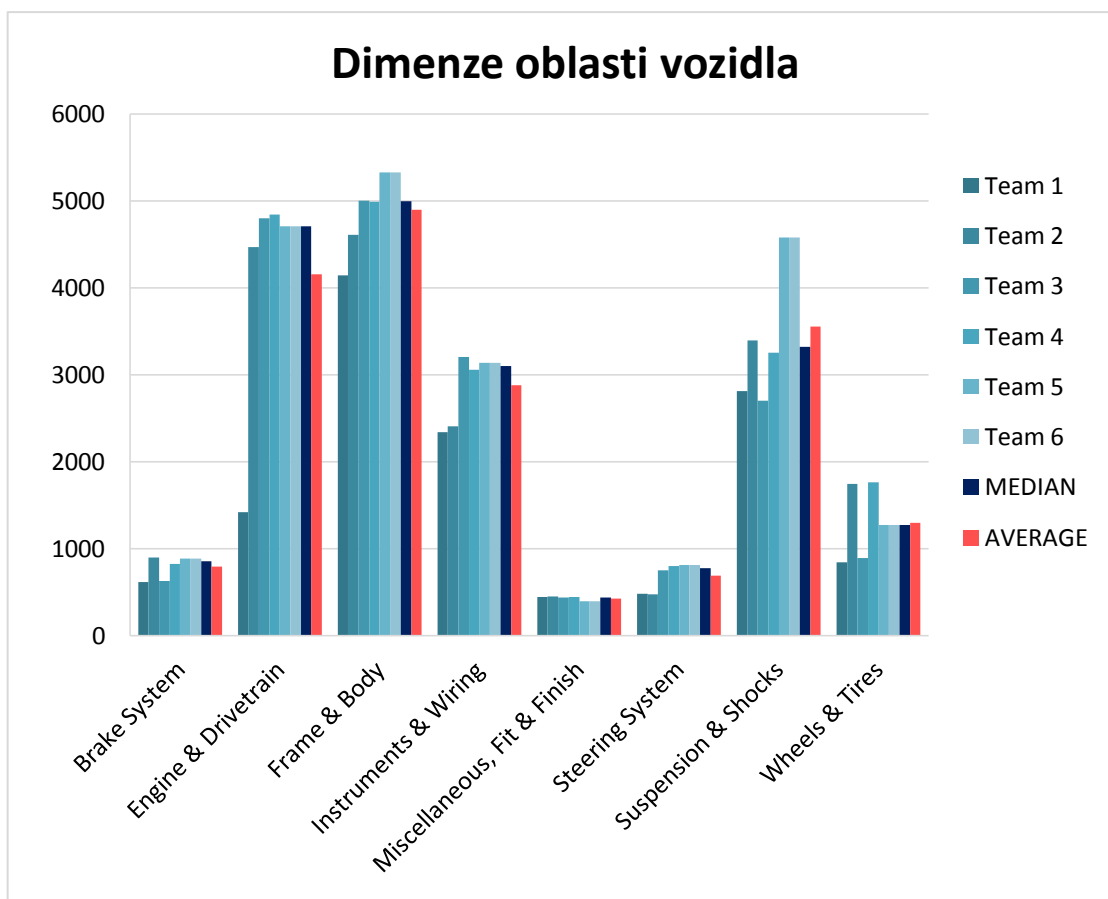
Pokud se však budou v datech vyskytovat výrazné odchylky, mohou takový průměr ovlivnit natolik, že by celková charakteristika dat byla zkreslena, což by následně mohlo zapříčinit nenalezení takovýchto anomálií [8]. Jelikož se ale v tomto konkrétním případě snažíme anomálie nalézt, je proto tato charakteristika nevhodná. Další statistickou charakteristikou je modus, jehož výsledkem je částka, která se v datech vyskytuje nejčastěji. Avšak pravděpodobnost, že se v datech budou vyskytovat stejné částky, je zde téměř nulová.

Tudíž budeme uvažovat jiný statistický nástroj a tím je medián. Ten je totiž vůči odlehlým hodnotám zcela rezistentní, a proto vhodný pro určení meze v našich datech. Pro výpočet mediánu je nejprve nutné seřadit částky do posloupnosti dle velikosti od nejmenší po největší částku. Následovně se určí medián vybráním částky neboli prvku, který se nachází uprostřed posloupnosti dat. V případě že počet dat v posloupnosti je sudý (tedy v posloupnosti neexistuje prostřední prvek), jsou vybrány dva prostřední prvky (částky), u kterých je vypočten aritmetický průměr, a tím je získán výsledný medián.

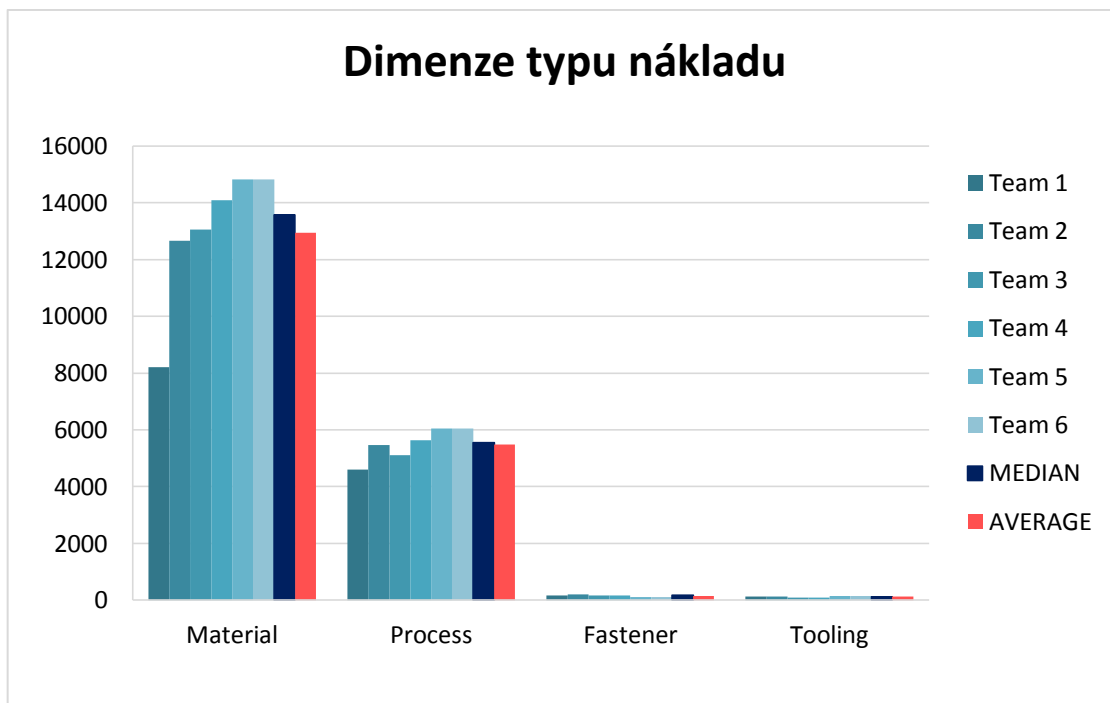
Poté přichází na řadu nalezení takových částek, které představují oproti vypočtené mezi výrazný výkyv. Určení hodnoty, pod kterou jsou náklady již kriticky nízké oproti vypočtenému mediánu, bude ponecháno uživateli. Ten si bude moci sám zvolit tuto hodnotu v podobě procentuální hranice, která bude určovat o kolik nízké náklady, než je daný medián, jsou abnormálně nízké.

Hledané anomálie se mohou projevit na více dimenzích datového skladu najednou. Mezi dimenze, ve kterých budou případné výkyvy ve výši nákladů jednotlivých týmů nejvíce zřetelné, patří především dimenze oblasti vozidla a dimenze typu nákladu. Analýza bude tedy probíhat v rámci jednotlivých oblastí komodit (dimenze oblasti vozidla) a v rámci konkrétních kategorií nákladů (dimenze typu nákladů).

Pro ověření funkčnosti takovéto analýzy byla na základě testovacích dat studentského týmu vygenerována data celkem šesti studentských týmů. Byla provedena analýza v rámci dimenze oblasti vozidla (viz Graf č. 4.1) a v rámci dimenze typu nákladu (viz Graf č. 4.2). Z nákladových položek týmu *Team 1* byl odebrán motor a převodovka. Tyto nákladové položky patří do oblasti *Engine & Drivetrain* a spadají do kategorie *Material*. V Grafu č. 4.1 a Grafu č. 4.2 jsou vykreslena data všech týmů spolu s vypočteným mediánem a průměrem. Z obou grafů lze říci, že medián charakterizuje danou množinu lépe nežli aritmetický průměr. Toto lze pozorovat například v Grafu č. 4.1 v oblasti *Engine & Drivetrain*, kdy průměr je ovlivněn nízkou hodnotou nákladů týmu *Team 1*.

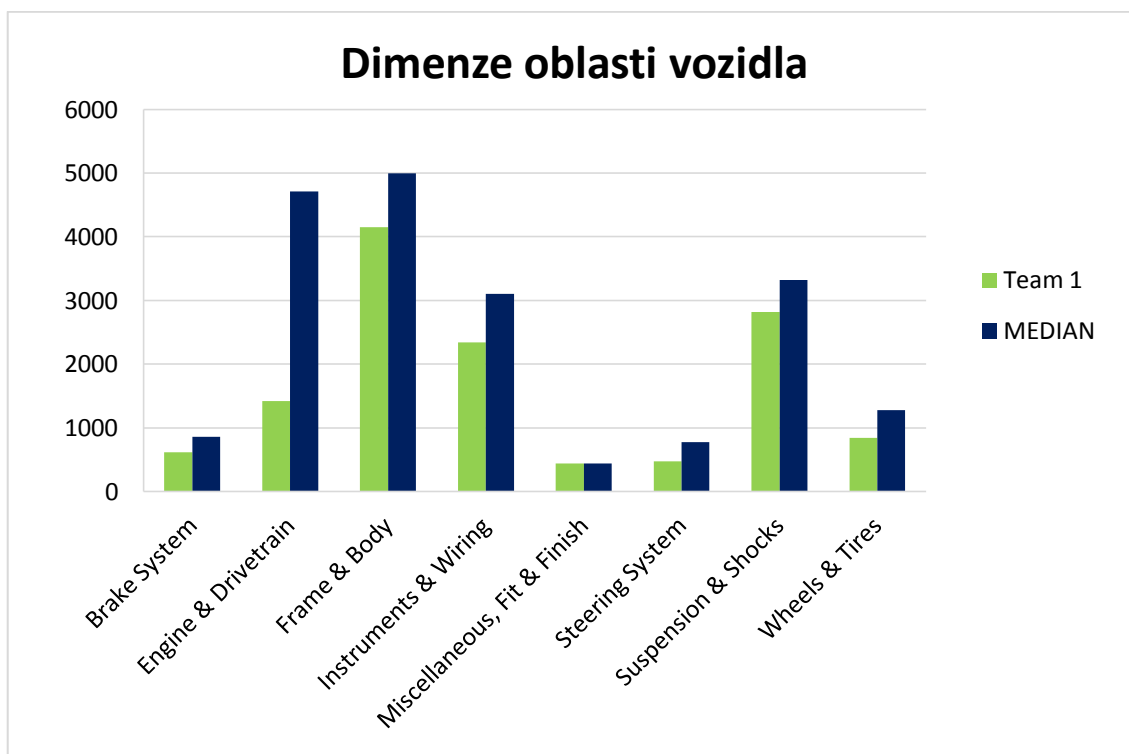


Graf č. 4.1: Dimenze oblasti vozidla, vykreslení mediánu



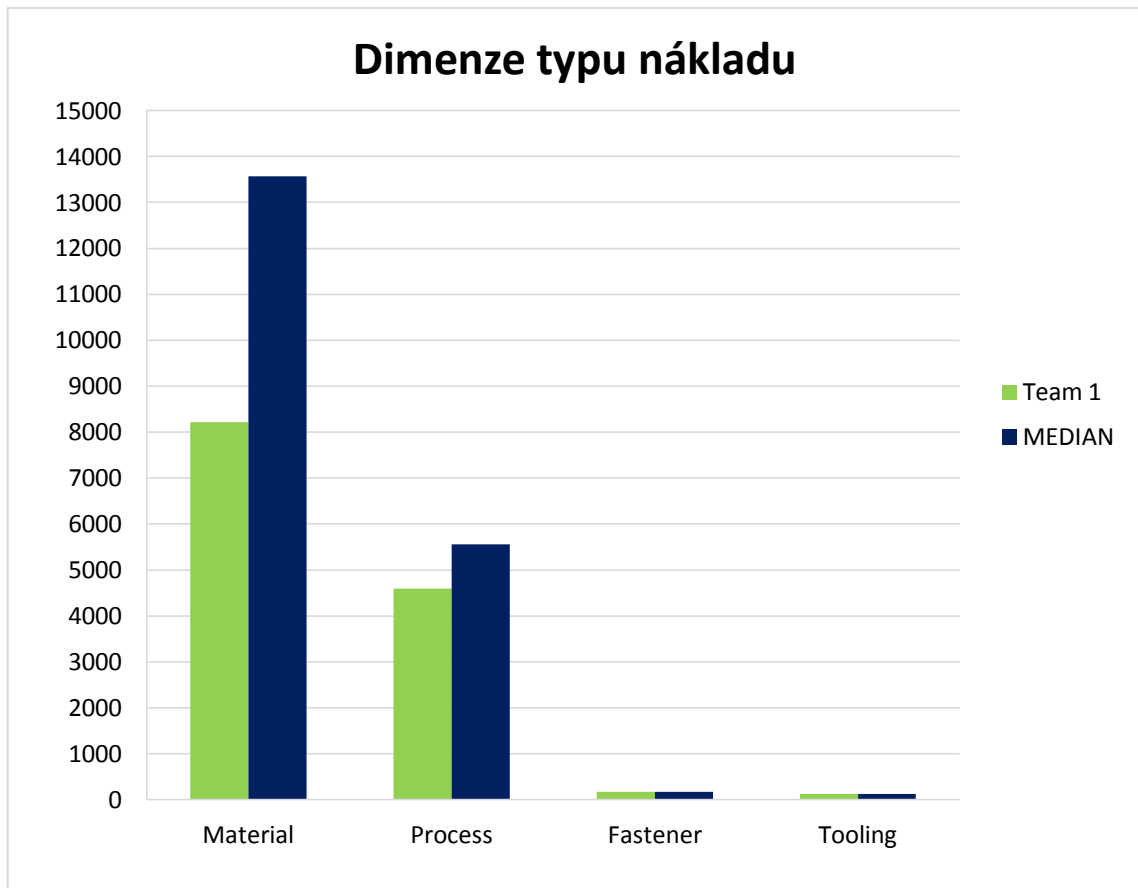
Graf č. 4.2: Dimenze typu nákladu, vykreslení mediánu

Pro větší přehlednost byly číselné řady týmu *Team 1* a vypočteného mediánu přesunuty samostatně do Grafu č. 4.3 a Grafu č. 4.4. V Grafu č. 4.3 si můžeme všimnout již zmíněného výrazného poklesu ceny nákladů v oblasti *Engine & Drivetrain*. Hodnota nákladů týmu *Team 1* v této oblasti je dokonce nižší o necelých 70 %, než činí vypočtený medián.



Graf č. 4.3: Analýza dimenze oblasti vozidla

Pokles ceny nákladů týmu *Team 1* v předchozí dimenzi se projevilo i v dimenzi typu nákladu (viz Graf č. 4.4). Zde v kategorii *Material* je výše nákladů týmu nižší o necelých 40%, než činí medián za všechny týmy.



Graf č. 4.4: Analýza dimenze typu nákladu

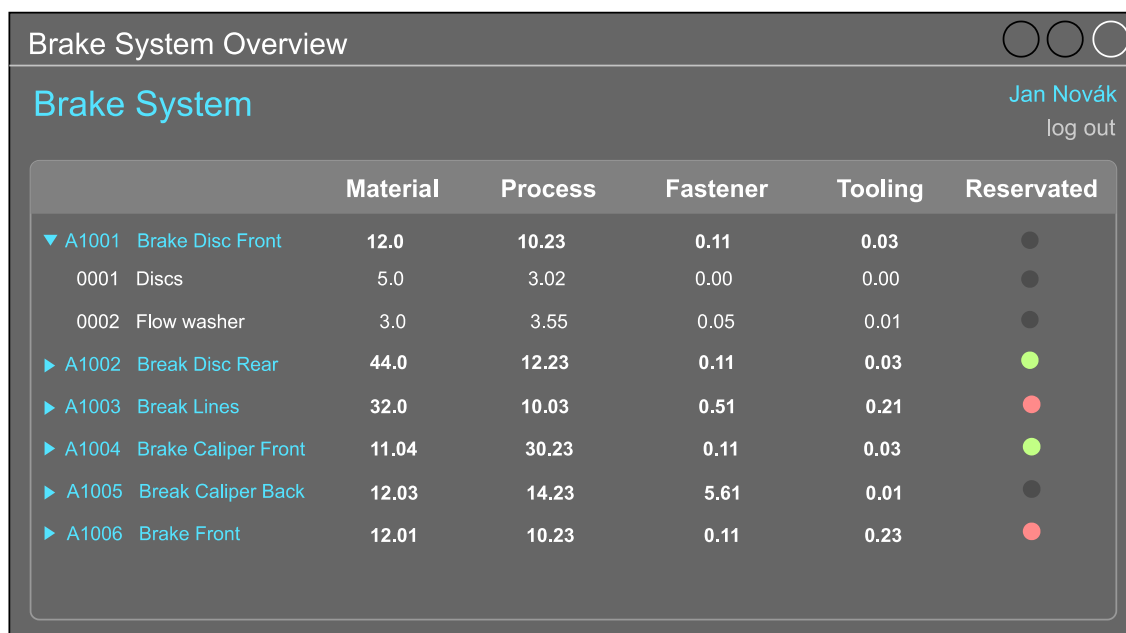
## 5 Návrh aplikační vrstvy

Tato kapitola se věnuje návrhu aplikace a dělí se na dvě stěžejní části. První část se zabývá návrhem oblasti aplikace sloužící pro práci s aktuálními daty a v části druhé je rozebrán návrh oblasti aplikace sloužící pro prohlížení historických dat.

### 5.1 Oblast aplikace pro aktuální data

Hlavním a zároveň vstupním oknem této části aplikace bude okno s přehledovou tabulkou nákladů za aktuální rok. V této tabulce bude náklad dělen do řádků dle oblastí komodit vozidla a do sloupců dle nákladové kategorie. U každé oblasti bude tlačítko, které po rozkliknutí přesměruje uživatele na seznam všech sestav dané oblasti komodit. Aby v rámci tohoto seznamu bylo možné zobrazit i dílčí části každé ze sestav, bude seznam implementován jako tzv. Tree View<sup>2</sup> (viz Obrázek č. 5.1). Zde si pak uživatel bude moci otevírat okna s podrobnostmi o jednotlivých sestavách či jejich dalších částí. Dále u každé ze sestav je prostřednictvím barevné indikace zobrazena informace o jejím aktuálním rezervačním módu. Tato barevná indikace značí celkem tři stavy:

- Šedá barva – indikuje, že daná sestava není nikým rezervována.
- Zelená barva – indikuje, že daná sestava je zarezervována aktuálním uživatelem
- Červená barva – indikuje, že daná sestava je zarezervována jiným uživatelem



The screenshot shows a window titled 'Brake System Overview' with a user 'Jan Novák' and a 'log out' button. The main content is a table with columns: Material, Process, Fastener, Tooling, and Reserved. The 'Reserved' column uses colored circles to indicate reservation status: grey for not reserved, green for reserved by the current user, and red for reserved by another user. The table lists components like Brake Disc Front, Discs, Flow washer, Break Disc Rear, Break Lines, Brake Caliper Front, Break Caliper Back, and Brake Front.

	Material	Process	Fastener	Tooling	Reserved
▼ A1001 Brake Disc Front	12.0	10.23	0.11	0.03	●
0001 Discs	5.0	3.02	0.00	0.00	●
0002 Flow washer	3.0	3.55	0.05	0.01	●
▶ A1002 Break Disc Rear	44.0	12.23	0.11	0.03	●
▶ A1003 Break Lines	32.0	10.03	0.51	0.21	●
▶ A1004 Brake Caliper Front	11.04	30.23	0.11	0.03	●
▶ A1005 Break Caliper Back	12.03	14.23	5.61	0.01	●
▶ A1006 Brake Front	12.01	10.23	0.11	0.23	●

Obrázek č. 5.1: Návrh okna se seznamem sestav konkrétní oblasti

V této části aplikace bude nejčastější aktivitou uživatelů zadávání nákladů za aktuální rok, a proto je zbytek kapitoly věnován návrhům právě těchto formulářových oken.

<sup>2</sup> Tree view je seznam se stromovou strukturou, který umožňuje hierarchický pohled na data.

## Formulář pro zadávání materiálu

Na Obrázku č. 5.2 lze vidět návrh okna formuláře pro přidání materiálu položce, kde jsou vyznačeny následující oblasti:

- 1) Oblast pro výběr typu materiálu
  - Přehled kategorií materiálu, umožňující vybrat jednotlivé kategorie
  - Pole umožňující vyhledávání typu materiálu dle názvu
  - Přehled typů materiálu, který je filtrovaný dle kategorií a dle názvu
- 2) Oblast pro nově vytvářený materiál
  - Upřesňující informace o zvoleném typu materiálu (název, kategorie, ...)
  - Oblast pro zadávání atributů materiálu nutné pro výpočet ceny materiálu
  - Cena za jednotku materiálu a cena celková

The screenshot shows a software window titled "Add Material Window". It is divided into three main sections. The left section, "Material Categories", contains a list of categories with checkboxes: "Brake System" (unchecked), "Bearings" (checked), "Control Module" (checked), and "Damper" (checked). The middle section, "List of Material types", features a search bar with the letter "B" and a magnifying glass icon, and a list of items: "Bearing, Needle", "Datalogger, Bosh", and "Damper, DNM Burner". The right section, "Calculation of Material", includes a "Name" field with "Damper, DNM Burner", a "Category" field with "Damper", and a "Comment" field with "comments". Below this are fields for "Use" (description of use), "Quantity" (2), and "unit". At the bottom, it shows "Unit price: \$ 0.68" and "Sub total price: \$ 1.36". An "Add Material" button is located at the bottom right. Red circles with numbers 1 and 2 highlight the search and calculation areas respectively.

Obrázek č. 5.2: Návrh okna formuláře pro zadání materiálu položky

## Formulář pro zadávání spojovacího materiálu

Na Obrázku č. 5.3 lze vidět návrh okna formuláře na přidání spojovacího materiálu, kde jsou barevně vyznačeny následující oblasti:

- 1) Oblast pro volbu typu spojovacího materiálu
  - Pole umožňující vyhledávání typu spojovacího materiálu dle jeho názvu
  - Přehled typů spojových materiálů, který je vyfiltrovaný dle názvu
- 2) Oblast pro nově vytvářený spojovací materiál
  - Informace o zvoleném typu spoj. materiálu (název, komentář, ...)
  - Oblast pro zadávání atributů spojovacího materiálu nutné pro výpočet ceny
  - Cena za jednotku spojovacího materiálu a cena celková



Obrázek č. 5.3: Návrh okna formuláře pro zadání spojovacího materiálu položce

## Formulář pro zadávání procesu

Na Obrázek č. 5.4 lze vidět návrh okna formuláře na přidání procesu položce. Na obrázku jsou barevně vyznačeny následující oblasti:

- 1) Oblast pro výběr typu procesu
  - Přehled kategorií typů procesů, umožňující vybrat jednotlivé kategorie
  - Pole umožňující vyhledávání typu procesu dle názvu
  - Přehled typů procesů, který je filtrovaný dle kategorií a dle názvu
- 2) Oblast pro nově vytvářený proces
  - Upřesňující informace o zvoleném typu procesu (název, kategorie, ...)
  - Oblast pro zadávání atributů procesu nutné pro výpočet jeho ceny
  - Cena za jednotku procesu a cena celková

Obrázek č. 5.4: Návrh okna formuláře pro zadání procesu položce

## Formulář pro zadávání opracování

Na Obrázku č. 5.5 lze vidět návrh okna formuláře pro přidání opracování položce. Na tomto obrázku jsou barevně vyznačeny následující oblasti:

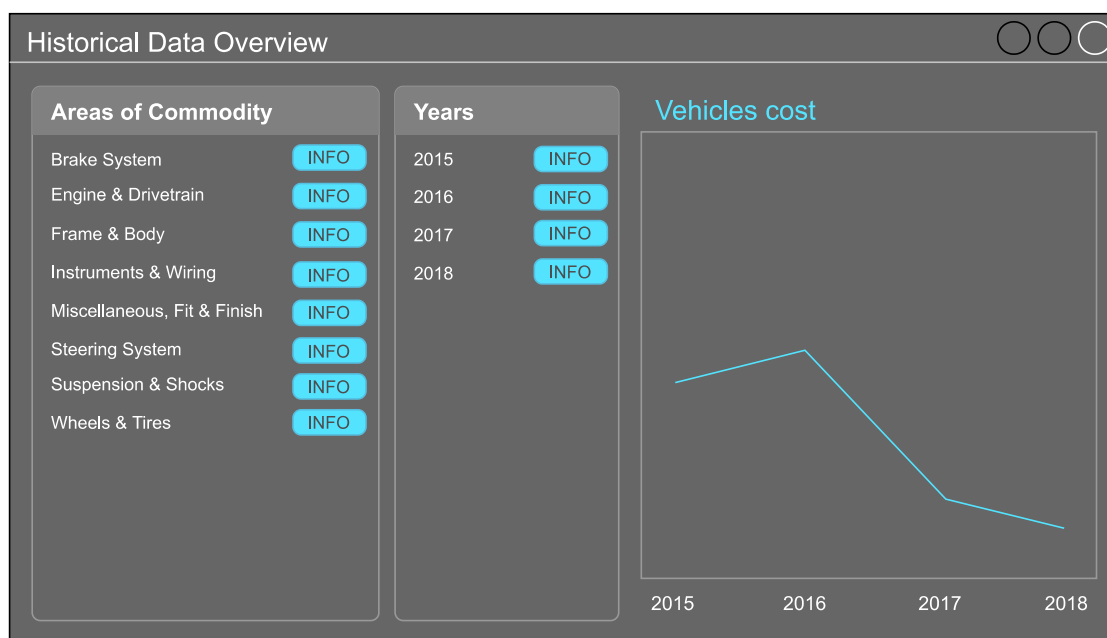
- 1) Oblast pro vybrání typu nově vytvářeného opracování:
  - Pole umožňující vyhledávání typu opracování dle názvu opracování
  - Přehled typů opracování, který je vyfiltrovaný dle názvu
- 2) Oblast pro nově vytvářené opracování:
  - Upřesňující informace o zvoleném typu opracování (název a komentář)
  - Oblast pro zadávání atributů opracování nutné především pro výpočet ceny
  - Cena za jednotku opracování a cena celková

Obrázek č. 5.5: Návrh okna formuláře pro zadání opracování položce

## 5.2 Oblast aplikace pro historická data

V této části aplikace bude vstupní a hlavní okno, jehož návrh je vykreslen na Obrázku č. 5.6. Toto okno bude obsahovat liniový graf znázorňující trend nákladů všech vytvořených vozidel. Dále zde bude seznam jednotlivých let, kde uživatel bude moci být přesměrován do oken s různými přehledy nákladů za daný rok. Jedná se především o přehledovou tabulku, kdy náklady jsou členěny dle oblastí komodit a dle jednotlivých nákladových kategorií (viz Obrázek č. 2.7). Dále pak grafické přehledy nákladů členěné pouze dle oblastí komodit (viz Obrázek č. 2.8) a dle kategorie (viz Obrázek č. 2.9).

V hlavním okně se pak bude nacházet seznam oblastí komodit, kde u každé oblasti bude existovat tlačítko pro otevření okna s náklady dané oblasti v průběhu let (viz Obrázek č. 2.4) a tlačítko pro zobrazení okna s náklady rozdělenými dle jednotlivých sestav oblasti (viz Obrázek č. 2.5). V tomto posledním okně si pak uživatel bude moci otevřít okno s podrobnostmi o jednotlivých sestavách (viz Obrázek č. 2.6).

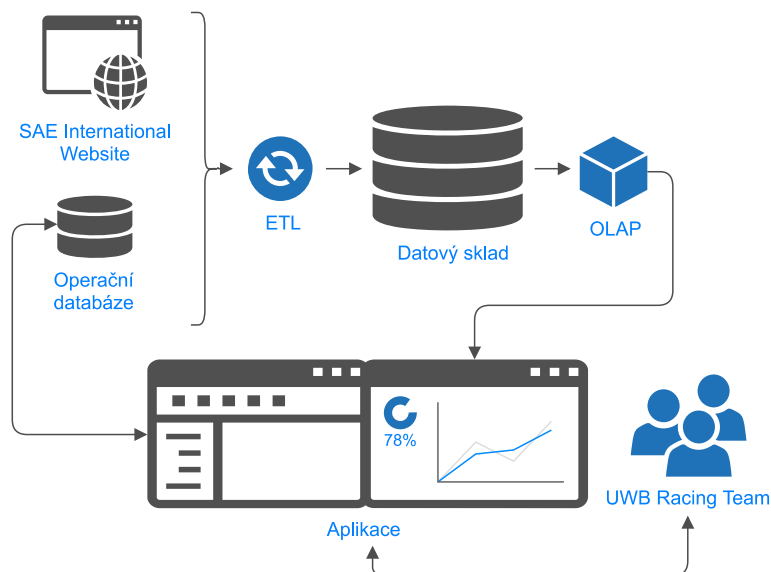


Obrázek č. 5.6: Hlavní okno části aplikace s historickými daty

## 5.3 Rozšíření aplikační vrstvy

Jak již bylo řečeno v kapitole č. 4.2.5, celkové řešení bude rozšířeno v rámci tzv. *Proof of concept*, a to spolu s aplikační vrstvou. V okamžiku, kdy organizace SAE International plně zpřístupní informace o nákladech všech týmů, které se soutěže účastní, lze tyto informace pak využít v rámci tohoto rozšíření, což je zobrazeno na Obrázku č. 5.7.

Mezi zdrojové systémy v konceptu Business Intelligence by v takovém případě přibyla webová aplikace organizace, odkud by se získávaly informace o nákladech ostatních týmů. Tyto informace spolu s aktuálními daty týmu UWB Racing Team Pilsen by se procesem ETL extrahovaly do datového skladu.



Obrázek č. 5.7: Koncept aplikace

Na základě vyhodnocení všech těchto dat je možné nejen sestavit žebříček nejlepších týmů, ale i provádět analýzu ve smyslu odhalení týmů s podezřele nízkými náklady.

Na Obrázku č. 5.8 je návrh okna, které bude poskytovat výčet nejlepších týmů, za aktuálně zvolený rok. Pro každý tým zde budou zobrazeny celkové sumy nákladů rozdělené do jednotlivých oblastí komodit vozidla (sloupce tabulky). V okně bude uživateli umožněno měnit počet vypsanych nejlepších týmů.

Na Obrázku č. 5.9 se nachází návrh okna, jehož účelem bude zobrazování tabulky všech týmů, jenž měly radikálně nízké náklady oproti mediánu nákladů všech týmu. Takovýto seznam se bude týkat vždy konkrétní oblasti komodit, kterou zde bude umožněno vybrat, a konkrétního roku, který taktéž bude možný v okně změnit. Dále uživatel bude moci zvolit procentuální hranici, jenž bude určovat minimální hodnotu, o kolik nižší náklady (než činí průměr za danou oblast) budou brán v potaz. To znamená, že při hodnotě hranice 50% se budou zobrazovat pouze takové týmy, jenž mají v rámci dané oblasti víc jak o 50 % nižší náklady než činí medián nákladů všech týmů.

Dle návrhu na Obrázku č. 5.9 bude vytvořeno další okno, které opět bude sloužit pro hledání anomálií v datech, tentokrát však v dimenzi týkající se typu nákladu. Bude zde tedy možné vybrat kategorii nákladů (materiál, proces, opracování či spojový materiál) a konkrétní rok, pro který se data budou zobrazovat. Na stejném principu pak bude vypočten medián za danou kategorii nákladů, dle kterého v závislosti na zadané procentuální hranici budou vypsány týmy s nízkými náklady.

Ranking

TOP 15 ranking for year: 2017 top teams: 15

	A1	A2	A3	A4	A5	A6	A7	A8	TOTAL
1 student team 23	798	5 342	5 123	3 765	798	1 234	3 200	1 908	22 159
2 student team 980	799	5 344	5 100	3 777	788	1 322	3 233	1 930	22 293
3 student team 234	800	5 344	5 167	3 789	821	1 230	3 280	2 102	22 533
4 student team 231	820	5 344	5 234	3 678	812	1 240	3 270	2 202	22 677
5 student team 90	824	5 350	5 167	3 726	823	1 230	3 300	2 300	22 730
6 student team 21	834	5 355	5 168	3 678	833	1 480	3 320	2 360	22 736
7 student team 332	840	5 355	5 170	3 788	830	1 460	3 323	2 370	22 341
8 student team 16	855	5 356	5 171	3 777	835	1 480	3 340	2 380	22 980
9 student team 22	861	5 366	5 172	3 788	836	1 980	3 360	2 390	23 002
10 student team 34	672	5 368	5 173	3 988	839	1 708	3 400	2 400	23 019
11 student team 45	872	5 388	5 173	3 789	844	1 530	3 500	2 432	23 108
12 student team 321	873	5 388	5 177	3 456	866	1 530	3 550	2 450	23 109
13 student team 232	873	5 389	5 178	3 567	860	1 560	3 560	2 500	23 260
14 student team 278	873	5 390	5 230	3 889	860	1 700	3 600	2 510	23 383
15 student team 187	875	5 390	5 833	3 676	870	1 900	3 950	2 812	23 496

Obrázek č. 5.8: Okno s žebříčkem nejlepších týmů

Areas of Commodity Overview

for year: 2017 Brake System Median: \$ 786.32 lower by: 50 %

**Areas of Commodity**

- Brake System
- Engine & Drivetrain
- Frame & Body
- Instruments & Wiring
- Miscellaneous, Fit & Finish
- Steering System
- Suspension & Shocks
- Wheels & Tires

**Teams with more than 50% lower cost than median**

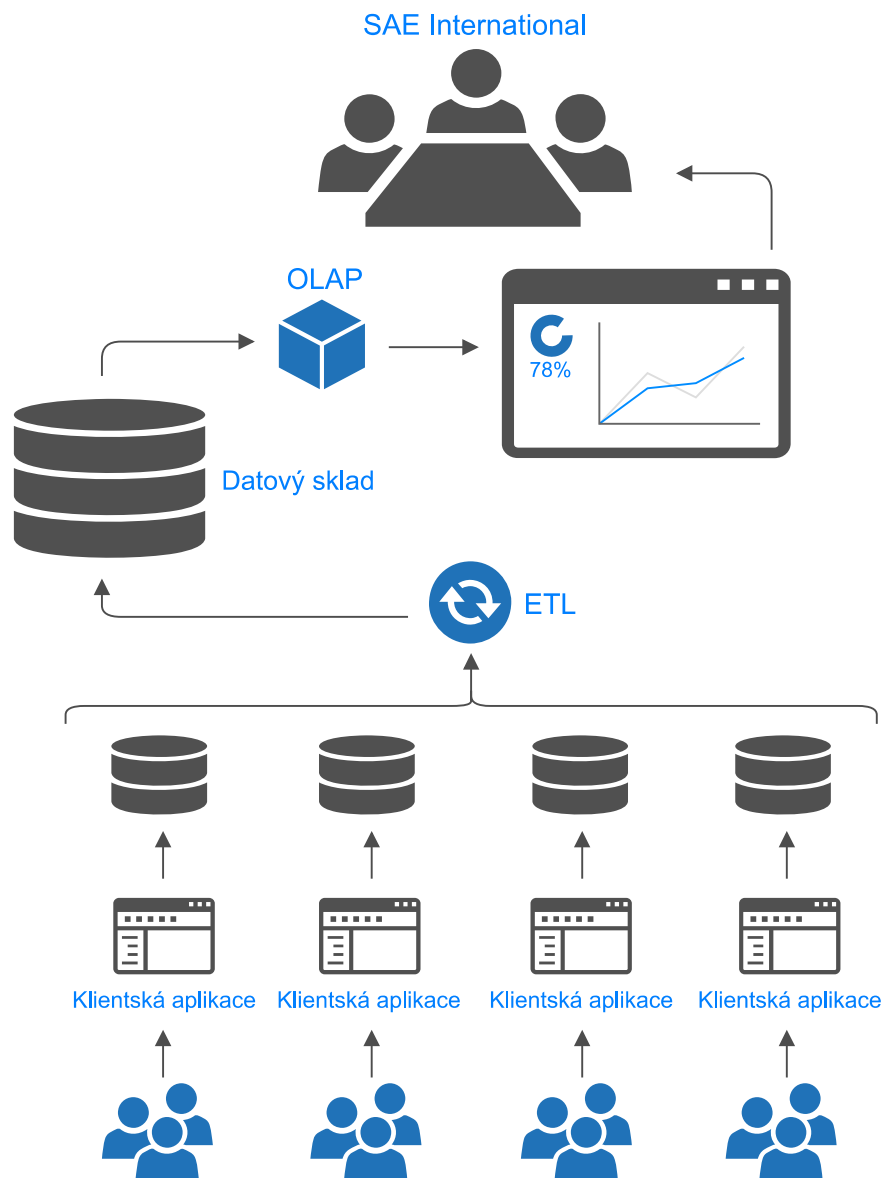
1 Student Team 23	\$ 310.23
2 Student Team 234	\$ 315.20
3 Student Team 83	\$ 340.94

Obrázek č. 5.9: Okno pro přehled týmů s výrazně nízkými náklady

Dále tato práce může sloužit jako podklad pro vytvoření poměrně rozsáhlého řešení jak pro všechny studentské týmy, které se soutěže účastní, tak i pro organizaci SAE International.

Část práce, která se soustředí na práci s aktuálními daty konkrétního týmu, by tak sloužila jednotlivým studentským skupinám. Jednalo by se tedy konkrétně o operační databázi spolu se zmíněnou částí aplikace (viz Obrázek č. 5.10). Tyto studentské systémy by pak sloužily jako zdrojové systémy, ze kterých by ETL procesem byla data

extrahována do datového skladu. Pro SAE International by pak sloužila část práce, která se zabývá historickými daty a jejich analýzou. Jedná se tedy o druhou část aplikace, datový sklad, OLAP technologii a ETL proces (viz Obrázek č. 5.10).



Obrázek č. 5.10: Koncept celkového řešení pro SAE International

## 6 Výběr technologie

Tato kapitola se věnuje výběru technologií, které budou použity pro implementaci celkového řešení.

### 6.1 Systém řízení báze dat

Pro výběr základní množiny systémů řízení báze dat byl použit server DB Engines (zdroj [9]), který se zabývá hodnocením takovýchto systémů. Tento server hodnotí systémy pro správu dat dle jejich aktuální popularity, kterou měří pomocí parametrů, jako je například počet zmínek o systému na webových stránkách, frekvence technických diskuzí o systému, počet profilů na profesionálních sítích (LinkedIn, Upwork), kde je daný systém zmíněn či počet pracovních nabídek, které se daného systému týkají [9]. Následovně je vypsáno osm systémů dle pořadí, v jakém se nacházeli na žebříčku hodnocení tohoto serveru.

- 1) Oracle (Oracle Database Express)
- 2) MySQL
- 3) Microsoft SQL
- 4) PostgreSQL
- 5) DB2
- 6) Microsoft Access
- 7) SQLite
- 8) Teradata

Aby systémy řízení báze dat nebyly zde hodnoceny pouze dle popularity, je v této práci využit další server (G2 Crowd), kde jsou systémy posuzovány dle spokojenosti uživatelů, kteří s nimi přímo pracují. Hodnocení systémů od uživatelů jsou získávány jak od komunity na tomto serveru tak i ze sociálních sítí a dalších online zdrojů [10]. Zde se nacházeli systémy pro správu dat řazené následovně:

- 1) Microsoft SQL
- 2) MySQL
- 3) Oracle
- 4) DB2
- 5) Amazon Relational Database Service (RDS)
- 6) PostgreSQL
- 7) Amazon Aurora
- 8) SAP HANA

Z obou těchto serverů byla vybrána základní množina systémů řízení dat, kam byly zahrnuty ty systémy, které se objevily na prvních devíti místech v hodnocení u obou serverů zároveň. Pozice každého ze systémů základní množiny byla vypočtena zprůměrováním jejich původních pozic ve výše vypsáných dvou množinách.

Vybraná základní množina je tedy seřazena následovně:

- 1) **MySQL, Microsoft SQL, Oracle**
- 2) **DB2**
- 3) **PostgreSQL**

Jelikož zadavatel vyžaduje pouze freeware systémy, byly poté porovnány bezplatné verze pěti výše vybraných systémů. Takové porovnání se nachází na předchozím serveru konkrétně v článku „18 Best Open-Source and Free Database Software“ [11]. Zde jsou na prvních třech místech **MySQL** (freeware edice MySQL Community), **Microsoft SQL Server** (freeware edice Microsoft SQL Server Express) a **PostgreSQL**. Na následující Tabulce č. 6.1 se nachází výsledné hodnocení těchto tří systémů. Systém MySQL získal v průměru 87%, MS SQL Server 89% a PostgreSQL získal 90%.

FEATURE RATING	MySQL	MS SQL Server	PostgreSQL
Query Language	90 %	90 %	92 %
Storage	89 %	89 %	90 %
Availability	90 %	91 %	92 %
Stability	88 %	90 %	94 %
Scalability	78 %	87 %	85 %
Security	84 %	89 %	90 %
Data Manipulation	87 %	89 %	90 %
<b>DIAMETER</b>	<b>87 %</b>	<b>89 %</b>	<b>90 %</b>

Tabulka č. 6.1: Srovnání MySQL, Microsoft SQL, PostgreSQL (zdroj: [11])

Na základě konzultace se zadavatelem byl z těchto tří systémů vybrán systém Microsoft SQL Server, konkrétně edice **Microsoft SQL Server 2014 Express**. Tato edice poskytuje kompletní databázový engine a podporuje manipulaci daty pomocí jazyka T-SQL. Obsahuje řadu vestavěných nástrojů, mezi které patří například Configuration Manager, SQL Server Management Studio, Integration ServiceSQL, Server Migration Assistant a další [12].

## 6.2 Programovací jazyk

Programovací jazyk, ve kterém bude aplikace realizována, je jazyk C#. C# je objektově orientovaný jazyk, který umožňuje vývojářům vytvářet různé zabezpečené a robustní aplikace, které běží na .NET Framework. V tomto jazyce lze vytvořit klienta Windows aplikace, webové služby XML, distribuované komponenty, aplikace typu klient-server, databázové aplikace a další. [13]



## 6.3 Technologie pro vývoj uživatelského rozhraní

Pro vývoj uživatelského rozhraní naší okenní formulářové aplikace v C# byly uvažovány následující frameworky:

- **WPF** (Windows Presentation Foundation)
- **WF** (Windows Forms)
- **UWP** (Universal Windows Platform)

V následující Tabulce č. 6.2 je vidět srovnání těchto tří frameworků. Nejstarší z nich je Windows Forms, který umožňuje jednoduchou tvorbu formulářových aplikací pomocí grafického designeru. O něco novější a komplexnější je Windows Presentation Foundation, který umožňuje tvorbu formulářů jak grafickým designérem, tak i pomocí jazyka XAML<sup>3</sup>, čímž lépe odděluje prezentační vrstvu od logiky aplikace. Dále taky obsahuje podporu pro rychlejší vykreslování, animace a Data Binding<sup>4</sup>.

Poslední a nejnovější je Universal Windows Platform, jehož model uživatelského rozhraní je opět založen na technologii XAML. Výhodou tohoto nástroje je spustitelnost takové aplikace v různých zařízeních s Windows 10, tedy nejen PC, Notebook, ale i mobil, Xbox a další. Dále také UWP aplikace deklarují k jakým zdrojům a datům mají přístup. Tento přístup pak musí uživatel povolit. [14]

.NET FRAMEWORK	WF	WPF	UWP
Poprvé představeno	rok 2002	rok 2006	rok 2012
Vykreslovací rozhraní	GDI	DirectX	DirectX
Podporované typy zařízení	PC, Notebook	PC, Notebook, Tablet	PC, Notebook, Tablet, Mobil, Xbox, ...
Podporovaný operační systém	>= Windows XP	>= Windows XP	>= Windows 10
Uživatelská interakce	optimalizováno pro klávesnici a myš	optimalizováno pro klávesnici a myš	optimalizováno i pro dotyk
Definice uživatelského rozhraní	v kódu (C#, VB,...)	XAML	XAML

Tabulka č. 6.2: Srovnání WF, WPF a UWP

<sup>3</sup> XAML je XML jazyk pro definování prvků uživatelského rozhraní

<sup>4</sup> Data Binding je objekt definující vazbu mezi cílovým objektem (element v XAML jazyce) a jakýmkoli zdrojem dat

Výhodou XAML, který je využíván posledními dvěma knihovnamy, je jeho bohatost jazyka, kterou tak nabízí širší možnosti při tvorbě uživatelského rozhraní. Dále pak umožňuje vytvořit tzv. *User Controls* objekty. Jedná se v podstatě o jakési šablony, ve kterých si vývojář může nadefinovat část uživatelského rozhraní, a tuto šablonu pak libovolně-krát použít při definování oken aplikace. To zabraňuje opakování se kódu, což je jedním z hlavních principů při tvorbě aplikace známý pod zkratkou *DRY* (Don't Repeat Yourself) [15]. Dále pak hierarchie elementů v XAML kódu je mnohem zřetelnější oproti definování elementů uživatelského rozhraní v C# u frameworku WF.

Ačkoli Universal Windows Platform poskytuje výrazně nejvíce výhod, je tato technologie dostupná pouze pro zařízení s operačním systémem Windows 10, což je dle požadavků zadavatele nežádoucí (viz kapitola 2).

Proto zvoleným nástrojem pro tvorbu naší formulářové aplikace je Windows Presentation Foundation, kdy aplikaci vytvořenou pomocí této knihovny a s nainstalovanou adekvátní verzí .NET Frameworku lze spustit na Windows XP a novějších operačních systémech.

## 6.4 Technologie pro manipulaci dat operační databáze

Pro získávání a ukládání dat byl uvažovány následující ORM<sup>5</sup> open source frameworky:

- NHibernate
- Entity Framework

NHibernate je jedním z nejdéle působících knihoven v oblasti objektově-relačního mapování. Pro popis entit a vztahů používá XML a automaticky generuje SQL pro načítání a manipulaci s daty.

Entity Framework (EF) je jedním z nejrozšířenějších ORM frameworků pro .NET platformu, který byl společností Microsoft vyvinut. EF podporuje sledování změn dat, rozlišení identity, *lazy*<sup>6</sup> a *eager*<sup>7</sup> loading a překlad dotazů nad daty [16]. Použití EF umožňuje vývojáři získávat data pomocí tzv. LINQ dotazů. LINQ neboli Language Integrated Query je .NET sada technologií se schopností integrace dotazů přímo do jazyka C#. Pomocí LINQ dotazu lze provádět operace filtrování, řazení a seskupení nad daným zdrojem dat s použitím minimálního množství kódu.

Nelze jednoznačně rozhodnout o tom, která z těchto knihoven je nejlepší volbou pro .NET projekt. Jelikož EF je jedna z nejaktivněji vyvíjejících se optimalizovaných knihoven ORM pro .NET platformu, byla posléze zvolena právě tato knihovna.

---

<sup>5</sup> ORM je zkratka pro objektově relační mapování, které zajišťuje automatickou konverzi dat mezi objektově orientovaným jazykem a relační databází.

<sup>6</sup> Lazy loading označuje strategii načítání dat z databáze, kdy data jsou načtena až ve chvíli, kdy jsou přímo dotazována.

<sup>7</sup> Eager loading označuje strategii načítání specifické sady souvisejících dat spolu s daty, které byly explicitně v dotazu požadovány.

## 7 Implementace

Tato kapitola pojednává o implementaci celkového řešení a je dělena do dvou stěžejních částí. Jedná se tedy o popis implementace databázové vrstvy a vrstvy aplikační.

### 7.1 Databázová vrstva

V rámci datové vrstvy byl implementován datový sklad dle návrhu viz Obrázek č. 4.7 a operační databáze dle návrhu viz Obrázek č. 4.1. Pro dotazování a vkládání dat do datového skladu byly vytvořeny tzv. uživatelem definované procedury a funkce, které jsou spolu s daty uloženy přímo v databázovém systému. Při použití uložených funkcí a procedur, místo poslání rozsáhlého dotazu, stačí tak poslat pouze vstupní parametry dané proceduře či funkci.

Tělo uložených procedur a funkcí je definováno pomocí jazyka T-SQL<sup>8</sup>, jehož plán je ukládán do mezipaměti a znovu používán při opakovaném spouštění. To znamená, že uživatelsky definovaná funkce či procedura nemusí být znovu zpracovávána a optimalizována s každým použitím, což má za následek mnohem rychlejší časy provedení. [10]

Ve vytvořených uložených procedurách a funkcích pro čtení dat jsou použity příkazy CUBE či ROLLUP, za cílem vytváření mezisoučtů buďto vybraných dimenzí (operátor ROLLUP) nebo všech dimenzí (operátor CUBE).

#### 7.1.1 Uživatelem definované procedury

Všechny vytvořené procedury slouží výhradně pro čtení dat a v jejich těle se nachází tzv. dynamické SQL, které je zde použito, jelikož výsledné tabulky obsahují proměnlivý počet sloupců. Dynamické SQL není příliš často využíváno, jelikož databázový systém nemůže vytvořit plně optimalizovaný spouštěcí plán takového příkazu. Mezi další nevýhody dynamického SQL patří také i hrozba tzv. SQL Injection. SQL Injection je technika napadení databázové vrstvy, kdy přes neošetřený vstupní parametr (řetězec) je do SQL příkazu vložen kód, což má většinou za následek provedení poškozujícího SQL příkazu. Jelikož však všechny vstupní parametry u všech vytvořených procedur i funkcí jsou čistě numerické, nemusíme se zde SQL Injection obávat.

Následující procedura [dbo].[SelectAreaThroughYears] (viz Zdrojový kód č. 7.1) slouží k získání přehledové tabulky, která se týká konkrétní oblasti komodit vozidla. Zde jsou náklady rozděleny dle typu nákladu (řádky) a let (sloupce). Vstupem této procedury je ID oblasti @AreaID, index určující ukazatel @Unit, který bude agregován (cena v dolarech, kvantita či poměr v procentech vůči celkové částce oblasti), a pak roky @FromYear a @ToYear určující časové období, ke kterému se výsledná data budou

---

<sup>8</sup> T-SQL (Transact-SQL) je procedurální jazyk používaný na systémech Microsoft SQL Server. Jedná se o rozšíření neprocedurálního jazyka SQL (Structured Query Language).

vztahovat. Posledním vstupním parametrem procedury je @TeamID, což je identifikátor studentského týmu, k jehož vozidlu se výsledný přehled nákladů bude vztahovat.

V těle procedury se tedy nachází výše zmíněné dynamické SQL, které je zde použito, jelikož výsledná tabulka obsahuje proměnlivý počet sloupců (počet let). Nejprve je zde vybrán seznam let, splňující podmínky vstupních parametrů, a ten je uložen do řetězce, který bude využit v dalším SELECT příkazu. V tomto řetězci se nachází předpřipravený výběr sloupců včetně jejich agregačních funkcí, které sčítají zadaný ukazatel, pokud náleží danému roku. Následuje tvorba příkazu SELECT, který je tvořen jako řetězec, do kterého se vloží předchozí řetězec s výběrem sloupců (dynamické SQL). Výsledný příkaz v řetězci je pak spuštěn pomocí příkazu EXEC.

Jelikož v tomto případě není počet sloupců tabulky vždy stejný, pak bychom použitím statického SQL nemohli data v požadovaném formátu z databázové vrstvy získat. V úvahu by nanejvýš přicházel postup, kdy bychom statickým SQL získaly tabulku, kde by data byla rozdělena do čtyř sloupců, tedy název kategorie, název typu, rok a sloupec, kde by se zadaný ukazatel agregoval (funkce SUM) dle konkrétního typu. Na konci tohoto výběru by byl přidán příkaz ROLLUP, který do výsledné tabulky zahrne i agregace (mezisoučty) pro vybrané kombinace sloupců, které jsou uvedeny v klauzuli GROUP BY. Následovně jsou vypsány dva příklady dat, přičemž první ukázka odpovídá datům, která by byla výsledkem tohoto statického SQL příkazu. V druhém příkladu je uvedena ukázka stejných dat, která byla získána naopak dynamickým SQL.

Categorie	Type	Year	Price
Fastener	Bolt, Grade 10.9	2017	0.6
Fastener	Retaining Ring	2017	0.384
Fastener	Tie Wrap	2017	7.6
Fastener	Bolt, Grade 10.9	2016	0.3
Fastener	Retaining Ring	2016	0.394
Fastener	Tie Wrap	2016	6.8
Fastener	Washer	2016	0.4
Fastener	Bolt, Grade 10.9	2015	0.6
Fastener	Retaining Ring	2015	0.384
Fastener	Washer	2015	0.4
Fastener	NULL	2017	8.184
Fastener	NULL	2016	7.894
Fastener	NULL	2015	1.384

Příklad č. 7.1: Ukázka dat získaná statickým SQL<sup>9</sup>

Categorie	Type	2017	2016	2015
Fastener	Bolt, Grade 10.9	0.6	0.3	0.6
Fastener	Retaining Ring	0.384	0.394	0.384
Fastener	Tie Wrap	7.6	6.8	-
Fastener	Washer	-	0.4	0.4
Fastener	NULL	8.184	7.894	1.384

Příklad č. 7.2: Ukázka dat získaná dynamickým SQL

<sup>9</sup> Pro zjednodušení jsou zobrazena data pouze za 3 roky a vztahují se k jediné kategorii nákladů (Fastener).

Tabulka získaná statickým SQL by tedy obsahovala několikanásobek řádků s povětšinou duplicitními hodnotami oproti tabulce získanou dynamickým SQL. Navíc tabulka získaná statickým SQL by pak musela být převedena do požadované podoby až v samotné aplikaci, kde by data musela být procházena minimálně dvěma cykly. Proto celkově by takovéto řešení vedlo k prodloužení času získání dat.

```

CREATE PROCEDURE [dbo].[SelectAreaThroughYears]
    @Unit int,
    @AreaID bigint,
    @FromYear int,
    @ToYear int,
    @TeamID bigint
AS BEGIN

-- Get name of column from dbo.CostFacts table
DECLARE @UnitColumn varchar(50) = [dbo].[GetUnitTypeColumn](@Unit)

DECLARE @SqlCommand varchar(max)
DECLARE @ColumnList varchar(max)
DECLARE @FirstPart varchar(35) = 'SUM(CASE WHEN y.Year='
DECLARE @SecondPart varchar(50) = ' THEN f.' + @UnitColumn + ' ELSE 0 END) as '

-- Get years (columns)
SELECT @ColumnList = ISNULL(@ColumnList + ', ', '') + @FirstPart +
    QUOTENAME(y.Year, '') + @SecondPart + QUOTENAME(y.Year, '')
FROM dbo.TimeDimension as y
WHERE @ToYear <= y.Year AND FromYear >= y.Year
GROUP BY y.Year
ORDER BY y.Year DESC

-- Get cost types (rows) summarized by assemblies
SET @SqlCommand = 'SELECT k.CategoryName, t.TypeName, ' + @ColumnList + '
FROM
    dbo.AssemblyDimension as a,
    dbo.ComponentDimension as c,
    dbo.CostFacts as f,
    dbo.TypeDimension as t,
    dbo.CategoryDimension as k,
    dbo.TimeDimension as y
WHERE
    a.AreaID=' + QUOTENAME(@AreaID, '') + ' AND
    a.AssemblyID=c.AssemblyID AND
    f.ComponentID=c.ComponentID AND
    f.TypeID=t.TypeID AND
    k.CategoryID=t.CategoryID AND
    f.TimeID=y.TimeID AND
    f.TeamID=' + QUOTENAME(@TeamID, '') + '
GROUP BY k.CategoryName, t.TypeName WITH CUBE'

EXEC (@SqlCommand)
END

```

### Zdrojový kód č. 7.1: Uložená procedura *SelectAreaThroughYears*

Data získána touto procedurou budou pak vizualizována v okně aplikace, které je popsáno v kapitole č. 2.2.2 (Obrázek č. 2.4). Podobným způsobem (použití dynamického SQL) jsou řešeny i další procedury.

Následující procedura [dbo].[SelectAreaYearAssemblies] (Zdrojový kód č. 7.2) získává tabulku, která se opět týká konkrétní oblasti komodit vozidla. Jsou zde taktéž rozlišovány náklady dle typu nákladu (řádky), ale do sloupců jsou náklady tříděny

dle sestav, na které se daná oblast dělí. Opět jsou zde dynamicky vybrány jednotlivé sestavy (sloupce), dle kterých jsou náklady v řádcích agregovány. Výsledná data této procedury budou použita pro okno aplikace definované v požadavcích zadavatele (viz Obrázek č. 2.5)

```

CREATE PROCEDURE [dbo].[SelectAreaYearAssemblies]
    @Year int,
    @Unit int,
    @AreaID bigint,
    @TeamID bigint
AS
BEGIN
    DECLARE @SqlCommand varchar(max)
    DECLARE @ColumnList varchar(max)
    DECLARE @UnitColumn varchar(50) = [dbo].[GetUnitTypeColumn](@Unit)
    DECLARE @FirstPart varchar(35) = 'SUM(CASE WHEN a.AssemblyNumber='
    DECLARE @SecondPart varchar(50) = ' THEN f.' + @UnitColumn + ' ELSE 0 END) as '
    -- Get area assemblies (columns)
    SELECT @ColumnList = ISNULL(@ColumnList + ', ', '') + @FirstPart +
    QUOTENAME(a.AssemblyNumber, '') + @SecondPart + QUOTENAME(a.AssemblyNumber, '')
    FROM
        dbo.ComponentDimension as c,
        dbo.AssemblyDimension as a,
        dbo.CostFacts as f,
        dbo.TimeDimension as y
    WHERE a.AreaID=@AreaID AND
        a.AssemblyID=c.AssemblyID AND f.ComponentID=c.ComponentID AND
        f.TimeID=y.TimeID AND y.Year=@Year AND f.TeamID=@TeamID
    GROUP BY a.AssemblyNumber
    -- Get cost types (rows) summarized by assemblies
    SET @SqlCommand = 'SELECT k.CategoryName, t.TypeName, ' + @ColumnList + '
    FROM
        dbo.AssemblyDimension as a,
        dbo.ComponentDimension as c,
        dbo.CostFacts as f,
        dbo.TypeDimension as t,
        dbo.CategoryDimension as k,
        dbo.TimeDimension as y
    WHERE
        a.AreaID=' + QUOTENAME(@AreaID, '') + ' AND
        a.AssemblyID=c.AssemblyID AND
        f.ComponentID=c.ComponentID AND f.TypeID=t.TypeID AND
        k.CategoryID=t.CategoryID AND f.TimeID=y.TimeID AND
        y.Year=' + QUOTENAME(@Year, '') + ' AND
        f.TeamID=' + QUOTENAME(@TeamID, '') + '
    GROUP BY k.CategoryName, t.TypeName WITH ROLLUP'
    EXEC(@SqlCommand)
END

```

### Zdrojový kód č. 7.2: Uložená procedura *SelectAreaYearAssemblies*

Následovně je ukázán příklad dat, které mohou být výsledkem takovéto procedury (pro zjednodušení se jedná opět o jednu kategorii nákladů a tři sestavy oblasti).

Categorie	Type	A1001	A1002	A1003
Fastener	Bolt, Grade 10.9	0.6	0.3	0.6
Fastener	Tie Wrap	7.6	6.8	-
Fastener	Washer	0.4	0.4	0.4
Fastener	NULL	8.6	7.5	1.0

### Příklad č. 7.3: Výsledná data uložené procedury *SelectAreaYearAssemblies*

Další vytvořenou procedurou je [dbo].[SelectAreasCostThroughYears], která vybírá tabulku obsahující informace o nákladech všech oblastí vozidla v průběhu let. V dané tabulce je tedy náklad tříděn do oblastí komodit vozidla a v rámci sloupců je tříděn dle daných let. Vstupem této procedury je pouze číselný index @Unit, který indikuje, jaký ukazatel nákladů zde bude sumarizován. Takto získaná data budou zobrazena v okně aplikace, které je definováno v požadavcích zadavatele viz kapitola 2.2.1 (Obrázek č. 2.3)

```
CREATE PROCEDURE [dbo].[SelectAreasCostThroughYears]
    @Unit int,
    @TeamID bigint
AS BEGIN

    DECLARE @SqlCommand varchar(max)
    DECLARE @ColumnList varchar(max)
    DECLARE @UnitColumn varchar(50) = [dbo].[GetUnitTypeColumn](@Unit);
    DECLARE @FirstPart varchar(35) = 'SUM(CASE WHEN y.Year='
    DECLARE @SecondPart varchar(50) = ' THEN f.' + @UnitColumn + ' ELSE 0 END) as '

    -- Get years (columns)
    SELECT @ColumnList = ISNULL(@ColumnList + ', ', '') + @FirstPart +
        QUOTENAME(y.Year, '') + @SecondPart + QUOTENAME(y.Year, '')
    FROM dbo.TimeDimension as y GROUP BY y.Year ORDER BY y.Year

    -- Get cost summarized by areas of commodity
    SET @SqlCommand = 'SELECT r.AreaName, ' + @ColumnList + '
    FROM
        dbo.AreaDimension as r,
        dbo.AssemblyDimension as a,
        dbo.ComponentDimension as c,
        dbo.CostFacts as f,
        dbo.TimeDimension as y,
        dbo.TeamDimension as e,
    WHERE
        r.AreaID=a.AreaID AND
        a.AssemblyID=c.AssemblyID AND
        f.ComponentID=c.ComponentID AND
        f.TimeID=y.TimeID AND
        f.TeamID=' + QUOTENAME(@TeamID, '') + '
    GROUP BY r.AreaName';
    EXEC (@SqlCommand)
END
```

### Zdrojový kód č. 7.3: Uložená procedura *SelectAreasCostThroughYears*

Pro ukázkou je následovně zobrazen příklad dat, která mohou být výstupem této procedury. Pro zjednodušení se jedná o tři roky a první čtyři oblasti, ke kterým se data budou vztahovat.

Area	2017	2016	2015
Brake System	617.06	629.06	887.32
Engine & Drivetrain	4420.34	4807.63	4708.83
Frame & Body	4147.54	5000.90	5223.96
Instruments & Wiring	3337.32	2904.78	3689.50

### Příklad č. 7.4: Ukázka výsledných dat procedury *SelectAreasCostThroughYears*

Následující uložená procedura [dbo].[SelectAssemblyPartsYear] získává tabulku, jejíž data se vztahují pouze ke konkrétní sestavě za konkrétní rok. Vstupem této

procedury je ID sestavy @AssemblyID, rok @Year, ke kterému se data budou vztahovat a index @Unit, který udává typ ukazatele, který ve výsledné tabulce bude sumarizován. Stejně jako u předchozích procedur je zde dynamickým SQL vybrány jednotlivé části sestavy (sloupce) a v řádcích výsledné tabulky je náklad opět tříděn dle daného nákladového typu.

```

CREATE PROCEDURE [dbo].[SelectAssemblyPartsYear]
    @Year int,
    @Unit int,
    @AssemblyID bigint,
    @TeamID bigint
AS BEGIN

-- Get name of CostFacts column based on given unit index
DECLARE @UnitColumn varchar(50) = [dbo].[GetUnitTypeColumn](@Unit);

DECLARE @SqlCommand varchar(max)
DECLARE @ColumnList varchar(max)
DECLARE @FirstPart varchar(35) = 'SUM(CASE WHEN c.ComponentOrder=';
DECLARE @SecondPart varchar(50) = ' THEN f.' + @UnitColumn + ' ELSE 0 END) as '

-- Get assembly parts (columns)
SELECT @ColumnList = ISNULL(@ColumnList + ', ', '') + @FirstPart +
    QUOTENAME(c.ComponentOrder, '')
    + @SecondPart + QUOTENAME(c.ComponentOrder, '')
FROM
    dbo.AssemblyDimension as a,
    dbo.ComponentDimension as c,
    dbo.CostFacts as f,
    dbo.TimeDimension as y
WHERE
    c.AssemblyID=@AssemblyID AND
    f.ComponentID=c.ComponentID AND
    f.TimeID=y.TimeID AND
    y.Year=@Year AND
    f.TeamID=@TeamID
GROUP BY c.ComponentOrder;

-- Get cost types (rows) summarized by assembly parts
SET @SqlCommand = 'SELECT k.CategoryName, t.TypeName,
    SUM(f.' + @UnitColumn + ') as ' + QUOTENAME('Total', '')
    + ', ' + @ColumnList + '
FROM
    dbo.ComponentDimension as c,
    dbo.CostFacts as f,
    dbo.TypeDimension as t,
    dbo.CategoryDimension as k,
    dbo.TimeDimension as y
WHERE
    c.AssemblyID=' + QUOTENAME(@AssemblyID, '') + ' AND
    f.ComponentID=c.ComponentID AND
    f.TypeID=t.TypeID AND k.CategoryID=t.CategoryID AND
    f.TimeID=y.TimeID AND y.Year=' + QUOTENAME(@Year, '') + ' AND
    f.TeamID=' + QUOTENAME(@TeamID, '') + '
GROUP BY
    k.CategoryName,
    t.TypeName
WITH ROLLUP';

EXEC (@SqlCommand);
END

```

Zdrojový kód č. 7.4: Uložená procedura *SelectAssemblyPartsYear*



Následovně je zobrazen příklad dat, která mohou být výstupem takovéto procedury. Pro zjednodušení se jedná o jedinou kategorii nákladů a tři části sestavy, ke kterým se data budou vztahovat.

Categorie	Type	TOTAL	001	002	003
Fastener	Bolt, Grade 10.9	1.5	0.6	0.3	0.6
Fastener	Tie Wrap	14.4	7.6	6.8	-
Fastener	Washer	1.2	0.4	0.4	0.4
Fastener	NULL	17.1	8.6	7.5	1.0

#### Příklad č. 7.5: Výsledná data procedury *SelectAssemblyPartsYear*

Poslední uloženou procedurou je [dbo].[GetTeamsOverview]. Ta získává přehled všech týmů seřazený dle celkových částek nákladů od nejnižší. Vstupní parametr @Year určuje, pro který rok se má daný přehled sestavit a parametr @Top určuje počet prvních týmů, které se v přehledu zobrazí.

```
CREATE PROCEDURE [dbo].[GetTeamsOverview]
    @Year int,
    @Top int
AS
BEGIN

    DECLARE @SqlCommand varchar(max)
    DECLARE @ColumnList varchar(max)
    DECLARE @FirstPart varchar(35)
    DECLARE @SecondPart varchar(50)
    DECLARE @TopStr varchar(10)

    SET @TopStr = CAST(@Top as varchar(10));
    SET @FirstPart = 'SUM(CASE WHEN r.AreaID='
    SET @SecondPart = ' THEN f.Price ELSE 0 END) as A'

    SELECT @ColumnList = ISNULL(@ColumnList + ', ', '') + @FirstPart +
        QUOTENAME(r.AreaID, '') + @SecondPart +
        CAST(r.AreaOrder as varchar(10))
    FROM dbo.AreaDimension as r
    ORDER BY r.AreaOrder

    SET @SqlCommand = 'SELECT TOP ' + @TopStr + '
        e.TeamName as Team, ' + @ColumnList + ' ,
        SUM(f.Price) as TOTAL
    FROM
        dbo.AreaDimension as r,
        dbo.AssemblyDimension as a,
        dbo.ComponentDimension as c,
        dbo.CostFacts as f,
        dbo.TeamDimension as e,
        dbo.TimeDimension as y
    WHERE
        r.AreaID=a.AreaID AND
        a.AssemblyID=c.AssemblyID AND
        f.ComponentID=c.ComponentID AND
        f.TimeID=y.TimeID AND
        f.TeamID=e.TeamID AND
        y.Year=' + QUOTENAME(@Year, '') + '
    GROUP BY e.TeamName'

    EXEC (@SqlCommand)
END
```

Zdrojový kód č. 7.5: Uložená procedura *GetTeamsOverview*

## 7.1.2 Uživatelem definované funkce

MS SQL Server 2014 Express umožňuje vytvořit tzv. *User-Defined Functions*. Tak jako funkce v jiných programovacích jazycích, tak i uživatelem definované funkce jsou rutinami, které přijímají parametry, provádějí výpočty a jejichž výstupem může být tabulka (*Table-valued Functions*) nebo skalární hodnota (*Scalar-valued Functions*) [17].

Jako první funkcí byla vytvořena funkce, která je použita v každé z výše jmenovaných procedur [dbo].[GetUnitTypeColumn]. Jedná se o *Scalar-Valued* funkci, která vrací na základě předaného indexu @Unit název sloupce tabulky [dbo].[CostFacts].

```
CREATE FUNCTION [dbo].[GetUnitTypeColumn]
(
    @Type int
)
RETURNS nvarchar(20)
AS
BEGIN

    DECLARE @Result nvarchar(20);

    SET @Result = CASE
        WHEN @Type=0 THEN 'Price'
        WHEN @Type=1 THEN 'Quantity'
        WHEN @Type=2 THEN 'RatioToArea'
        ELSE 'RatioToArea'
    END

    RETURN @Result;
END
```

Zdrojový kód č. 7.6: Uložená funkce *GetUnitTypeColumn*

Dále byly vytvořeny *Scalar-Valued* funkce, které vždy vracejí procentuální poměr dvou skutečností vůči celé oblasti komodit, vůči konkrétní sestavě či vůči všem položkám z celého roku. Patří mezi ně následující:

```
[dbo].[GetPictureRatioInYear]
[dbo].[GetPictureRatioInArea]
[dbo].[GetPictureRatioInAssembly]
[dbo].[GetRepairedRatioInYear]
[dbo].[GetRepairedRatioInArea]
[dbo].[GetRepairedRatioInAssembly]
```

První tři funkce vrací kolik procent všech komponent (v daném roce, oblasti vozidla či sestavě) obsahují nákras dané komponenty. Další tři funkce vrací, kolik komponent bylo v průběhu roku opraveno (tedy komponenty, jejichž verze je vyšší než 1). Jelikož všechny tyto funkce jsou principově stejné, je zde následovně zobrazena pouze jediná funkce [dbo].[GetPictureRatioInArea]. Vstupními parametry této funkce ID oblasti @AreaID a rok @Year, ke kterému se výsledný poměr bude vztahovat. V této funkci je nejprve spočten počet všech komponent v dané oblasti komodit a následovně počet všech komponent v dané oblasti, u kterých je k dispozici nákras.

Z těchto dvou sum je poté vypočten výsledný procentuální poměr, který tato funkce vrací.

```
CREATE FUNCTION [dbo].[GetPictureRatioInArea]
(
    @Year int, @AreaID bigint, @TeamID bigint
)
RETURNS decimal (4,2) AS
BEGIN
    DECLARE @ComponentCount decimal (10,1)
    DECLARE @PicturesCount decimal (10,1)
    DECLARE @Result decimal (4,2)

    -- Get number of all components in given area of commodity
    SELECT @ComponentCount = COUNT(DISTINCT k.ComponentID)
    FROM
        dbo.TimeDimension as c,
        dbo.CostFacts as f,
        dbo.ComponentDimension as k,
        dbo.AssemblyDimension as s
    WHERE
        c.Year=@Year AND
        c.TimeID=f.TimeID AND
        k.ComponentID=f.ComponentID AND
        s.AssemblyID=k.AssemblyID AND
        s.AreaID=@AreaID AND
        f.TeamID=@TeamID;

    -- Get number of components to which exist picture
    SELECT @PicturesCount = COUNT(DISTINCT k.ComponentID)
    FROM
        dbo.TimeDimension as c,
        dbo.CostFacts as f,
        dbo.ComponentDimension as k,
        dbo.AssemblyDimension as s
    WHERE
        c.Year=@Year AND
        c.TimeID=f.TimeID AND
        k.ComponentID=f.ComponentID AND
        s.AssemblyID=k.AssemblyID AND
        k.ComponentPictureExists=1 AND
        s.AreaID=@AreaID AND
        f.TeamID=@TeamID;

    SET @Result = (@PicturesCount / @ComponentCount) * 100;
    RETURN @Result;
END
```

#### Zdrojový kód č. 7.7: Uložená funkce *GetPictureRatioInArea*

Následovně byla vytvořena *Table-Valued* funkce vracející tabulku, kdy každý řádek odpovídá roku, za který bylo vozidlo vytvořeno. Tabulka je dále rozdělena do dvou sloupců, kdy v prvním sloupci je daný rok a v druhém je celková částka nákladů.

```
CREATE FUNCTION [dbo].[GetVehiclesCost] (@TeamID bigint)
RETURNS TABLE AS
RETURN
(
    SELECT y.Year, SUM(f.Price) AS 'Price'
    FROM dbo.CostFacts AS f, dbo.TimeDimension AS y
    WHERE f.TimeID=y.TimeID AND f.TeamID=@TeamID
    GROUP BY y.Year
)
```

#### Zdrojový kód č. 7.8: Uložená funkce *GetVehiclesCost*

Následující vytvořenou *Table-Valued* funkcí, je funkce vracející celkové částky za jednotlivé oblasti komodit vozidla pro konkrétní rok. Daný rok je tedy vstupním parametrem této funkce. V řádcích tabulky se nachází názvy oblastí a k nim souvztažně celkové částky nákladů.

```
CREATE FUNCTION [dbo].[GetAreaSumsYearTable] (@Year int,@TeamID bigint)
RETURNS TABLE AS
RETURN
(
SELECT
    r.AreaName,
    SUM(f.Price) AS 'Price'
FROM
    dbo.AreaDimension AS r,
    dbo.AssemblyDimension AS a,
    dbo.ComponentDimension AS c,
    dbo.CostFacts AS f,
    dbo.TimeDimension AS y
WHERE
    r.AreaID=a.AreaID AND
    a.AssemblyID=c.AssemblyID AND
    f.ComponentID=c.ComponentID AND
    f.TimeID=y.TimeID AND
    y.Year=@Year AND
    f.TeamID=@TeamID
GROUP BY r.AreaName
)
```

#### Zdrojový kód č. 7.9: Uložená funkce *GetAreaSumsYearTable*

Následující *Table-Valued* funkce vrací tabulku obsahující celkové sumy za dílčí kategorie nákladů pro konkrétní rok. Funkce má tedy opět jako vstupní parametr rok. V řádcích výstupní tabulky se budou vždy nacházet názvy kategorií a k nim celkové sumy nákladů dané kategorie.

```
CREATE FUNCTION [dbo].[GetCategorySumsYearTable] (@Year int, @TeamID bigint)
RETURNS TABLE AS
RETURN
(
SELECT
    k.CategoryName,
    SUM(f.Price) AS 'Price'
FROM
    dbo.CostFacts AS f,
    dbo.TimeDimension AS y,
    dbo.TypeDimension AS t,
    dbo.CategoryDimension AS k
WHERE
    f.TimeID=y.TimeID AND
    f.TypeID=t.TypeID AND
    t.CategoryID=k.CategoryID AND
    y.Year=@Year AND
    f.TeamID=@TeamID
GROUP BY k.CategoryName
)
```

#### Zdrojový kód č. 7.10: Uložená funkce *GetCategorySumsYearTable*

Dále byla vytvořena *Table-Valued* funkce [dbo].[GetAreaTeamsSums], která zajišťuje získání seznamu týmů s celkovými částkami za danou oblast komodit (parametr @AreaID) a pro daný rok (paramater @Year). Na podobném principu byla vytvořena funkce [dbo].[GetCategoryTeamsSums] (viz Zdrojový kód č. 7.12), která taktéž zajišťuje získání seznamu týmu, tentokrát ale s celkovými částkami za danou kategorii nákladu (parametr @CategoryID).

```
CREATE FUNCTION [dbo].[GetAreaTeamsSums] (@AreaID bigint, @Year int)
RETURNS TABLE AS
RETURN
(
SELECT
    e.TeamName as Team,
    c.CountryName as Country,
    SUM(f.Price) as SumPrice
FROM
    dbo.AssemblyDimension as s,
    dbo.ComponentDimension as k,
    dbo.CostFacts as f,
    dbo.TimeDimension as y,
    dbo.TeamDimension as e,
    dbo.CountryDimension as c
WHERE
    s.AreaID=@AreaID AND
    k.AssemblyID=s.AssemblyID AND
    f.ComponentID=k.ComponentID AND
    f.TimeID=y.TimeID AND
    y.Year=@Year AND
    f.TeamID=e.TeamID AND
    e.CountryID=c.CountryID
GROUP BY e.TeamName, c.CountryName
)
```

#### Zdrojový kód č. 7.11: Uložená funkce *GetAreaTeamsSums*

```
CREATE FUNCTION [dbo].[GetCategoryTeamsSums] (@CategoryID bigint, @Year int)
RETURNS TABLE AS
RETURN
(
SELECT
    e.TeamName as Team,
    c.CountryName as Country,
    SUM(f.Price) as SumPrice
FROM
    dbo.TypeDimension as t,
    dbo.CostFacts as f,
    dbo.TimeDimension as y,
    dbo.TeamDimension as e,
    dbo.CountryDimension as c
WHERE
    t.CategoryID=@CategoryID AND
    f.TypeID=t.TypeID AND
    f.TimeID=y.TimeID AND
    y.Year=@Year AND
    f.TeamID=e.TeamID AND
    e.CountryID=c.CountryID
GROUP BY e.TeamName, c.CountryName
)
```

#### Zdrojový kód č. 7.12: uložená funkce *GetCategoryTeamsSums*

Poslední vytvořenou *Table-Valued* funkcí je funkce, jejímž výstupem je tabulka poskytující všeobecný přehled nákladů za daný rok (viz Zdrojový kód č. 7.13). Náklady jsou v tabulce děleny do řádků odpovídající oblastem komodit vozidla a do sloupců odpovídající jednotlivým kategoriím nákladů.

```

CREATE FUNCTION [dbo].[GetMainYearTable]
(
    @Year int,
    @TeamID bigint
)
RETURNS TABLE
AS
RETURN
(
SELECT
    CASE WHEN GROUPING(r.AreaName) = 1 THEN 'TOTAL' ELSE r.AreaName
    END as 'Area Of Comodity',
    SUM(CASE WHEN k.CategoryName='Material' THEN f.Price ELSE 0 END) AS 'Material',
    SUM(CASE WHEN k.CategoryName='Process' THEN f.Price ELSE 0 END) AS 'Process',
    SUM(CASE WHEN k.CategoryName='Fastener' THEN f.Price ELSE 0 END) AS 'Fastener',
    SUM(CASE WHEN k.CategoryName='Tooling' THEN f.Price ELSE 0 END) AS 'Tooling',
    SUM(f.Price) as 'TOTAL'
FROM
    dbo.AreaDimension AS r,
    dbo.AssemblyDimension AS a,
    dbo.ComponentDimension AS c,
    dbo.CostFacts AS f,
    dbo.TypeDimension AS t,
    dbo.CategoryDimension AS k,
    dbo.TimeDimension AS y
WHERE
    r.AreaID=a.AreaID AND
    a.AssemblyID=c.AssemblyID AND
    f.ComponentID=c.ComponentID AND
    f.TypeID=t.TypeID AND
    k.CategoryID=t.CategoryID AND
    f.TimeID=y.TimeID AND
    y.Year=@Year AND
    f.TeamID=@TeamID
GROUP BY r.AreaName WITH ROLLUP
)

```

### Zdrojový kód č. 7.13: Uložená funkce *GetMainYearTable*

Příkladem výstupních dat této funkce mohou být následující data:

Area Of Comodity	Material	Process	Fastener	Tooling	TOTAL
Brake System	617.06	629.06	87.32	23.96	629
Engine & Drivetrain	4420.34	4807.63	8.83	96	4800
Frame & Body	4147.54	50.90	3.96	52	5003
Instruments & Wiring	3337.32	904.78	9.50	3.96	3204
Brake System	617.06	629.06	887.32	23.96	629
Engine & Drivetrain	1420.34	807.63	80.83	96	4800
Frame & Body	1147.54	1000.90	3.96	52	5003
Instruments & Wiring	3337.32	1904.78	9.50	3.96	3204
TOTAL	13780	5800	1300	167	18349

### Příklad č. 7.6: Výstupní data funkce *GetMainYearTable*

## 7.2 Aplikační vrstva

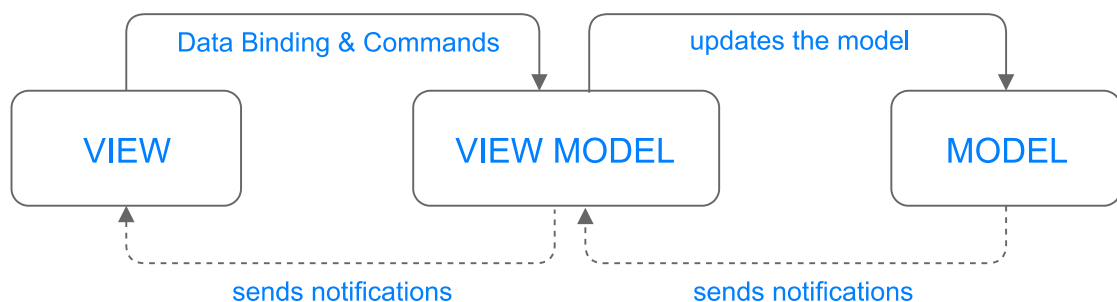
V průběhu implementace aplikační vrstvy byla využita webová služba *Bitbucket*, která slouží pro usnadnění vývoje softwaru pomocí verzovacích nástrojů Git a Mercurial. Verze zdrojového kódu byly postupně přidávány do Bitbucket repozitáře, spolu s tzv. *issues trackers*, které slouží pro sledování nových požadavků na funkcionalitu softwaru, informování o případných chybách, které mají být opraveny či další úkoly.

### 7.2.1 Architektura aplikace

V rámci této práce byla architektura aplikace implementována konkrétně dle vzoru *Model-View-ViewModel* (MVVM). Tento vzor patří mezi všeobecně známé vzorové modely MVx. Účelem těchto vzorů je členění architektury tak, aby jednotlivé části softwaru byly rozděleny do vrstev se společnou funkcionalitou. Jde především o oddělení prezentační vrstvy s vrstvou business logiky aplikace. Takovéto rozdělení architektury aplikace do vrstev je známé pod pojmem *Separation of concerns* a patří mezi hlavní principy softwarové architektury [15]. Oddělení logiky aplikace od uživatelského rozhraní pomáhá řešit řadu problémů při vývoji a může usnadnit testování aplikace, její údržbu a případně další vývoj aplikace [18].

Mezi vzorové modely softwarových architektur MVx patří *Model-View-Controller* (MVC), *Model-View-Presenter* (MVP) a již zmíněný MVVM. Jak již název napovídá, tento vzor se dělí na tři klíčové části (viz Obrázek č. 7.1):

- Model – entitní objekty, DAO<sup>10</sup> objekty, pravidla
- View – uživatelské rozhraní
- ViewModel – agent nebo prostředník mezi modelem a uživatelským rozhraním



Obrázek č. 7.1: Vzor MVVM (zdroj [18])

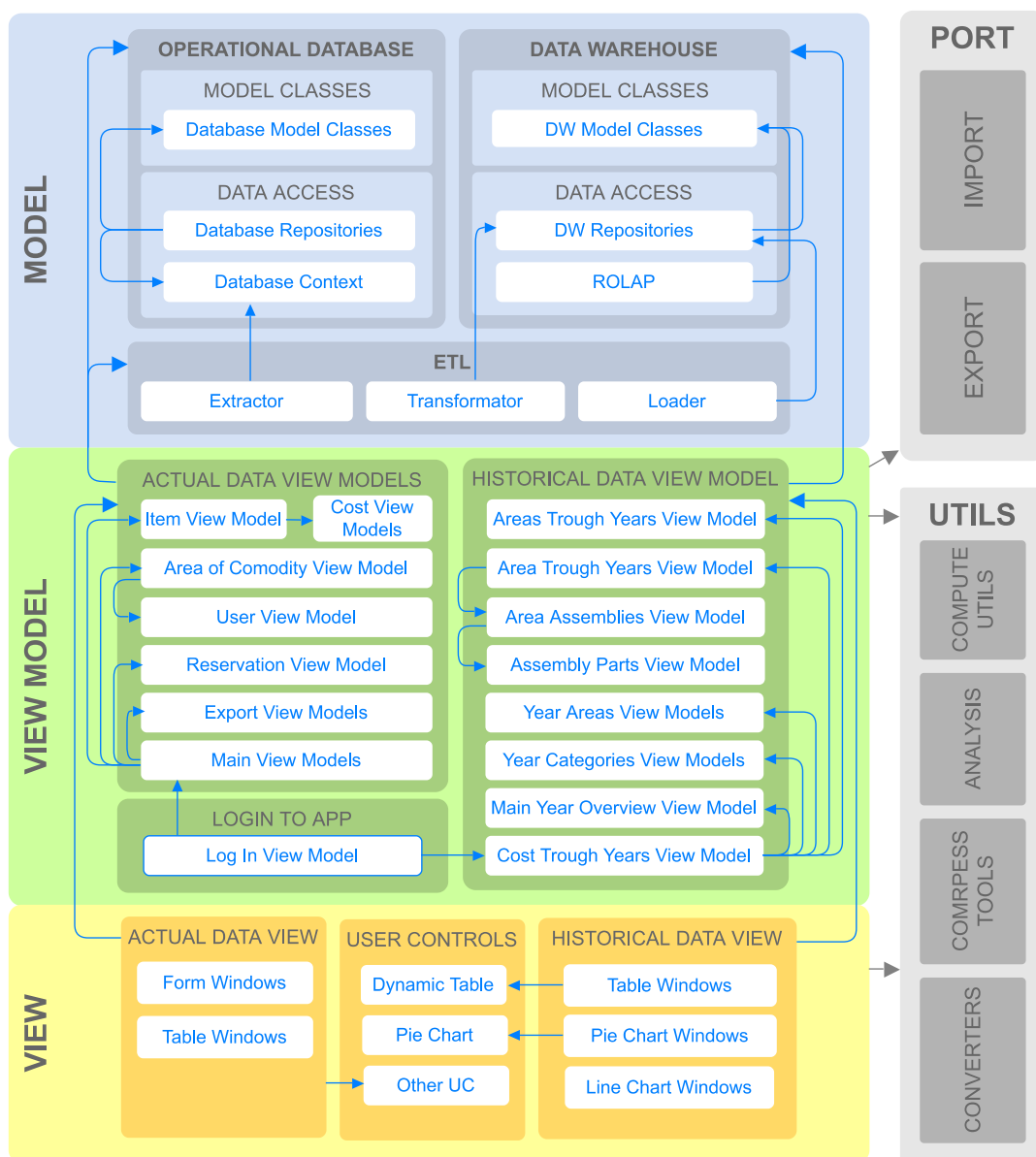
V části *Model* se tedy nacházejí třídy s přístupem ke zdrojovým datům a třídy, na která se zdrojová data mapují. Ve *View* části se nacházejí objekty, které definují uživatelské rozhraní XAML kódem (viz kapitola 6.3). Vrstva *ViewModel* pak slouží jako propojení obou částí. Objekt *ViewModel* je vždy nastaven jako tzv. *Data Context*<sup>11</sup>

<sup>10</sup> DAO (Data Access Object) abstrahuje a zapouzdřuje veškerý přístup ke zdrojům dat. Spravuje spojení se zdrojem dat, aby mohl data získávat a ukládat.

<sup>11</sup> *Data Context* – základní zdroj dat pro objekt ve *View* vrstvě

pro *View* objekt. To znamená, že vybrané atributy elementů ve *View* objektu mohou být přímo napojeny na hodnoty atributů *ViewModelu*. To je provedeno pomocí tzv. Data Binding, což je objekt definující vazbu mezi cílovým objektem (element v XAML jazyce) a jakýkoli zdrojem dat (atribut *ViewModelu*). Uživatel pak může skrze *View* objekt měnit přímo atributy *ViewModelu*, a tím upravovat entitní třídy či spouštět akce v *Model* části. Provedené změny ve vrstvě *Model* se pak skrze *ViewModel* projeví ve *View* vrstvě (viz Obrázek č. 7.1).

Na následujícím Obrázku č. 7.2 se nachází graficky znázorněná architektura naší aplikace. Jsou zde naznačeny vrstvy dle MVVM vzorového modelu. Vrstva *Model* je rozdělena na tři klíčové části. Mezi ně patří část určená operační databázi, část pro datový sklad a část věnována ETL procesu (viz kapitola 3.1.1).



Obrázek č. 7.2: Architektura aplikace<sup>12</sup>

<sup>12</sup> Pro přehlednost jsou implementované třídy zobrazeny souhrnně dle typu (co se funkcionality tříd týče).



Pro datový sklad byly vytvořeny *Repository* třídy, kdy každá ze tříd odpovídá jedné z dimenzí datového skladu a jejímž prostřednictvím se přistupuje k datům v dané dimenzi. Dále byla vytvořena třída *ROLAP*, která získává data z více dimenzí, tedy poskytuje multidimenzionální pohledy na data (viz kapitola 3.3.1).

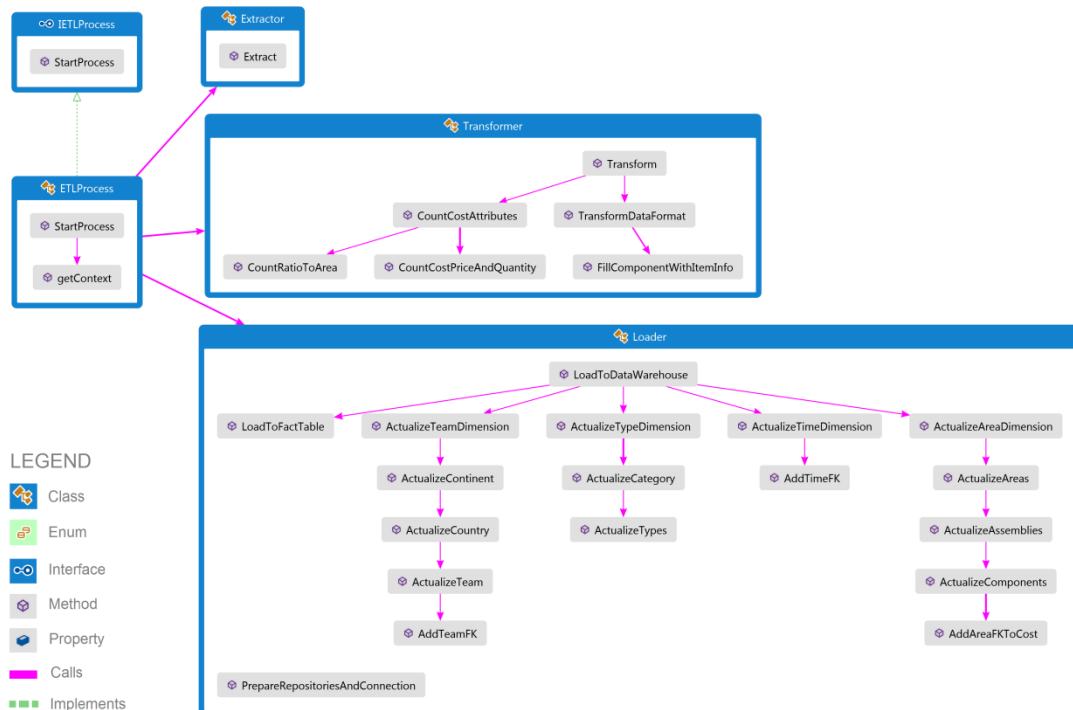
Pro operační databázi byly taktéž vytvořeny *Repository* třídy, které přistupují k datům jednotlivých tabulek databáze a třída *Database Context*, která udržuje celý kontext databáze. Poslední částí *Model* vrstvy je ETL Proces, pro jehož stěžejní podprocesy byly vytvořeny třídy *Extractor*, *Transformator* a *Loader* (viz Obrázek č. 7.2). Implementaci ETL procesu se blíže věnuje kapitola 7.2.2.

Ve vrstvě *ViewModel* jsou objekty rozděleny dle toho, zda operují s daty pro aktuální rok nebo s daty historickými. Součástí této vrstvy je i *LoginViewModel* objekt, který slouží jako rozcestník při spuštění aplikace a zajišťuje přesměrování aplikace buďto do aplikační části s aktuálními daty, nebo do části s přehledem historických dat. Tato vrstva komunikuje s vedlejší vrstvou aplikace *Utils*, jenž poskytuje třídy pro analýzu dat, počítání vzorců, či datovou konverzi. Dále komunikuje i s vedlejší vrstvou *Port*, která slouží pro import a export dat, které jsou blíže popsány v kapitolách 2.1.1 a 2.1.4.

Poslední aplikační vrstvou je *View*, kde se dané objekty opět dělí dle typu dat, která jsou vizualizována. Součástí vrstvy jsou tzv. *User Controls* objekty, které jsou blíže popsány v kapitole 6.4.

## 7.2.2 ETL

ETL proces neboli datová pumpa (viz kapitola 3.1.1) byla implementována v rámci aplikační vrstvy a celý její běh je zajišťován třídou s názvem *ETLProcess*. Tato třída pracuje s třídami odpovídajícím stěžejním krokům, ze kterých se proces ETL skládá. To lze vidět na následujícím Obrázku č. 7.3.



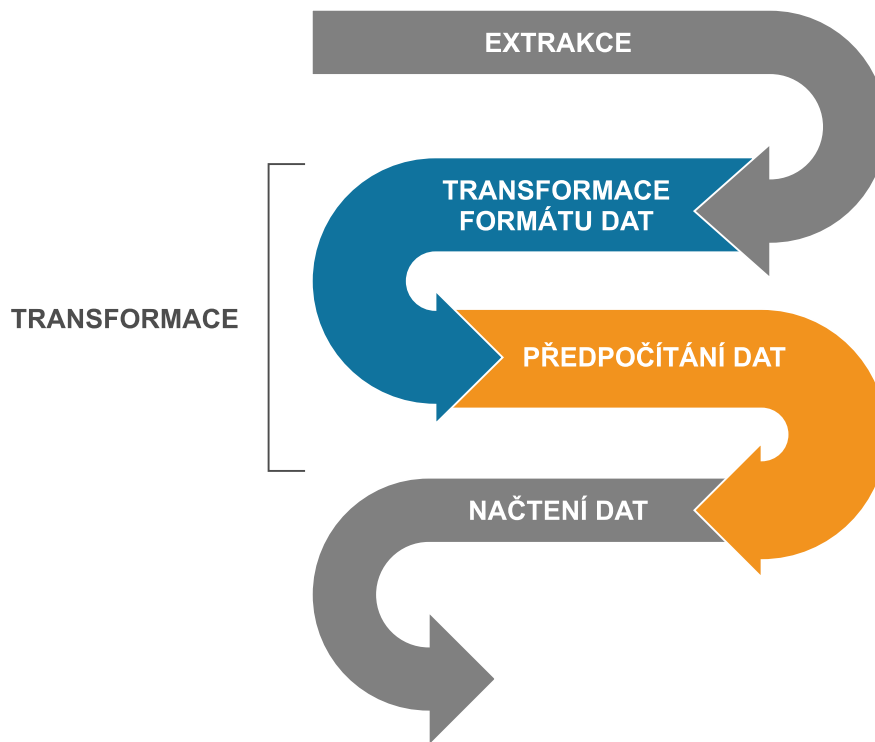
Obrázek č. 7.3: Implementace ETL procesu (Code Map diagram)

### Extrakce dat

Prvním krokem procesu je extrahování dat z operační databáze, pro které byla vytvořena třída *Extractor*. Jelikož extrakce dat probíhá pouze z jediného systému, má tato třída jedinou metodu vracející seznam všech oblastí komodit, které v sobě nesou reference na další data z operační databáze.

### Transformace dat

Tato získaná data jsou pak vstupem dalšího kroku, kterým je transformace, dělí se do dvou hlavních podprocesů (viz Obrázek č. 7.4). Celkově pro tento proces byla implementována třída *Transformer*, ve které jsou oba tyto podprocesy obsaženy. Data jsou tedy nejprve upravena do cílového formátu, ve kterém budou ukládána do datového skladu. Tato transformace formátu dat je ukázána na následujícím Obrázku č. 7.5. Zde je zobrazen na levé straně formát dat zdrojové, operační databáze a na straně druhé je pak formát cílový. Na obrázku jsou barevně naznačeny modely dat, kdy související objekty jak ve zdrojovém tak v cílovém formátu mají vždy stejnou barvu.



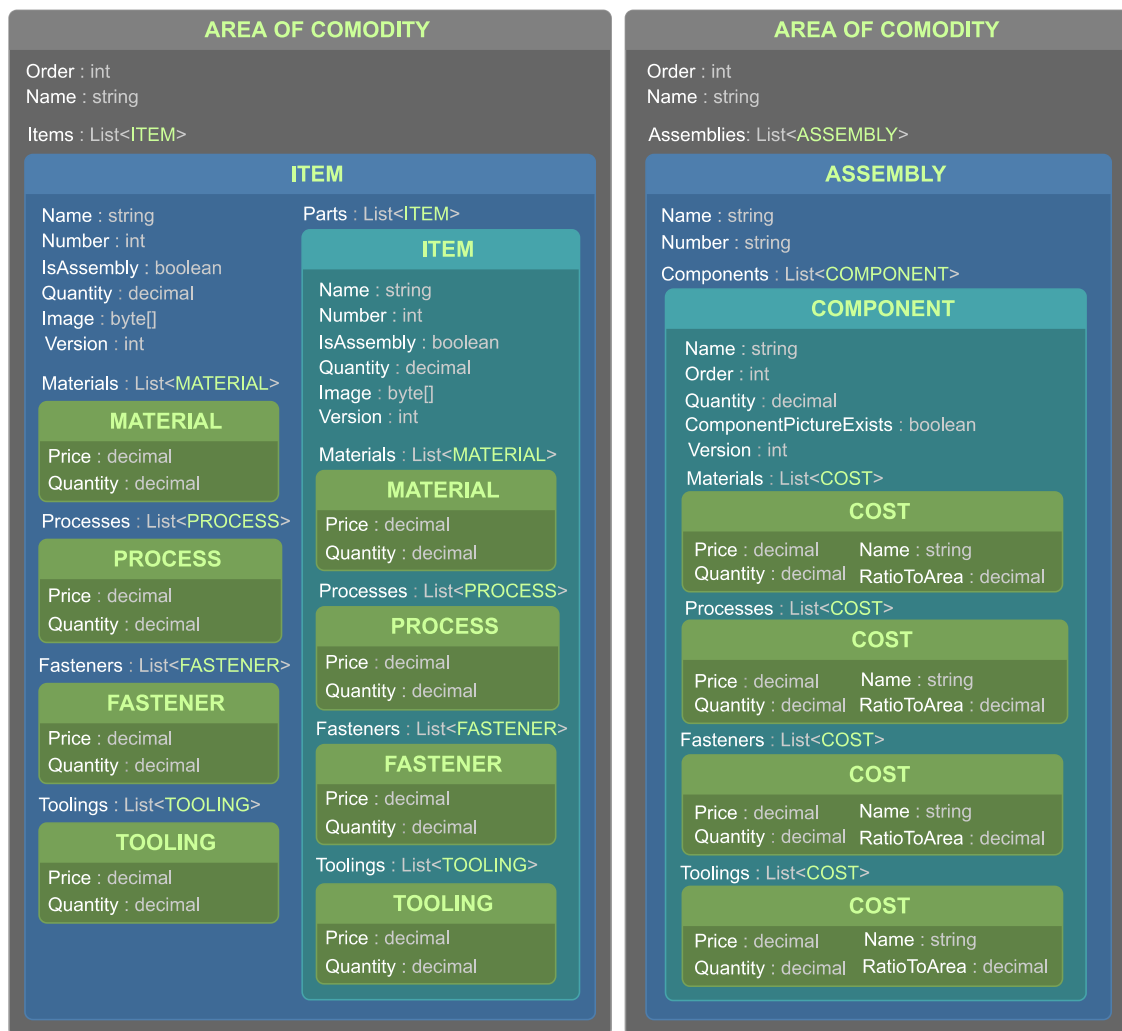
Obrázek č. 7.4: Proces transformace v rámci ETL procesu

K nejmarkantnější transformaci zde dochází u sestav, kdy náklady patřící přímo sestavě, informace o její kvantitě a další atributy jsou přesunuty vždy do nové komponenty, která je součástí seznamu komponent sestavy. Atribut pořadí nově vytvořené komponenty je pak pokaždé nastaven na hodnotu 0. Příklad této transformace lze pozorovat na ukázce vstupních dat (Příklad č. 7.7) a dat cílových (Příklad č. 7.8).

Druhou výraznou změnou ve formátu dat je sjednocení objektů nákladů `MATERIAL`, `PROCESS`, `FASTENER`, `TOOLING` do jednotného nákladového objektu `COST`, který odpovídá konkrétnímu záznamu tabulky faktů v datovém skladu. Aby byla zachována informace o kategorii daného nákladu, jsou tyto nákladové objekty roztříděny dle kategorií v rámci komponenty, ke které patří (viz Obrázek č. 7.5, objekt `COMPONENT`). Dále tento objekt obsahuje řetězec, který jednoznačně určuje, o jaký typ nákladu se jedná.

V dalším podprocesu transformace pak dochází k předpočítání faktů nákladů (viz Obrázek č. 7.5, atributy objektu `COST`). Jedná se konkrétně o celkové hodnoty ceny a kvantity nákladu a poměru ceny vůči celkové ceně oblasti komodit (ukazatele viz kapitola 4.2.2).

V celkové ceně a kvantitě nákladu je zahrnuta i kvantita komponenty, ke které daný náklad patří. To znamená, že v rámci tohoto procesu je kvantita nákladu a jeho cena vždy umocněna kvantitou její komponenty. Při zpracovávání těchto dvou ukazatelů je zjišťována celková nákladová suma za oblast komodit, která slouží pro dopočtení třetího ukazatele, kdy se vůči této sumě vypočte podíl ceny konkrétního nákladu.



Obrázek č. 7.5: Zdrojový a cílový formát dat<sup>13</sup>

<sup>13</sup> Pro přehlednost nejsou zde zobrazeny všechny objekty zdrojových dat a u všech objektů jsou zobrazeny atributy pouze takové, které mají v rámci transformace význam.

```

AREA:
  Name: „Brake System“,
  Order: 1,
  Items: {
    ITEM:
      Name: „Brake Assembly“,
      Number: 1,
      Quantity: 13,
      Version: 1,
      IsAssembly: true,
      Image: { 0x00, 0x0a, 0x0f, 0x44, 0x44, 0x43, 0x55, 0x00}
      Materials: {
        MATERIAL: { Price: 10, Quantity: 2 }
      },
      Parts: {
        ITEM:
          Name: „Brake Part“,
          Number: 1,
          Quantity: 3,
          Version: 3,
          IsAssembly: false,
          Processes: {
            PROCESS: { Price: 3, Quantity: 1 }
          }
        }
      }
  }

```

#### Příklad č. 7.7: Ukázka zdrojových dat

```

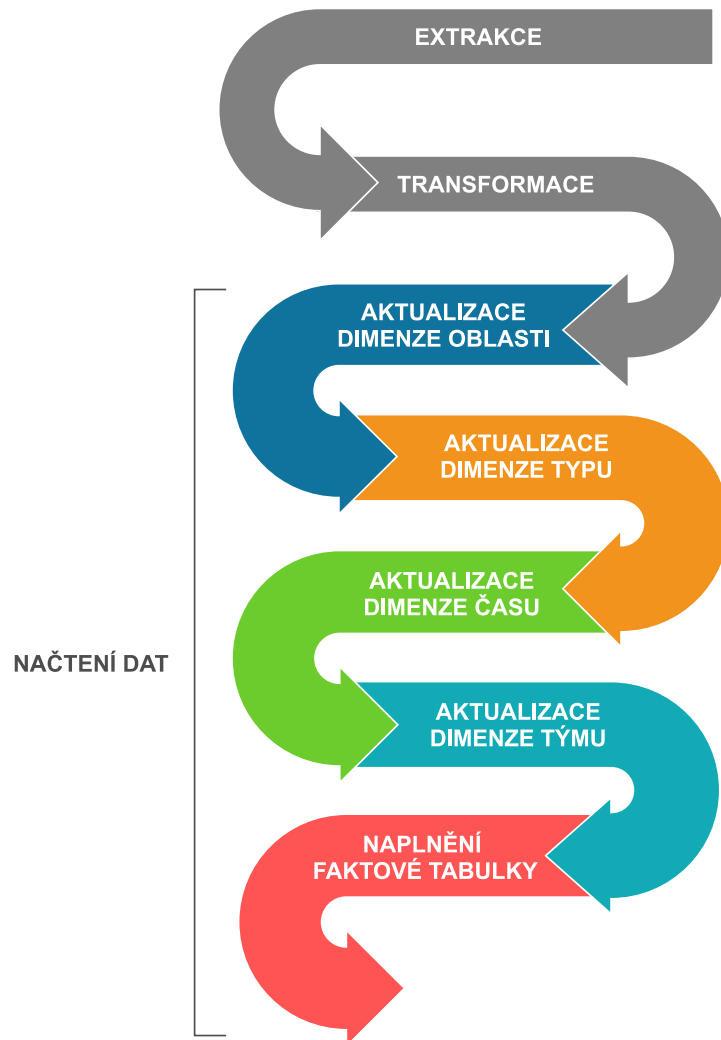
AREA:
  Name: „Brake System“,
  Order: 1,
  Assemblies: {
    ASSEMBLY:
      Name: „Brake Assembly“,
      Number: „A1001“,
      Components: {
        COMPONENT:
          Name: „Brake Assembly“,
          Order: 0,
          Quantity: 13,
          Version: 1,
          ComponentPictureExists: true,
          Materials: {
            COST: { Price: 130, Quantity: 26 }
          }
        COMPONENT:
          Name: „Brake Part“,
          Order: 1,
          Quantity: 3,
          Version: 3,
          ComponentPictureExists: false,
          Processes: {
            COST: { Price: 9, Quantity: 3 }
          }
        }
      }
  }

```

#### Příklad č. 7.8: Ukázka cílových dat

## Načtení dat

Transformovaná data jsou vstupem posledního hlavního procesu ETL, kterým je načtení dat do datového skladu. Tento krok se skládá z pěti podprocesů (viz Obrázek č. 7.6). Třída, ve které jsou všechny tyto podprocesy implementovány, se nazývá *Loader*. První čtyři podprocesy se věnují aktualizaci všech dimenzionálních tabulek datového skladu, a poslední se zabývá samotným plněním dat do tabulky faktů.



Obrázek č. 7.6: Proces načtení dat v rámci ETL procesu

Aktualizací je zde myšlen proces, v rámci kterého se kontroluje, zda daný záznam v dimenzionální tabulce již existuje. V případě, že daný záznam v dimenzi neexistuje, je následně do této tabulky nově přidán.

V rámci aktualizace souhrnné dimenze oblasti vozidla, jsou aktualizovány tyto konkrétní tabulky dimenzí:

```
[dbo].[AreaDimension]  
[dbo].[AssemblyDimension]  
[dbo].[ComponentDimension]
```

Při aktualizaci dimenze týkající se typu nákladů jsou aktualizovány tabulky:

```
[dbo].[CategoryDimension]
[dbo].[TypeDimension]
```

U dimenze týmu jsou aktualizovány tabulky:

```
[dbo].[ContinentDimension]
[dbo].[CountryDimension]
[dbo].[TeamDimension]
```

U časové dimenze je pak aktualizována jediná tabulka:

```
[dbo].[TimeDimension]
```

Při aktualizaci všech dimenzí se postupně do všech objektů `COST` zapisují cizí klíče k odpovídajícím záznamům v tabulkách:

```
[dbo].[ComponentDimension]
[dbo].[TypeDimension]
[dbo].[TimeDimension]
[dbo].[TeamDimension]
```

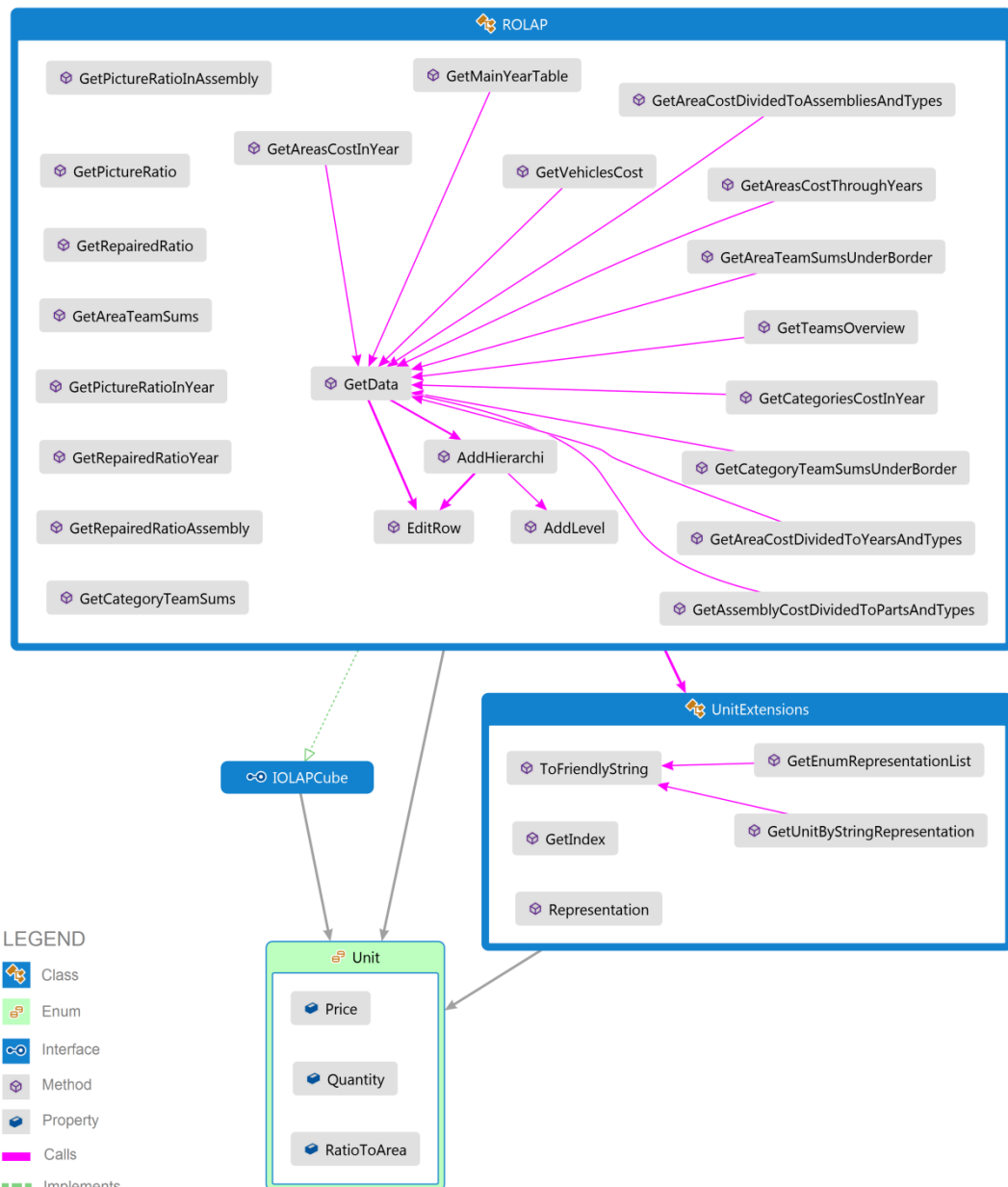
Po aktualizaci těchto dimenzí následuje plnění tabulky faktů `[dbo].[CostFacts]`. Do této tabulky jsou přidávány informace z objektu `COST`.

Počet nových záznamů do této tabulky se v průměru pohybuje nad 2200. Pro plnění tabulky těmito daty byla použita vestavěná třída `System.Data.SqlClient.SqlBulkCopy`, kterou .NET platforma poskytuje. Ta umožňuje efektivní a hromadné uložení dat do tabulky databázového systému SQL Server. Dle zdroje [19] je použití `SqlBulkCopy` při vkládání množiny záznamů rychlejší nežli opakované použití příkazu `INSERT` pro každý záznam množiny. Tento rozdíl ve výkonu je znatelný již při vkládání 1000 záznamů do tabulky. Zdroj uvádí, že vložení 1000 testovacích záznamů příkazem `INSERT` trvalo 205 milisekund a použitím `SqlBulkCopy` pouhých 17 milisekund.

Vložení dat do faktové tabulky datového skladu nám celý proces načtení dat končí a s ním i celý proces ETL.

## 7.2.3 OLAP

Technologie zvaná OLAP (viz kapitola 3.3.1) byla implementována jako třída *ROLAP*, která se dotazuje nad datovým skladem prostřednictvím uložených procedur a funkcí na databázovém systému (viz kapitoly 7.1.1 a 7.1.2). Tato třída implementuje rozhraní *IOLAPCube* (viz Obrázek č. 7.7), přes které aplikační vrstva *ViewModel* získává data (viz kapitola 7.2.1).



Obrázek č. 7.7: Implementace OLAP (Code Map diagram)

OLAP kostka je zde tedy vytvořena prostřednictvím jazyka SQL, konkrétně použitím klauzule *CUBE*, případně *ROLLUP* ve zmíněných uložených procedurách a funkcích. Pomocí těchto klauzulí získáme multidimenzionální přehled všech možných



(či vybraných) dimenzí a můžeme takto provést sumarizaci a analýzu dat současně podle více zadaných kritérií [20].

Všechny metody vně třídy *ROLAP*, které získávají data z datového skladu, volají metodu *GetData* (viz Obrázek č. 7.7), která zajišťuje čtení dat z předaného objektu *SQLDataReader* a zároveň mapuje získaná data dané entitní třídy, které pak jsou pak návratovou hodnotou této metody.

V rámci implementace OLAP byl vytvořen výčtový typ *Unit*, který slouží pro určení typu ukazatele (viz kapitola 4.2.2). Označuje tedy měrnou jednotku, přes kterou budou data uloženými procedurami a funkcemi agregována.

## 7.2.4 Použité knihovny

V rámci implementace aplikační vrstvy byly použity následující knihovny:

*EntityFramework*

**Licence:** Microsoft License

**Popis:** ORM framework, který vývojářům na platformě .NET umožňuje pracovat s databází s použitím objektů .NET.

*MStest*

**Licence:** Microsoft License

**Popis:** Sada nástrojů pro efektivní testování softwaru

*InteractiveDataDisplay.WPF*

**Licence:** Microsoft License

**Popis:** Sada ovládacích prvků WPF pro přidávání interaktivní vizualizace dynamických dat do aplikace. Umožňuje vytvářet liniové grafy, bublinové grafy, tepelné mapy a další složité 2D grafy, které jsou ve vědeckém softwaru velmi běžné.

*WFPieChart*

**Licence:** CPOL License

**Popis:** Knihovna pro vytvoření interaktivního výsečového grafu

*7-zip LZMA*

**Licence:** GNU LGPL

**Popis:** Knihovna pro komprimaci souborů.

*ByteSize*

**Licence:** MIT License

**Popis:** ByteSize je třída nástrojů, která usnadňuje reprezentaci velikosti bajtů v kódu.

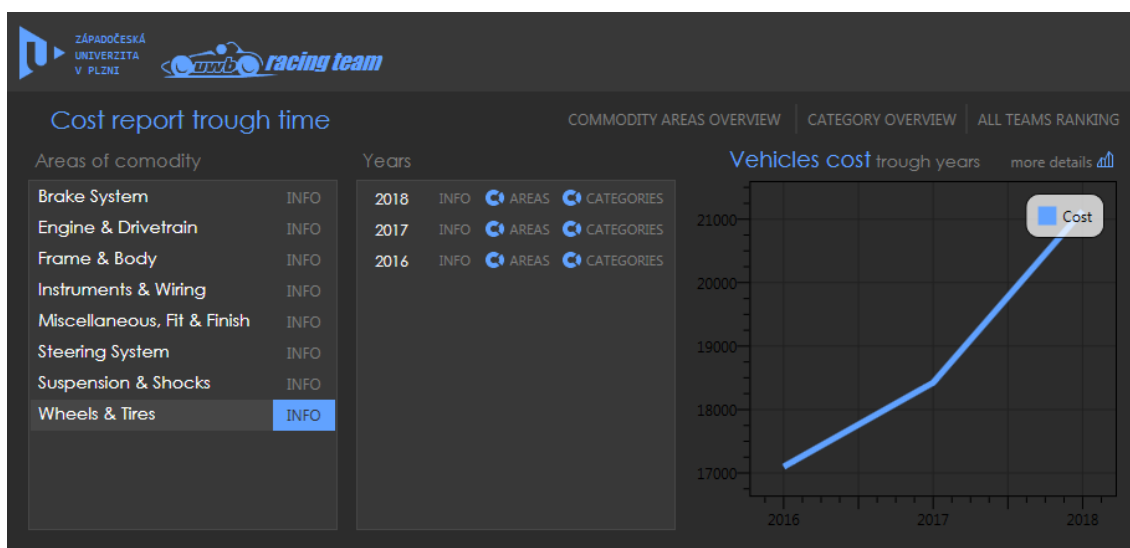
*ExcelFormulaParser*

**Licence:** MIT License

**Popis:** Knihovna pro vypočtení hodnoty dle EXCEL vzorce

## 7.2.5 Uživatelské rozhraní

V této kapitole rozebereme některá okna uživatelské rozhraní, která jsou stěžejní pro prezentační vrstvu našeho celkového řešení. Nejprve se podíváme na část aplikace určenou pro prohlížení historických dat. Zde figuruje jako hlavní okno této části okno popsané v kapitole 5.2, jehož výsledná podoba je zachycena na Obrázku č. 7.8.



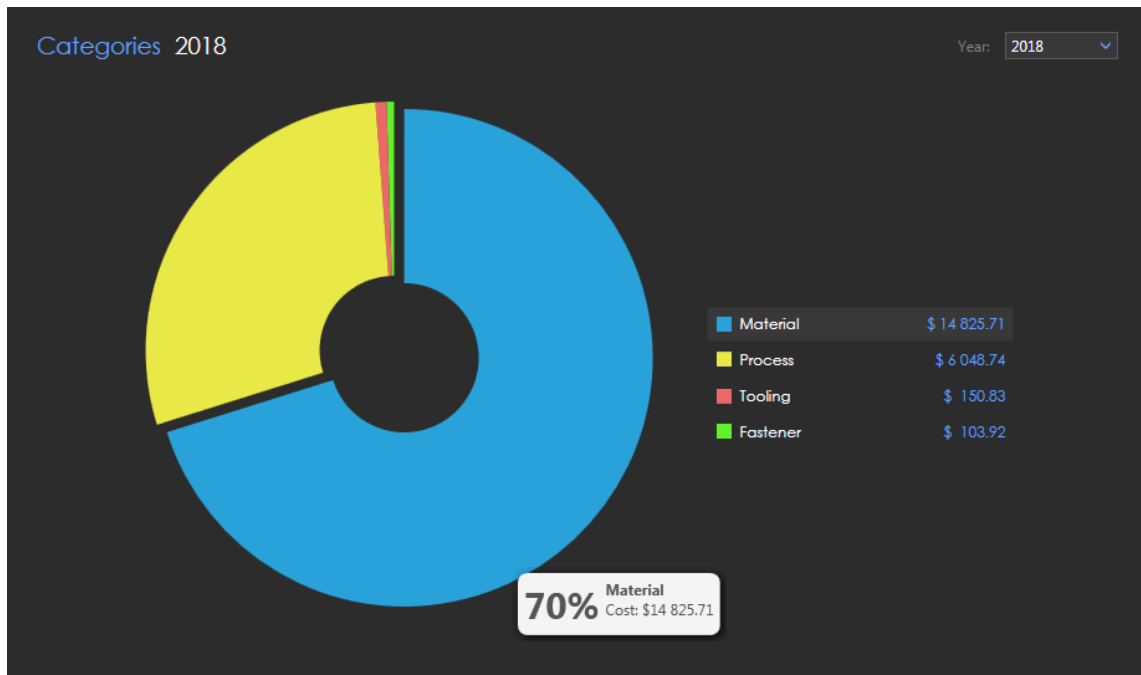
Obrázek č. 7.8: Hlavní okno aplikační části určené pro historická data

Toto okno tedy obsahuje výčet všech let, za která v datovém skladu existují data o nákladech vytvořených vozidel. U každého roku si lze zobrazit přehled nákladů, které jsou děleny dle nákladových kategorií, a to kliknutím na tlačítko „CATEGORIES“. Tento přehled nákladů se pak zobrazí v okně „Categories in 20xx“, kde se nachází zmíněná data v koláčovém grafu (viz Obrázek 7.9). Dále si lze pro konkrétní rok zobrazit přehled nákladů tentokrát dělený dle oblastí komodit. Opět po kliknutí na tlačítko „AREAS“ v řádku daného roku se zobrazí okno „Areas of Commodity in 20xx“, kde se taktéž nachází data v koláčovém grafu (viz Obrázek č. 7.10). Po kliknutí na tlačítko „INFO“ v řádku konkrétního roku se zobrazí tabulka popisující náklady pro daný rok.

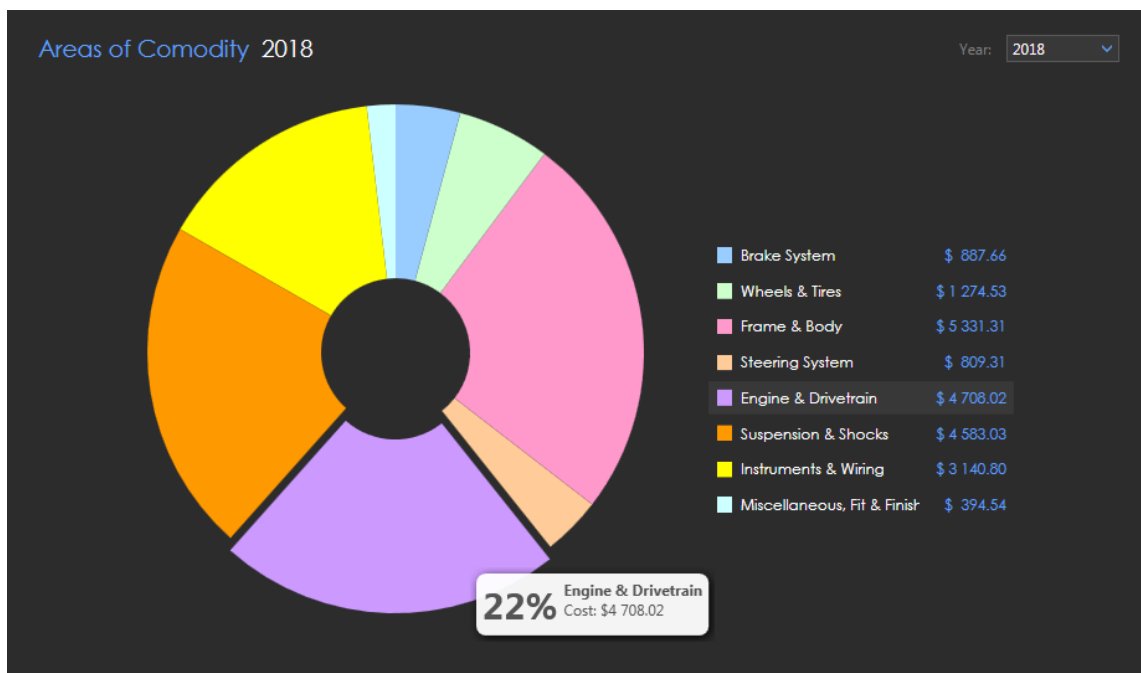
V obou výše zmíněných oknech s koláčovými grafy, lze u každé výseče přejetím myši zjistit detailnější informace o dané nákladové položce (podíl v %, název a částka). Pro vizualizaci a koláčového grafu byla použita třída *BaseWPFHelper.cs*, která zajišťuje uživatelsky přívětivé prohlížení jednotlivých výsečí grafu.

Dále je v tomto okně obsažen výčet všech oblastí komodit vozidla. U každé oblasti existuje tlačítko „INFO“, které slouží pro zobrazení okna s bližšími informacemi o nákladech dané oblasti v průběhu let. Toto okno lze vidět na Obrázku č. 7.11. Z tohoto okna pak uživatel může být přesměrován na veškerá okna s přehledy týkající se vybrané oblasti, které jsou blíže popsány v kapitolách 2.2.2, 2.2.3 a 2.2.4. V těchto oknech se nacházejí tabulky, které jsou založeny na podobném principu. Aby bylo uživateli umožněno měnit úroveň podrobnosti zobrazených dat, jsou tyto tabulky

implementovány jako tzv. Tree View (viz Obrázek č. 7.11 a Obrázek č. 7.12). Tím pak uživatel využívá OLAP funkcionalit jako jsou *Roll-up* a *Drill-down* (viz kapitola 3.3.1). Dále pak například v okně na Obrázku č. 7.11 si uživatel může vybrat rozsah sloupců (rozsah let), které se mají v tabulce zobrazit. To se opět týká OLAP funkcionalit, konkrétně funkce *Slice* (viz kapitola 3.3.1).



Obrázek 7.9: Přehled nákladů daného roku (dělení dle nákladových kategorií)



Obrázek č. 7.10: Okno pro přehled nákladů daného roku (dělení dle oblasti komodit)

A1 Brake System

Number of highest values per year:  Range:  to  Unit:

	2018	2017	2016
<b>FASTENER</b>	<b>1.68</b>	<b>9.18</b>	<b>13.18</b>
Bolt, Grade 10.9 (SAE 8)	1.21	1.20	1.20
Retaining Ring, R-ring	0.31	0.70	0.70
Tie Wrap	-	7.20	11.20
Washer, Grade 8.8 (SAE 5)	0.16	0.08	0.08
<b>MATERIAL</b>	<b>766.27</b>	<b>517.87</b>	<b>500.57</b>
Adapter/L.P./Barb to Male Pipe/Straight/Aluminum/Anodized	3.68	3.68	3.68
Adapter/L.P./Elbow/FeMale to Male Pipe/45 deg./Aluminum/Anodized	-	86.52	-
Adapter/L.P./Elbow/FeMale to Male Pipe/90 deg./Aluminum/Anodized	40.31	-	86.52
Adapter/L.P./Male Flare to Pipe//Aluminum/Nickel-Plated	18.88	18.88	18.88
Adapter/L.P./Union Tee//Brass/	5.80	16.42	16.42
Banjo Bolt, Aluminum	12.92	12.92	12.92
Banjo Fitting, Aluminium	-	23.00	23.00
Banjo Fitting, Aluminium	23.00	-	-
Brake Caliper, ISR, 22-048	192.00	96.00	96.00
Brake Caliper, ISR, 22-049	194.00	97.00	97.00
Brake Pad, Iron or Steel Rotor	13.15	6.58	2.19
Cable Tie Mount, Adhesive Backed	2.70	-	-
Fitting/L.P./Straight/Aluminum/Anodized	184.76	116.28	103.36
Hose, High Pressure, Stainless Steel Braided Outer	62.19	27.73	27.73
Sensor, Fluid Pressure	8.00	8.00	8.00
Steel, Alloy	4.88	-	-

All displayed values are in \$

Obrázek č. 7.11: Okno pro přehled nákladů oblasti komodit během let

A1 Brake System

Number of highest values:  Year:  Unit:

	A1001	A1002	A1003	A1004	A1005
<b>FASTENER</b>	<b>0.35</b>	<b>0.35</b>	<b>0.64</b>	<b>0.64</b>	<b>7.20</b>
Bolt, Grade 10.9 (SAE 8)	-	-	0.60	0.60	-
Retaining Ring, R-ring	0.35	0.35	-	-	-
Tie Wrap	-	-	-	-	7.20
Washer, Grade 8.8 (SAE 5)	-	-	0.04	0.04	-
<b>MATERIAL</b>	<b>2.51</b>	<b>2.35</b>	<b>100.39</b>	<b>99.19</b>	<b>313.43</b>
Adapter/L.P./Barb to Male Pipe/Straight/Aluminum/Anodized	-	-	-	-	3.68
Adapter/L.P./Elbow/FeMale to Male Pipe/45 deg./Aluminum/Anodized	-	-	-	-	86.52
Adapter/L.P./Male Flare to Pipe//Aluminum/Nickel-Plated	-	-	-	-	18.88
Adapter/L.P./Union Tee//Brass/	-	-	-	-	16.42
Banjo Bolt, Aluminum	-	-	-	-	12.92
Banjo Fitting, Aluminium	-	-	-	-	23.00
Brake Caliper, ISR, 22-048	-	-	96.00	-	-
Brake Caliper, ISR, 22-049	-	-	-	97.00	-
Brake Pad, Iron or Steel Rotor	-	-	4.39	2.19	-
Fitting/L.P./Straight/Aluminum/Anodized	-	-	-	-	116.28
Hose, High Pressure, Stainless Steel Braided Outer	-	-	-	-	27.73
Sensor, Fluid Pressure	-	-	-	-	8.00
Steel, Alloy (per kg)	0.22	0.22	-	-	-
Steel, Stainless (per kg)	2.18	2.02	-	-	-
Washer, Grade 8.8 (SAE 5)	0.12	0.12	-	-	-
<b>PROCESS</b>	<b>26.16</b>	<b>25.56</b>	<b>10.16</b>	<b>10.16</b>	<b>30.58</b>

Components having picture: **30.77 %**

Repaired components: **0.00 %**

All displayed values are in \$

Obrázek č. 7.12: Okno s přehledem nákladů oblasti (dělené dle sestav oblasti)

## 8 Testování

Při testování implementovaného řešení byly použity tři druhy testů:

- Testování jednotek (Unit testing)
- Integroční testy
- Testovací scénáře

Při testování jednotek u objektově orientovaného programování se jedná o testování jednotlivých metod konkrétních tříd.

U integračního testování probíhá kontrola komunikace mezi dvěma komponentami nebo i různými systémy. U všech integračních testů byla přidána anotace s kategorií testu, díky čemuž lze filtrovat testy, které mají být spuštěny. U testů byla vždy přidána kategorie s názvem „Integration“, což umožňuje spouštět testy pouze integrační či naopak všechny ostatní.

Tyto dva druhy testů jsou zahrnuty v *CostReportUWBRacingTeam.Tests*, což je testovací projekt, který pro testování používá framework *MSTest*. Ten je jedním z nejpoužívanějších .NET testovacích frameworků.

Aplikace byla otestována integračními a jednotkovými testy s více než 40 % pokrytím kódu. Otestovány byly stěžejní funkcionality, mezi které patří především:

- Extrakce dat z operační databáze do datového skladu (ETL proces)
- Získávání dat z datového skladu (OLAP technologie)
- Analytický nástroj pro vyhledávání nejnákladnějších položek
- Operace s daty operační databáze
- Výpočet částky dle MS EXCEL vzorce (knihovna *ExcelFormulaPraser*)

### 8.1 ETL

Za účelem testování datové pumpy (viz kapitola 3.1.1) byl vytvořen jak testovací datový sklad se stejnou datovou strukturou, uloženými funkcemi a procedurami tak itestovací operační databáze. Otestovány byly postupně všechny tři stěžejní procesy tedy extrakce, transformace a načtení dat.

#### 8.1.1 Extrakce

Při testování extrakce byla testovací data uložena do testovací operační databáze a poté bylo porovnáváno, zda extrahovaná data jsou shodná s daty testovacími. Zde jelikož se pracovalo s databází, byl proveden integrační test (viz úvod kapitoly 8).

Abychom zajistili, že se testovací třída připojí k testovací operační databázi, byl použit tzv. Mock objekt. Použitím Mock objektů se jedná o techniku nahrazení reálného objektu testovací fasádou, která žádné funkcionality nahrazeného objektu neprovádí.

Pouze se jako nahrazený objekt tváří a lze u ní nastavit chování, jaké v dané situaci potřebujeme. V tomto případě byla takto nahrazena třída *ConfigurationManager*, která v architektuře aplikace zajišťuje získání tzv. *ConnectionString*. Jedná se o řetězec obsahující informace o nastavení připojení k nějakému zdroji dat. U testovací fasády bylo nastaveno, že pokud bude nad třídou volána funkcionální vracející řetězec s připojením k operační databázi, bude vždy vrácen řetězec s připojením k testovací operační databázi.

### 8.1.2 Transformace

Při testování transformačního procesu bylo použito jednotkové testování, kdy byla připravena testovací data, která pak byla porovnávána s transformovanými daty.

### 8.1.3 Načtení dat

Pro testování procesu načtení dat byly vytvořeny opět integrační testy, kdy se tentokrát testovala komunikace s datovým skladem. Zde byly otestovány čtyři stěžejní podprocesy načtení (viz kapitola 8.1.3) nezávisle na sobě. Podobně jako u extrakce byl vytvořen Mock objekt nahrazující třídu *ConfigurationManager*, který tentokrát vracel řetězec s připojením k testovacímu datovému skladu.

U každého podprocesu načtení byla vytvořena testovací data jako vstup daného podprocesu a poté byl vždy zkontrolován stav dat v testovacím datovém skladu.

## 8.2 OLAP

Veškeré metody implementované v rámci OLAP, byly otestovány integračními testy. Byly otestovány tudíž i uložené procedury a funkce, které byly implementovány v datové vrstvě celkového řešení (viz kapitoly 7.1.1 a 7.1.2) a jsou využívány OLAP technologií. Byly vytvořeny testovací data, kterými je při testování naplněn testovací datový sklad, a po jejichž provedení jsou následně porovnány získané výsledky s očekávanými daty.

Na základě těchto testů byly odhaleny chyby v uložených funkcích, kde byla zjištěna chybná deklarace návratové hodnoty. Jedná se konkrétně o následující uložené funkce:

```
[dbo].[GetPictureRatioInYear]
[dbo].[GetPictureRatioInArea]
[dbo].[GetPictureRatioInAssembly]
[dbo].[GetRepairedRatioInYear]
[dbo].[GetRepairedRatioInArea]
[dbo].[GetRepairedRatioInAssembly]
```

## **8.3 Uživatelské rozhraní**

Za účelem otestování uživatelského rozhraní aplikace byly vytvořeny testovací scénáře, které jsou obsaženy jak v příloze této práce (viz Příloha A), tak i v souboru „*testscenario.docx*“ v adresáři „*attachment*“ na přiloženém CD.

## 9 Závěr

V rámci této práce byl proveden sběr požadavků a analýza dat zadavatele. Další součástí práce je teoretická část, která rozebírá pojem Business Intelligence. V té je obsažen popis všech jeho hlavních komponent, které jsou pro tuto práci nezbytné, tedy například datový sklad, OLAP technologie či ETL proces. Na základě dat a požadavků zadavatele byl pak sestaven návrh datové a aplikační vrstvy celkového řešení, dle čehož byla následovně provedena implementace.

Do datové vrstvy celkového řešení byla kromě datového skladu zahrnuta také implementace operační databáze, která uchovává data za aktuální rok. Ta slouží jako zdrojový systém, ze kterého datový sklad získává data ETL procesem (viz kapitola 8.1). Aplikace tudíž byla rozšířena o část, zajišťující tvorbu a editaci těchto zdrojových dat, včetně automatizace výpočtu cen jednotlivých nákladových položek. Tím byl z velké části eliminován lidský faktor a snížila se tak chybovost ve výpočtu celkového nákladu vytvořeného vozidla. Ta se v předcházejících letech projevila zvýšením celkových částek v průměru o necelých 10%, než činily skutečné náklady. Na základě implementovaného řešení lze však předpokládat, že se chybovost v budoucnu sníží až na nulu.

Pro umožnění analýzy a porovnání dat v datovém skladu byly v aplikaci implementovány přehledové tabulky a grafy, jejichž účelem je podpora strategického rozhodování studentského týmu. Aplikace uživatelům pomáhá zejména v oblasti snižování nákladů, kdy pro přehledové tabulky byl implementován analytický nástroj, jenž usnadňuje vyhledávání nejnákladnějších položek.

Dále tato práce figuruje také jako tzv. *Proof of concept* (viz kapitoly 4.2.5 a 5.3), kdy v případě že organizace SAE International plně zpřístupní informace o nákladech všech týmů, které se soutěže účastní, lze prostřednictvím implementovaného řešení provést analýzu nákladů napříč všemi týmy.



## Zdroje

- [1] **SAE INTERNATIONAL.** *SAE International.* [Online] [Citace: 29. Srpen 2018.] <https://www.sae.org/>.
- [2] **SAE INTERNATIONAL.** Series Resources. *FORMULA SAE.* [Online] [Citace: 23. Zář 2018.] <https://www.fsaeonline.com/cdsweb/gen/DocumentResources.aspx>.
- [3] **NOVOTNÝ, O., POUR, J., SLÁNSKÝ, D.** *Business Intelligence – Jak využít bohatství ve vašich datech.* 1. vyd. Praha : Grada publishing, a. s., 2005. ISBN 80-247-1094-3.
- [4] **INMON, William H.** *Building the Data Warehouse.* 4. vyd. New York : John Wiley & Sons, 2005. ISBN 0-7645-9944-5.
- [5] **HUMPRIES M., kol.** *Data Warehousing – návrh a implementace.* 1. vyd. Praha : Computer Press, 2001. ISBN 80-7226-560-1.
- [6] **LACKO, L.** *Datové sklady, OLAP a dolování dat s příklady v SQL Serveru a Oracle.* 1. vyd. Brno : Computer Press, 2003. ISBN 80-7226-969-0.
- [7] **GÁLA, L., POUR, J., ŠEDIVÁ, Z.** *Podniková informatika.* 2., přeprac. a aktualiz. vyd. Praha : Grada, 2009. ISBN 978-80-247-2615-1.
- [8] **KLADIVO, P.** *Základy statistiky.* 1. vyd. Olomouc : Univerzita Palackého v Olomouci, 2013. ISBN 978-80-244-3841-2.
- [9] **SOLID IT GmbH.** DB-Engines Ranking. *DB-Engines.* [Online] Solid IT GmbH, 1. Srpen 2018. [Citace: 10. Zář 2018.] <https://db-engines.com/en/ranking>.
- [10] **G2 CROWD.** Best Relational Databases Software. *G2 Crowd.* [Online] G2 Crowd, 11. Zář 2019. [Citace: 11. Zář 2019.] <https://www.g2.com/categories/relational-databases>.
- [11] **WALKER, A.** 18 Best Open-Source and Free Database Software. *G2 Crowd.* [Online] G2 Crowd, 5. Prosinec 2017. [Citace: 12. Zář 2018.] <https://learn.g2crowd.com/free-database-software>.
- [12] **PETERSEN, T., MILENER, G., GUYER, C., ROTH, J.** Features Supported by the Editions of SQL Server 2014. *Microsoft.* [Online] Microsoft Corporation, 24. Květen 2017. [Citace: 26. Zář 2019.] <https://docs.microsoft.com/en-us/sql/getting-started/features-supported-by-the-editions-of-sql-server-2014>.
- [13] **WENZEL, M., LATHAM, L., WAGNER, B., LEVIN I. a další.** Introduction to the C# language. *Microsoft.* [Online] Microsoft Corporation, 20. Červenec 2015. [Citace: 1. Zář 2018.] <https://docs.microsoft.com/en-ca/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>.

- [14] **KENNEDY, J., SCHOFIELD, M., SATRAN, M.** Choose your technology. *Microsoft*. [Online] Microsoft, 3. Prosinec 2018. [Citace: 27. Listopad 2018.] <https://docs.microsoft.com/en-us/windows/desktop/choose-your-technology>.
- [15] **WENZEL, M., SMITH, S., HUANG, J., WAGNER, B., JENKS, A., JONES, M., LATHAM, L.** Architectural principles. *Microsoft*. [Online] Microsoft Corporation, 16. Únor 2019. [Citace: 22. Únor 2019.] <https://docs.microsoft.com/en-us/dotnet/standard/modern-web-apps-azure-architecture/architectural-principles>.
- [16] **VEGA, D.** Entity Framework 6. *Microsoft*. [Online] Microsoft Corporation, 23. Říjen 2016. [Citace: 2. Listopad 2018.] <https://docs.microsoft.com/cs-cz/ef/ef6/>.
- [17] **ROTH J., LOPES P., MILENER G., a další.** User-Defined Functions. *Microsoft*. [Online] Microsoft Corporation, 5. Srpen 2016. [Citace: 23. Říjen 2018.] <https://docs.microsoft.com/cs-cz/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-2017>.
- [18] **BRITCH, D.** Vzor Model-View-ViewModel. *Microsoft*. [Online] Microsoft Corporation, 7. Srpen 2017. [Citace: 6. Únor 2019.] <https://docs.microsoft.com/cs-cz/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>.
- [19] **GVIRTZ, S.** SQL Server for developers: Improve Data loading performance using Bulk Insert & SqlBulkCopy. *Shahar Gvirtz's Weblog*. [Online] Shahar Gvirtz, 25. červen 2017. [Citace: 1. prosinec 2018.] <https://weblogs.asp.net/shahar/sql-server-for-developers-improve-data-loading-performance-using-bulk-insert-sqlbulkcopy>.
- [20] **LACKO, L.** *Business Intelligence v SQL Serveru 2005: Reportovací, analytické a další datové služby*. Brno : Computer Press, a. s., 2006. ISBN 80-251-1110-5.
- [21] **GÁLA, L., POUR, J., TOMAN, P.** *Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi, technologie informačních systémů, řízení a rozvoj podnikové informatiky*. 1. vyd. Praha : Grada, 2006. ISBN 80-247-1278-4.

## Seznam příloh

A	Testovací scénáře uživatelského rozhraní .....	91
A.1	Otevření aplikační části s historickými daty .....	91
A.2	Zobrazení liniového grafu s přehledem nákladů .....	92
A.3	Zobrazení přehledu nákladu kategorií za rok .....	93
A.4	Změna dat v přehledu nákladů kategorií za rok .....	94
A.5	Zobrazení okna s přehledem nákladů oblastí za rok .....	95
A.6	Změna dat v přehledu nákladů oblastí za rok .....	96
A.7	Zobrazení přehledu nákladů za rok .....	97
A.8	Změna dat v přehledu nákladů za rok .....	98
A.9	Zobrazení přehledu nákladů za oblast komodit v průběhu let .....	99
A.10	Změna dat v přehledu nákladů za oblast komodit v průběhu let .....	100
A.11	Změna dat v přehledu nákladů za oblast komodit v průběhu let .....	101
A.12	Změna dat v přehledu nákladů za oblast komodit v průběhu let .....	102
A.13	Zobrazení přehledu nákladů konkrétní oblasti pro konkrétní rok .....	103
A.14	Změna dat v přehledu nákladů konkrétní oblasti pro konkrétní rok .....	104
A.15	Změna dat v přehledu nákladů za oblast komodit pro konkrétní rok .....	105
A.16	Změna dat v přehledu nákladů za oblast komodit pro konkrétní rok .....	106
A.17	Zobrazení okna konkrétní sestavy .....	107
A.18	Změna zobrazených dat v okně konkrétní sestavy .....	108
A.19	Změna zobrazených dat v okně konkrétní sestavy .....	109
A.20	Změna zobrazených dat v okně konkrétní sestavy .....	110
A.21	Zobrazení okna s analýzou kategorií nákladů .....	111
A.22	Změna dat v okně s analýzou kategorií nákladů .....	112
A.23	Změna dat v okně s analýzou kategorií nákladů .....	113
A.24	Změna dat v okně s analýzou kategorií nákladů .....	114
A.25	Zobrazení okna s analýzou oblastí komodit .....	115
A.26	Změna dat v okně s analýzou oblastí komodit .....	116
A.27	Změna dat v okně s analýzou oblastí komodit .....	117
A.28	Změna dat v okně s analýzou oblastí komodit .....	118
A.29	Otevření aplikační části s aktuálními daty .....	119
A.30	Zobrazení sestav konkrétní oblasti .....	120
A.31	Otevření okna s detaily sestavy .....	121

A.32	Rezervace sestavy.....	122
A.33	Přepnutí sestavy do editovacího módu.....	123
A.34	Přepnutí sestavy do módu prohlížení.....	124
A.35	Zobrazení nákladových položek kategorie sestavy.....	125
A.36	Úprava informací sestavy.....	126
A.37	Změna pořadí sestavy.....	127
A.38	Změna pořadí sestavy.....	128
A.39	Vytvoření sestavy.....	129
A.40	Vytvoření nové části sestavy.....	130
A.41	Otevření části sestavy.....	131
A.42	Přidání materiálu sestavě.....	132
A.43	Přidání procesu sestavě.....	133
A.44	Přidání spojovacího materiálu sestavě.....	134
A.45	Přidání opracování sestavě.....	135
A.46	Editace nákladové položky v rámci sestavy.....	136
A.47	Editace pořadí nákladové položky v rámci sestavy.....	137
A.48	Odstranění nákladové položky v rámci sestavy.....	138
A.49	Odstranění sestavy.....	139
A.50	Import souborů s typy nákladu.....	140
A.51	Export obrázků sestav.....	141
A.52	Spuštění ETL procesu.....	142
A.53	Export zprávy o nákladech.....	143

# A Testovací scénáře uživatelského rozhraní

## A.1 Otevření aplikační části s historickými daty

<b>ID :</b>	#1
<b>Název:</b>	Otevření aplikační části s historickými daty
<b>Popis:</b>	Testovací scénář pro zobrazení hlavního okna aplikační části pro prohlížení historických dat.
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Spustí aplikaci</li><li>2. Klikni na tlačítko „<i>Show historical data</i>“</li></ol>
<b>Očekávaný výsledek:</b>	Zobrazení okna s názvem „ <i>Historical Data Overview</i> “

Testovací scénář A. 1

## A.2 Zobrazení liniového grafu s přehledem nákladů

<b>ID :</b>	#2
<b>Název:</b>	Zobrazení liniového grafu s přehledem nákladů
<b>Popis:</b>	Testovací scénář pro zobrazení okna s liniovým grafem, kde je vykreslen vývoj nákladů dílčích oblastí vozidla v průběhu let.
<b>Podmínky:</b>	Otevřené okno s názvem „ <i>Historical Data Overview</i> “ (hlavní okno části aplikace s přehledem historických dat)
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Klikni na tlačítko „<i>more details</i>“ (umístěné vpravo nad liniovým grafem „<i>Vehicle cost trough years</i>“)</li></ol>
<b>Očekávaný výsledek:</b>	Zobrazení okna s názvem „ <i>Cost Trough Years</i> “

Testovací scénář A. 2

## A.3 Zobrazení přehledu nákladu kategorií za rok

<b>ID :</b>	#3
<b>Název:</b>	Zobrazení okna s koláčovým grafem poskytující přehled nákladů kategorií
<b>Popis:</b>	Testovací scénář pro zobrazení okna s koláčovým grafem s náklady konkrétního roku rozlišovaných dle nákladových kategorií
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017</li><li>• Otevřené okno s názvem „<i>Historical Data Overview</i>“ (hlavní okno části aplikace s přehledem historických dat).</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší v tabulce „<i>Years</i>“ na řádek začínající číslem „2017“</li><li>2. Klikni na tlačítko „<i>CATEGORIES</i>“ umístěné v daném řádku</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Zobrazení okna s názvem „<i>Categories in 2017</i>“</li><li>• Zobrazení koláčového grafu v okně „<i>Categories in 2017</i>“ s daty za rok 2017</li><li>• V kombinovaném poli v pravém horním rohu okna „<i>Categories in 2017</i>“</li></ul>

Testovací scénář A. 3

## A.4 Změna dat v přehledu nákladů kategorií za rok

<b>ID :</b>	#4
<b>Název:</b>	Změna zobrazených dat v okně s koláčovým grafem poskytující přehled nákladů kategorií
<b>Popis:</b>	<p>Okno „Categories in 20xx“:</p> <p>Testovací scénář pro změnu dat v okně koláčovým grafem, který zobrazuje náklady konkrétního roku rozlišovaných dle nákladových kategorií. V tomto scénáři se změní daný rok, na základě kterého se mají změnit i zobrazená data.</p>
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za roky 2017 a 2018</li><li>• Otevřené okno s názvem „Categories in 2017“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na kombinované pole „Year“</li><li>2. Kliknutím otevři rozbalovací menu kombinovaného pole</li><li>3. Kliknutím vyber hodnotu „2018“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Název okna se změnil z „Categories in 2017“ na „Categories in 2018“</li><li>• V kombinovaném poli je zobrazena hodnota „2018“</li><li>• V koláčovém grafu v okně se zobrazí data za rok 2018</li></ul>

Testovací scénář A. 4



## A.5 Zobrazení okna s přehledem nákladů oblastí za rok

<b>ID :</b>	#5
<b>Název:</b>	Zobrazení okna s koláčovým grafem poskytující přehled nákladů oblastí komodit
<b>Popis:</b>	Testovací scénář pro zobrazení okna „ <i>Areas Of Commodity in 20xx</i> “, kde se nachází koláčový graf s náklady za konkrétní rok dělené dle oblastí komodit.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017</li><li>• Otevřené okno s názvem „<i>Historical Data Overview</i>“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší v tabulce „<i>Years</i>“ na řádek začínající číslem „2017“</li><li>2. Klikni na tlačítko „<i>AREAS</i>“ umístěné v daném řádku</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Zobrazení okna s názvem „<i>Areas Of Commodity in 2017</i>“</li><li>• V kombinovaném poli je zobrazena hodnota „2017“</li><li>• V koláčovém grafu v okně jsou zobrazena data za rok 2017</li></ul>

Testovací scénář A. 5

## A.6 Změna dat v přehledu nákladů oblastí za rok

<b>ID :</b>	#6
<b>Název:</b>	Změna zobrazených dat v okně „ <i>Areas of Commodity in 20xx</i> “
<b>Popis:</b>	<p>Testovací scénář pro změnu dat v okně, kde jsou zobrazeny náklady konkrétního roku rozlišené dle oblasti komodit. V tomto scénáři se změní daný rok, na základě kterého se mají změnit i zobrazená data.</p>
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence dat o nákladech v datovém skladu za roky 2017 a 2018</li><li>• Otevřené okno s názvem „<i>Areas Of Commodity in 2017</i>“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na kombinované pole „<i>Year</i>“</li><li>2. Kliknutím otevři rozbalovací menu kombinovaného pole</li><li>3. Kliknutím vyber hodnotu „<i>2018</i>“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Název okna se změnil na „<i>Areas of Commodity in 2018</i>“</li><li>• V kombinovaném poli je zobrazena hodnota „<i>2018</i>“</li><li>• V koláčovém grafu v okně se zobrazí data za rok 2018</li></ul>

Testovací scénář A. 6

## A.7 Zobrazení přehledu nákladů za rok

<b>ID :</b>	#7
<b>Název:</b>	Zobrazení okna s přehledem nákladů všech oblastí komodit a nákladových kategorií
<b>Popis:</b>	Testovací scénář pro zobrazení okna „ <i>Cost Overview 20xx</i> “, kde se nachází přehled s náklady za konkrétní rok dělené dle oblastí komodit a nákladových kategorií.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017</li><li>• Otevřené okno s názvem „<i>Historical Data Overview</i>“ (hlavní okno části aplikace s přehledem historických dat).</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na seznam let „<i>Years</i>“ na řádek začínající číslem 2017</li><li>2. Klikni na tlačítko „<i>INFO</i>“ v řádku</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Zobrazení okna s názvem „<i>Cost Overview 2017</i>“</li><li>• V kombinovaném poli je zobrazena hodnota „<i>2017</i>“</li><li>• V tabulce v okně jsou zobrazena data za rok 2017</li></ul>

Testovací scénář A. 7

## A.8 Změna dat v přehledu nákladů za rok

<b>ID :</b>	#8
<b>Název:</b>	Změna dat v okně s přehledem nákladů všech oblastí komodit a nákladových kategorií
<b>Popis:</b>	Testovací scénář pro změnu dat v okně „ <i>Cost Overview 20xx</i> “, kde se nachází přehled s náklady za konkrétní rok dělené dle oblastí komodit a nákladových kategorií.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017 a 2018</li><li>• Otevřené okno s názvem „<i>Cost Overview 2017</i>“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na kombinované pole „<i>Year</i>“</li><li>2. Kliknutím otevři rozbalovací menu kombinovaného pole</li><li>3. Kliknutím vyber hodnotu „<i>2018</i>“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Název okna se změnil z „<i>Cost Overview in 2017</i>“ na „<i>Cost Overview in 2018</i>“</li><li>• V kombinovaném poli je zobrazena hodnota „<i>2018</i>“</li><li>• V přehledu v okně se zobrazí data za rok 2018</li></ul>

Testovací scénář A. 8

## A.9 Zobrazení přehledu nákladů za oblast komodit v průběhu let

<b>ID :</b>	#9
<b>Název:</b>	Zobrazení okna s přehledem nákladů za oblast komodit v průběhu let
<b>Popis:</b>	Testovací scénář pro zobrazení okna s přehledem nákladů za oblast komodit v průběhu let.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017 a 2018</li><li>• Otevřené okno s názvem „<i>Historical Data Overview</i>“ (hlavní okno části aplikace s přehledem historických dat).</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na řádek „<i>Brake System</i>“ v seznamu „<i>Areas</i>“</li><li>2. Klikni na tlačítko „<i>INFO</i>“ v řádku</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Zobrazení okna „<i>Brake System</i>“</li><li>• V okně jsou data za roky 2017 a 2018</li></ul>

Testovací scénář A. 9

## A.10 Změna dat v přehledu nákladů za oblast komodit v průběhu let

<b>ID :</b>	#10
<b>Název:</b>	Změna dat v okně s přehledem nákladů za oblast komodit v průběhu let
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s přehledem nákladů za oblast komodit v průběhu let. Změna se týká v počtu zvýrazněných nejvyšších hodnot.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017 a 2018</li><li>• Otevřené okno s názvem „Brake System“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Přepiš v textovém poli „<i>Number of highest values per year</i>“ hodnotu 5 na 10</li><li>2. Klikni na tlačítko „<i>Reload</i>“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• V okně jsou data za roky 2017 a 2018</li><li>• V rámci každého sloupce je obarveno 10 nejvyšších hodnot</li></ul>

Testovací scénář A. 10

## A.11 Změna dat v přehledu nákladů za oblast komodit v průběhu let

<b>ID :</b>	#11
<b>Název:</b>	Změna dat v okně s přehledem nákladů za oblast komodit v průběhu let
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s přehledem nákladů za oblast komodit v průběhu let. Změna se týká v rozsahu let, za která jsou data zobrazována.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2016, 2017 a 2018</li><li>• Otevřené okno s názvem „Brake System“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Do prvního textového pole „Range“ zapiš hodnotu 2016</li><li>2. Do druhého textového pole „Range“ zapiš hodnotu 2017</li><li>3. Klikni na tlačítko „Reload“</li></ol>
<b>Očekávaný výsledek:</b>	V okně jsou zobrazena data za roky 2016 a 2017

Testovací scénář A. 11

## A.12 Změna dat v přehledu nákladů za oblast komodit v průběhu let

<b>ID :</b>	#12
<b>Název:</b>	Změna dat v okně s přehledem nákladů za oblast komodit v průběhu let
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s přehledem nákladů za oblast komodit v průběhu let. Změna se týká měrné jednotky zobrazených dat.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2016, 2017 a 2018</li><li>• Otevřené okno s názvem „Brake System“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na kombinované pole „Unit“</li><li>2. Kliknutím zobraz rozbalovací menu</li><li>3. Klikni na hodnotu „pcs“ v rozbalovacím menu</li><li>4. Klikni na tlačítko „Reload“</li></ol>
<b>Očekávaný výsledek:</b>	V okně jsou data týkající se počtu jednotlivých nákladových položek

Testovací scénář A. 12



## A.13 Zobrazení přehledu nákladů konkrétní oblasti pro konkrétní rok

<b>ID :</b>	#13
<b>Název:</b>	Zobrazení okna přehledu nákladů konkrétní oblasti pro konkrétní rok
<b>Popis:</b>	Testovací scénář pro zobrazení okna přehledu nákladů konkrétní oblasti pro konkrétní rok
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2016, 2017 a 2018</li><li>• Otevřené okno s názvem „<i>Brake System</i>“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Klikni na nadpis (tlačítko) sloupce 2017</li></ol>
<b>Očekávaný výsledek:</b>	Zobrazení okna „ <i>Brake System</i> “, kde se nachází náklady pro oblast Brake System za rok 2017

Testovací scénář A. 13

## A.14 Změna dat v přehledu nákladů konkrétní oblasti pro konkrétní rok

<b>ID :</b>	#14
<b>Název:</b>	Změna dat v okně s přehledem nákladů konkrétní oblasti pro konkrétní rok
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s přehledem nákladů konkrétní oblasti pro konkrétní rok. Změna se týká v počtu zvýrazněných nejvyšších hodnot.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2016, 2017 a 2018</li><li>• Otevřené okno s názvem „Brake System“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Přepiš v textovém poli „<i>Number of highest values</i>“ hodnotu 5 na 10</li><li>2. Klikni na tlačítko „<i>Reload</i>“</li></ol>
<b>Očekávaný výsledek:</b>	V tabulce je zvýrazněno 10 nejvyšších hodnot

Testovací scénář A. 13

## A.15 Změna dat v přehledu nákladů za oblast komodit pro konkrétní rok

<b>ID :</b>	#15
<b>Název:</b>	Změna dat v okně s přehledem nákladů za oblast komodit pro konkrétní rok
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s přehledem nákladů za oblast komodit pro konkrétní rok. Změna se týká měrné jednotky zobrazených dat.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2016, 2017 a 2018</li><li>• Otevřené okno s názvem „Brake System“ s přehledem nákladů za oblast komodit pro konkrétní rok</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na kombinované pole „Unit“</li><li>2. Kliknutím zobraz rozbalovací menu</li><li>3. Klikni na hodnotu „pcs“ v rozbalovacím menu</li><li>4. Klikni na tlačítko „Reload“</li></ol>
<b>Očekávaný výsledek:</b>	V okně jsou data týkající se počtu jednotlivých nákladových položek

Testovací scénář A. 15

## A.16 Změna dat v přehledu nákladů za oblast komodit pro konkrétní rok

<b>ID :</b>	#16
<b>Název:</b>	Změna dat v okně s přehledem nákladů za oblast komodit pro konkrétní rok
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s přehledem nákladů za oblast komodit pro konkrétní rok. Změna se týká roku, za který jsou data zobrazena.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2016, 2017 a 2018</li><li>• Otevřené okno s názvem „Brake System“ s přehledem nákladů za oblast komodit rok 2017</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na kombinované pole „Year“</li><li>2. Kliknutím zobraz rozbalovací menu</li><li>3. Klikni na hodnotu „2018“ v rozbalovacím menu</li><li>4. Klikni na tlačítko „Reload“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• V kombinovaném poli je zobrazena hodnota „2018“</li><li>• V přehledu jsou zobrazena data za rok 2018</li></ul>

Testovací scénář A. 16

## A.17 Zobrazení okna konkrétní sestavy

<b>ID :</b>	#17
<b>Název:</b>	Zobrazení okna s detaily konkrétní sestavy v dané oblasti komodit
<b>Popis:</b>	Testovací scénář pro zobrazení okna konkrétní sestavy pro danou oblast komodit.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017</li><li>• Otevřené okno s názvem „Brake System“ s přehledem nákladů za oblast komodit rok 2017</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Klikni na nadpis (tlačítko) sloupce „A1001“</li></ol>
<b>Očekávaný výsledek:</b>	Zobrazení okna „Assembly A1001“ s daty o sestavě A1001

Testovací scénář A. 17

## A.18 Změna zobrazených dat v okně konkrétní sestavy

<b>ID :</b>	#18
<b>Název:</b>	Změna zobrazených dat v okně konkrétní sestavy v dané oblasti komodit
<b>Popis:</b>	Testovací scénář pro změnu zobrazených dat v okně konkrétní sestavy v dané oblasti komodit. Změna se týká zvýraznění jiného počtu nejvyšších hodnot v přehledu.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017 a 2018</li><li>• Otevřené okno s názvem „<i>Assembly A1001</i>“ s přehledem konkrétní sestavy</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Do pole „<i>Number of highest values</i>“ napiš hodnotu 10</li><li>2. Klikni na tlačítko „<i>Reload</i>“</li></ol>
<b>Očekávaný výsledek:</b>	V okně „ <i>Assembly A1001</i> “ je zvýrazněno 10 nejvyšších hodnot

Testovací scénář A. 18

## A.19 Změna zobrazených dat v okně konkrétní sestavy

<b>ID :</b>	#19
<b>Název:</b>	Změna zobrazených dat v okně konkrétní sestavy v dané oblasti komodit
<b>Popis:</b>	Testovací scénář pro změnu zobrazených dat v okně konkrétní sestavy v dané oblasti komodit. Změna se týká výběru jiného roku, pro který se mají data zobrazit
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017 a 2018</li><li>• Otevřené okno s názvem „<i>Assembly A1001</i>“ s přehledem konkrétní sestavy</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na kombinované pole „<i>Year</i>“</li><li>2. Kliknutím otevři rozbalovací menu kombinovaného pole</li><li>3. Klikni na hodnotu „<i>2018</i>“</li><li>4. Klikni na tlačítko „<i>Reload</i>“</li></ol>
<b>Očekávaný výsledek:</b>	V okně „ <i>Assembly A1001</i> “ jsou zobrazena data týkající se sestavy A1001 za rok 2018

Testovací scénář A. 19

## A.20 Změna zobrazených dat v okně konkrétní sestavy

<b>ID :</b>	#20
<b>Název:</b>	Změna zobrazených dat v okně konkrétní sestavy v dané oblasti komodit
<b>Popis:</b>	Testovací scénář pro změnu zobrazených dat v okně konkrétní sestavy v dané oblasti komodit. Změna se týká výběru jiné měrné jednotky, pro kterou jsou data zobrazena
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017 a 2018</li><li>• Otevřené okno s názvem „<i>Assembly A1001</i>“ s přehledem konkrétní sestavy</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na kombinované pole „<i>Unit</i>“</li><li>2. Kliknutím otevři rozbalovací menu kombinovaného pole</li><li>3. Klikni na hodnotu „<i>pcs</i>“</li><li>4. Klikni na tlačítko „<i>Reload</i>“</li></ol>
<b>Očekávaný výsledek:</b>	V okně „ <i>Assembly A1001</i> “ jsou zobrazena data jako počet jednotlivých nákladových položek sestavy A1001

Testovací scénář A. 20



## A.21 Zobrazení okna s analýzou kategorií nákladů

<b>ID :</b>	#21
<b>Název:</b>	Zobrazení okna s analýzou kategorií nákladů napříč všemi studentskými týmy
<b>Popis:</b>	Testovací scénář pro zobrazení okna s analýzou kategorií nákladů napříč všemi studentskými týmy
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017</li><li>• Existence dat pro více studentských týmů</li><li>• Otevřené okno s názvem „<i>Historical Data Overview</i>“ (hlavní okno části aplikace s přehledem historických dat).</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Klikni na tlačítko „<i>CATEGORY OVERVIEW</i>“</li></ol>
<b>Očekávaný výsledek:</b>	Zobrazení okna „ <i>CATEGORY OVERVIEW</i> “ s přehledem za rok 2017

Testovací scénář A. 21

## A.22 Změna dat v okně s analýzou kategorií nákladů

<b>ID :</b>	#22
<b>Název:</b>	Změna dat v okně s analýzou kategorií nákladů napříč všemi studentskými týmy
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s analýzou kategorií nákladů napříč všemi studentskými týmy. Změna se týká volby jiné nákladové kategorie
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017</li><li>• Existence dat pro více studentských týmů</li><li>• Otevřené okno s názvem „CATEGORY OVERVIEW“ s defaultně nastavenou kategorií nákladů „Material“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V Group Boxu „Category“ klikni na Radio Button „Tooling“</li></ol>
<b>Očekávaný výsledek:</b>	V okně „CATEGORY OVERVIEW“ se zobrazí data pro kategorii nákladů „Tooling“

Testovací scénář A. 22

## A.23 Změna dat v okně s analýzou kategorií nákladů

<b>ID :</b>	#23
<b>Název:</b>	Změna dat v okně s analýzou kategorií nákladů napříč všemi studentskými týmy
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s analýzou kategorií nákladů napříč všemi studentskými týmy. Změna se týká volby jiného roku, pro který jsou data zobrazována.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017 a 2018</li><li>• Existence dat pro více studentských týmů</li><li>• Otevřené okno s názvem „CATEGORY OVERVIEW“ s defaultně nastavenou kategorií nákladů „Material“ a rokem „2018“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na kombinované pole „Year“</li><li>2. Kliknutím otevři rozbalovací menu</li><li>3. Klikni na hodnotu „2017“</li></ol>
<b>Očekávaný výsledek:</b>	V okně „CATEGORY OVERVIEW“ se zobrazí data pro kategorii nákladů „Material“ pro rok 2017

Testovací scénář A. 23

## A.24 Změna dat v okně s analýzou kategorií nákladů

<b>ID :</b>	#24
<b>Název:</b>	Změna dat v okně s analýzou kategorií nákladů napříč všemi studentskými týmy
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s analýzou kategorií nákladů napříč všemi studentskými týmy. Změna se týká volby procentuální hranice
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2018</li><li>• Existence dat pro více studentských týmů</li><li>• Otevřené okno s názvem „CATEGORY OVERVIEW“ s defaultně nastavenou kategorií nákladů „Material“ a rokem „2018“ a procentuální hranicí 50%</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Do textového pole „Lower by“ zapište hodnotu „10“</li></ol>
<b>Očekávaný výsledek:</b>	V okně „CATEGORY OVERVIEW“ se zobrazí data pro kategorii nákladů „Material“ pro rok 2018, konkrétně seznam všech týmů, jejichž suma nákladů pro kategorii „Material“ je nižší o 10% než je hodnota v poli „Median“

Testovací scénář A. 24

## A.25 Zobrazení okna s analýzou oblastí komodit

<b>ID :</b>	#25
<b>Název:</b>	Zobrazení okna s analýzou oblastí komodit napříč všemi studentskými týmy
<b>Popis:</b>	Testovací scénář pro zobrazení okna s analýzou oblastí komodit napříč všemi studentskými týmy
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017</li><li>• Existence dat pro více studentských týmů</li><li>• Otevřené okno s názvem „<i>Historical Data Overview</i>“ (hlavní okno části aplikace s přehledem historických dat).</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Klikni na tlačítko „<i>COMMODITY AREAS OVERVIEW</i>“</li></ol>
<b>Očekávaný výsledek:</b>	Zobrazení okna „ <i>COMMODITY AREAS OVERVIEW</i> “ s přehledem za rok 2017

Testovací scénář A. 25

## A.26 Změna dat v okně s analýzou oblastí komodit

<b>ID :</b>	#26
<b>Název:</b>	Změna dat v okně s analýzou oblastí komodit napříč všemi studentskými týmy
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s analýzou oblastí komodit napříč všemi studentskými týmy. Změna se týká volby jiné oblasti komodit
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017</li><li>• Existence dat pro více studentských týmů</li><li>• Otevřené okno s názvem „<i>COMMODITY AREAS OVERVIEW</i>“ s defaultně nastavenou oblastí „<i>Brake System</i>“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V Group Boxu „<i>Area of Commodity</i>“ klikni na Radio Button „<i>Engine &amp; Drivetrain</i>“</li></ol>
<b>Očekávaný výsledek:</b>	V okně „ <i>COMMODITY AREAS OVERVIEW</i> “ se zobrazí data pro oblast „ <i>Engine &amp; Drivetrain</i> “

Testovací scénář A. 26

## A.27 Změna dat v okně s analýzou oblastí komodit

<b>ID :</b>	#27
<b>Název:</b>	Změna dat v okně s analýzou oblastí komodit napříč všemi studentskými týmy
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s analýzou oblastí komodit napříč všemi studentskými týmy. Změna se týká volby jiného roku, pro který jsou data zobrazována.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2017 a 2018</li><li>• Existence dat pro více studentských týmů</li><li>• Otevřené okno s názvem „<i>COMMODITY AREAS OVERVIEW</i>“ s defaultně nastavenou oblastí „<i>Brake System</i>“ a rokem „2018“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najed' myší na kombinované pole „<i>Year</i>“</li><li>2. Kliknutím otevři rozbalovací menu</li><li>3. Klikni na hodnotu „2017“</li></ol>
<b>Očekávaný výsledek:</b>	V okně „ <i>COMMODITY AREAS OVERVIEW</i> “ se zobrazí data pro oblast „ <i>Brake System</i> “ pro rok 2017

Testovací scénář A. 27

## A.28 Změna dat v okně s analýzou oblastí komodit

<b>ID :</b>	#28
<b>Název:</b>	Změna dat v okně s analýzou oblastí komodit napříč všemi studentskými týmy
<b>Popis:</b>	Testovací scénář pro změnu dat v okně s analýzou oblastí komodit napříč všemi studentskými týmy. Změna se týká volby procentuální hranice.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence historických dat v datovém skladu za rok 2018</li><li>• Existence dat pro více studentských týmů</li><li>• Otevřené okno s názvem „<i>COMMODITY AREAS OVERVIEW</i>“ s defaultně nastavenou oblastí „<i>Brake System</i>“ a rokem „2018“ a procentuální hranicí 50%</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Do textového pole „<i>Lower by</i>“ zapište hodnotu „10“</li></ol>
<b>Očekávaný výsledek:</b>	V okně „ <i>COMMODITY AREAS OVERVIEW</i> “ se zobrazí data pro oblast „ <i>Brake System</i> “ pro rok 2018, konkrétně seznam všech týmů, jejichž suma nákladů pro oblast „ <i>Brake System</i> “ je nižší o 10% než je hodnota v poli „ <i>Median</i> “

Testovací scénář A. 28



## A.29 Otevření aplikační části s aktuálními daty

<b>ID :</b>	#29
<b>Název:</b>	Otevření aplikační části s aktuálními daty
<b>Popis:</b>	Testovací scénář pro zobrazení hlavního okna aplikační části pro prohlížení a editaci dat pro aktuální rok.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence uživatelských dat v operační databázi</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Spustí aplikaci</li><li>2. Klikni na tlačítko „<i>Show actual data</i>“</li><li>3. Kliknutím zvol ve zobrazeném seznamu jeden z uživatelských profilů</li><li>4. Klikni na tlačítko „<i>Continue</i>“</li></ol>
<b>Očekávaný výsledek:</b>	Zobrazení okna s názvem „ <i>UWB Racing Team Database</i> “

Testovací scénář A. 29

## A.30 Zobrazení sestav konkrétní oblasti

<b>ID :</b>	#30
<b>Název:</b>	Otevření seznamu sestav konkrétní oblasti
<b>Popis:</b>	Testovací scénář pro zobrazení seznamu sestav konkrétní oblasti.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „<i>UWB Racing Team Database</i>“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V seznamu oblastí komodit „<i>Areas</i>“ najed' myší na jednu z oblastí</li><li>2. Klikni na název zvolené oblasti</li></ol>
<b>Očekávaný výsledek:</b>	V okně s názvem „ <i>UWB Racing Team Database</i> “ se zobrazí seznam s výčtem jednotlivých sestav zvolené oblasti

Testovací scénář A. 30

## A.31 Otevření okna s detaily sestavy

<b>ID :</b>	#31
<b>Název:</b>	Otevření okna s detaily sestavy
<b>Popis:</b>	Testovací scénář pro zobrazení okna s detaily sestavy.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“</li><li>• Otevřený seznam sestav konkrétní oblasti „Brake System“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V seznamu sestav najed' myší na první ze sestav</li><li>2. Klikni na název zvolené sestavy</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Zobrazí se okno s názvem „A1001 0000“</li><li>• V okně v oblasti „Information“ je pro pole „System“ zobrazena hodnota „Brake System“</li></ul>

Testovací scénář A. 31

## A.32 Rezervace sestavy

<b>ID :</b>	#32
<b>Název:</b>	Rezervace sestavy
<b>Popis:</b>	Testovací scénář pro rezervaci sestavy.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“ se seznamem sestav pro oblast „Brake System“</li><li>• Otevřené okno sestavy „A1001 0000“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V pravém horním rohu okna „A1001 0000“ klikni na tlačítko „Reservate this assembly“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• V okně s názvem „A1001 0000“ v pravém horním rohu se zobrazí tlačítko „Cancel reservation“ a Toggle Button se stavy „View“ a „Edit“</li><li>• V okně „UWB Racing Team Database“ je v seznamu sestav daná sestava „A1001 0000“ označena v řádku zelenou indikací</li></ul>

Testovací scénář A. 32

## A.33 Přepnutí sestavy do editovacího módu

<b>ID :</b>	#33
<b>Název:</b>	Přepnutí módu sestavy do módu editačního
<b>Popis:</b>	Testovací scénář přepnutí módu sestavy do módu editačního.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“ se seznamem sestav pro oblast „Brake System“</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Rezervovaná sestava „A1001 0000“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V pravém horním rohu okna „A1001 0000“ klikni na tlačítko název stavu Toggle Buttonu „Edit“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• V okně s názvem „A1001 0000“ v pravém horním rohu se zvýrazní nápis „Edit“</li><li>• V okně se nově zobrazí textová pole, která lze editovat</li></ul>

Testovací scénář A. 33

## A.34 Přepnutí sestavy do módu prohlížení

<b>ID :</b>	#34
<b>Název:</b>	Přepnutí módu sestavy do módu pro prohlížení
<b>Popis:</b>	Testovací scénář pro přepnutí módu sestavy do módu prohlížení.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“ se seznamem sestav pro oblast „Brake System“</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Rezervovaná sestava „A1001 0000“</li><li>• Sestava v editačním módu „Edit“ „A1001 0000“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V pravém horním rohu okna „A1001 0000“ klikni na tlačítko název stavu Toggle Buttonu „View“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• V okně s názvem „A1001 0000“ v pravém horním rohu se zvýrazní nápis „View“</li><li>• V okně zmizí textová pole pro editaci</li></ul>

Testovací scénář A. 34

## A.35 Zobrazení nákladových položek kategorie sestavy

<b>ID :</b>	#35
<b>Název:</b>	Zobrazení jednotlivých nákladových položek sestavy dle zvolné nákladové kategorie
<b>Popis:</b>	Testovací scénář pro zobrazení jednotlivých nákladových položek sestavy dle zvolné nákladové kategorie.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“ se seznamem sestav pro oblast „Brake System“</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Defaultně zobrazen seznam nákladových položek „Material“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V pravém horním rohu okna „A1001 0000“ v oblasti „Display“ klikni na název „Fasteners“</li></ol>
<b>Očekávaný výsledek:</b>	V okně s názvem „A1001 0000“ v dolní polovině se zobrazí tabulka s jednotlivými nákladovými položkami sestavy patřící do kategorie „Fasteners“

Testovací scénář A. 35

## A.36 Úprava informací sestavy

<b>ID :</b>	#36
<b>Název:</b>	Úprava jednotlivých informací sestavy
<b>Popis:</b>	Testovací scénář pro upravení informací o dané sestavě.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Sestava „A1001 0000“ v editačním módu</li><li>• Sestava obsahuje nákladové položky</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Do textového pole „Name“ napište hodnotu „Assembly Name“</li><li>2. Do textového pole „Description“ napište hodnotu „Assembly Description“</li><li>3. Do textového pole „Quantity“ napište hodnotu „2“</li><li>4. Klikněte na mód „View“ v horním pravém rohu okna</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Textová pole zmizela a namísto nich jsou zobrazeny popisky s hodnotami, které v krocích byly zadány</li><li>• Pro atribut „Name“ je zobrazena hodnota „Assembly Name“</li><li>• Pro atribut „Description“ je zobrazena hodnota „Assembly Description“</li><li>• Pro atribut „Quantity“ je zobrazena hodnota „2“</li><li>• Celková cena sestavy se zdvojnásobila</li></ul>

Testovací scénář A. 36



## A.37 Změna pořadí sestavy

<b>ID:</b>	#37
<b>Název:</b>	Úprava pořadí sestavy
<b>Popis:</b>	Testovací scénář pro upravení pořadí dané sestavy.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Sestava „A1001 0000“ v editačním módu</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V levém horním rohu okna u číselného označení sestavy (atribut „Number“) „A1001 0000“ klikněte na tlačítko po pravé straně (ikona tužky)</li><li>2. V nově zobrazeném okně „A1001 0000 number“ najedzte myší na tlačítko ve tvaru šipky dolů, kterým se pořadí sestavy navyšuje o jedna</li><li>3. Klikněte na toto tlačítko, dokud se červený indikátor (křížek) vedle označení sestavy v tomto okně nezmění na zelený indikátor (háček)</li><li>4. Klikněte na tlačítko „Save new number“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Název okna sestavy se změnil na „A100x 0000“, kde x je číslo nově zvoleného pořadí</li><li>• Pro atribut „Number“ se změnila hodnota na „A100x 0000“, kde x je číslo nově zvoleného pořadí</li></ul>

Testovací scénář A. 37

## A.38 Změna pořadí sestavy

<b>ID:</b>	#38
<b>Název:</b>	Úprava pořadí sestavy
<b>Popis:</b>	Testovací scénář pro upravení pořadí dané sestavy.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Sestava „A1001 0000“ v editačním módu</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V levém horním rohu okna u číselného označení sestavy (atribut „Number“) „A1001 0000“ klikněte na tlačítko po pravé straně (ikona tužky)</li><li>2. V nově zobrazeném okně „A1001 0000 number“ najedte myší na tlačítko ve tvaru šipky dolů, kterým se pořadí sestavy navýšuje o jedna</li><li>3. Klikněte na toto tlačítko, dokud se červený indikátor (křížek) vedle označení sestavy v tomto okně nezmění na zelený indikátor (háček)</li><li>4. Klikněte na tlačítko „Save new number“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Název okna sestavy se změnil na „A100x 0000“, kde x je číslo nově zvoleného pořadí</li><li>• Pro atribut „Number“ se změnila hodnota na „A100x 0000“, kde x je číslo nově zvoleného pořadí</li></ul>

Testovací scénář A. 38

## A.39 Vytvoření sestavy

<b>ID:</b>	#39
<b>Název:</b>	Vytvoření sestavy
<b>Popis:</b>	Testovací scénář pro vytvoření nové sestavy.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“ se seznamem sestav pro oblast „Brake System“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Klikněte v pravém rohu okna „UWB Racing Team Database“ na tlačítko „Add new assembly“</li></ol>
<b>Očekávaný výsledek:</b>	V seznamu sestav okna „UWB Racing Team Database“ se zobrazila nově přidána sestava „A100x 0000“, kde x je číslem pořadí nově přidané sestavy

Testovací scénář A. 39

## A.40 Vytvoření nové části sestavy

<b>ID:</b>	#40
<b>Název:</b>	Vytvoření nové části sestavy
<b>Popis:</b>	Testovací scénář pro vytvoření nové části sestavy.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“ se seznamem sestav pro oblast „Brake System“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najedte myší na řádek sestavy „A1001 0000“ v seznamu sestav okna „UWB Racing Team Database“</li><li>2. Klikněte na tlačítko se zeleným křížkem na konci řádku</li></ol>
<b>Očekávaný výsledek:</b>	V seznamu sestav okna „UWB Racing Team Database“ se pro sestavu „A1001 0000“, rozbalil seznam jejich částí, kde se nachází nově přidaná část s označením „000x“, kde x je číslem pořadí nově přidané části sestavy

Testovací scénář A. 40

## A.41 Otevření části sestavy

<b>ID:</b>	#41
<b>Název:</b>	Otevření části sestavy
<b>Popis:</b>	Testovací scénář pro otevření části sestavy.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“ se seznamem sestav pro oblast „Brake System“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Najedte myší na řádek sestavy „A1001 0000“ v seznamu sestav okna „UWB Racing Team Database“</li><li>2. Klikněte na tlačítko se zeleným křížkem na konci řádku</li></ol>
<b>Očekávaný výsledek:</b>	V seznamu sestav okna „UWB Racing Team Database“ se pro sestavu „A1001 0000“, rozbil seznam jejich částí, kde se nachází nově přidaná část s označením „000x“, kde x je číslem pořadí nově přidané části sestavy

Testovací scénář A. 41

## A.42 Přidání materiálu sestavě

<b>ID:</b>	#42
<b>Název:</b>	Přidání materiál sestavě
<b>Popis:</b>	Testovací scénář pro vytvoření nákladové položky kategorie „Material“ pro konkrétní sestavu. (Tento postup lze opakovat pro část sestavy)
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Sestava „A1001 0000“ v editačním módu</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V okně zvolte v oblasti „Display“ kategorii „Material“</li><li>2. V levém dolním rohu klikněte na tlačítko „Add material to item“</li><li>3. V nově otevřeném okně s názvem „Add material to A1001 0000“ najedte myší na oblast „Category“ a zvolte zde hodnotu „Controls“</li><li>4. Ve vedlejší tabulce zvolte položku „Cable, Pull“</li><li>5. V pravé části okna zapište do textového pole „Use“ hodnotu „material use“</li><li>6. V pravé části okna zapište do textového pole „Quantity“ hodnotu „1“</li><li>7. V pravé části okna zapište do textového pole „Size1“ hodnotu „100“</li><li>8. Klikněte na tlačítko „Add material to item“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• V seznamu materiálu v okně sestavy „A1001 0000“ přibyl řádek s nově vytvořeným materiálem</li><li>• V řádku se nachází ve sloupci „Materials“ hodnota „Cable, Pull“</li><li>• V řádku se nachází ve sloupci „Use“ hodnota „material use“</li><li>• V řádku se nachází ve sloupci „Unit Cost“ hodnota „\$ 0.35“</li><li>• V řádku se nachází ve sloupci „Size1“ hodnota „100“</li><li>• V řádku se nachází ve sloupci „Unit1“ hodnota „mm“</li><li>• V řádku se nachází ve sloupci „Quantity“ hodnota „1“</li><li>• V řádku se nachází ve sloupci „Sub Total“ hodnota „\$ 0.35“</li></ul>

Testovací scénář A. 42

## A.43 Přidání procesu sestavě

<b>ID:</b>	#43
<b>Název:</b>	Přidání procesu sestavě
<b>Popis:</b>	Testovací scénář pro vytvoření nákladové položky kategorie „Process“ pro konkrétní sestavu. (Tento postup lze opakovat pro část sestavy)
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Sestava „A1001 0000“ v editačním módu</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V okně zvolte v oblasti „Display“ kategorii „Process“</li><li>2. V levém dolním rohu klikněte na tlačítko „Add process to item“</li><li>3. V nově otevřeném okně s názvem „Add process to A1001 0000“ najedte myší na oblast „Category“ a zvolte zde hodnotu „Coating“</li><li>4. Ve vedlejší tabulce zvolte položku „Exhaust System Ceramic Coating“</li><li>5. V pravé části okna zapište do textového pole „Use“ hodnotu „process use“</li><li>6. V pravé části okna zapište do textového pole „Quantity“ hodnotu „3“</li><li>7. Klikněte na tlačítko „Add process to item“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• V seznamu procesů v okně sestavy „A1001 0000“ přibyl řádek s nově vytvořeným procesem</li><li>• V řádku se nachází ve sloupci „Processes“ hodnota „Exhaust System Ceramic Coating“</li><li>• V řádku se nachází ve sloupci „Use“ hodnota „process use“</li><li>• V řádku se nachází ve sloupci „Unit Cost“ hodnota „\$ 0.25“</li><li>• V řádku se nachází ve sloupci „Unit“ hodnota „m“</li><li>• V řádku se nachází ve sloupci „Quantity“ hodnota „3“</li><li>• V řádku se nachází ve sloupci „Sub Total“ hodnota „\$ 0.75“</li></ul>

Testovací scénář A. 43

## A.44 Přidání spojovacího materiálu sestavě

<b>ID:</b>	#44
<b>Název:</b>	Přidání spojovacího materiálu sestavě
<b>Popis:</b>	Testovací scénář pro vytvoření nákladové položky kategorie „Fasteners“ pro konkrétní sestavu. (Tento postup lze opakovat pro část sestavy)
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Sestava „A1001 0000“ v editačním módu</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V okně zvolte v oblasti „Display“ kategorii „Fasteners“</li><li>2. V levém dolním rohu klikněte na tlačítko „Add fastener to item“</li><li>3. V tabulce zvolte položku „Tie Wrap“</li><li>4. V pravé části okna запиšte do textového pole „Use“ hodnotu „fastener use“</li><li>5. V pravé části okna запиšte do textového pole „Quantity“ hodnotu „2“</li><li>6. Klikněte na tlačítko „Add fastener to item“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• V seznamu spojovacího materiálu v okně sestavy „A1001 0000“ přibyl řádek s nově vytvořeným spojovacím materiálem</li><li>• V řádku se nachází ve sloupci „Fasteners“ hodnota „Tie Wrap“</li><li>• V řádku se nachází ve sloupci „Use“ hodnota „fastener use“</li><li>• V řádku se nachází ve sloupci „Unit Cost“ hodnota „\$ 0.04“</li><li>• V řádku se nachází ve sloupci „Unit1“ hodnota „unit“</li><li>• V řádku se nachází ve sloupci „Quantity“ hodnota „2“</li><li>• V řádku se nachází ve sloupci „Sub Total“ hodnota „\$ 0.08“</li></ul>

Testovací scénář A. 44



## A.45 Přidání opracování sestavě

<b>ID:</b>	#45
<b>Název:</b>	Přidání opracování sestavě
<b>Popis:</b>	Testovací scénář pro vytvoření nákladové položky kategorie „Toolings“ pro konkrétní sestavu. (Tento postup lze opakovat pro část sestavy)
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Sestava „A1001 0000“ v editačním módu</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V okně zvolte v oblasti „Display“ kategorii „Toolings“</li><li>2. V levém dolním rohu klikněte na tlačítko „Add tooling to item“</li><li>3. V tabulce zvolte položku „Braze“</li><li>4. V pravé části okna запиšte do textového pole „Use“ hodnotu „tooling use“</li><li>5. V pravé části okna запиšte do textového pole „Quantity“ hodnotu „2“</li><li>6. V pravé části okna запиšte do textového pole „PVF“ hodnotu „3000“</li><li>7. V pravé části okna запиšte do textového pole „FracInId“ hodnotu „1“</li><li>8. Klikněte na tlačítko „Add tooling to item“</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• V seznamu opracování v okně sestavy „A1001 0000“ přibyl řádek s nově vytvořeným opracováním</li><li>• V řádku se nachází ve sloupci „Toolings“ hodnota „Braze“</li><li>• V řádku se nachází ve sloupci „Use“ hodnota „tooling use“</li><li>• V řádku se nachází ve sloupci „Unit Cost“ hodnota „\$ 500“</li><li>• V řádku se nachází ve sloupci „Unit“ hodnota „point“</li><li>• V řádku se nachází ve sloupci „PVF“ hodnota „3000“</li><li>• V řádku se nachází ve sloupci „FracInId“ hodnota „1“</li><li>• V řádku se nachází ve sloupci „Quantity“ hodnota „2“</li><li>• V řádku se nachází ve sloupci „Sub Total“ hodnota „\$ 0.33“</li></ul>

Testovací scénář A. 45

## A.46 Editace nákladové položky v rámci sestavy

<b>ID:</b>	#46
<b>Název:</b>	Editace nákladové položky konkrétní kategorie
<b>Popis:</b>	Testovací scénář pro editaci nákladové položky kategorie „Fasteners“ pro konkrétní sestavu. (Tento postup lze opakovat pro všechny kategorie)
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Sestava „A1001 0000“ v editačním módu</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V okně zvolte v oblasti „Display“ kategorii „Fasteners“</li><li>2. V tabulce na řádku, který chceme editovat, klikneme na tlačítko s modrou tužkou</li><li>3. V nově otevřeném okně postupujeme stejně jako v testovacím scénáři A.44 (pro kategorii materiálů A.42, procesů A.43 a opracování A.45)</li></ol>
<b>Očekávaný výsledek:</b>	Daný řádek se spojovacím materiálem obsahuje nově zadané hodnoty

Testovací scénář A. 46

## A.47 Editace pořadí nákladové položky v rámci sestavy

<b>ID:</b>	#47
<b>Název:</b>	Editace nákladové položky konkrétní kategorie
<b>Popis:</b>	Testovací scénář pro editaci pořadí nákladové položky kategorie „Fasteners“ pro konkrétní sestavu. (Tento postup lze opakovat pro všechny kategorie)
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Sestava „A1001 0000“ v editačním módu</li><li>• Sestava obsahuje alespoň dva spojovací materiály</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V okně zvolte v oblasti „Display“ kategorii „Fasteners“</li><li>2. V tabulce klikneme na řádek, u kterého chceme měnit pořadí</li><li>3. V dolní části okna klikneme na modrou šipku dolů</li></ol>
<b>Očekávaný výsledek:</b>	Daný řádek se spojovacím materiálem má vyšší pořadí a tudíž je o řádek níže v tabulce

Testovací scénář A. 47

## A.48 Odstranění nákladové položky v rámci sestavy

<b>ID:</b>	#48
<b>Název:</b>	Odstranění nákladové položky konkrétní kategorie
<b>Popis:</b>	Testovací scénář pro odstranění nákladové položky kategorie „Fasteners“ pro konkrétní sestavu. (Tento postup lze opakovat pro všechny kategorie)
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno sestavy „A1001 0000“</li><li>• Sestava „A1001 0000“ v editačním módu</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V okně zvolte v oblasti „Display“ kategorii „Fasteners“</li><li>2. V tabulce spojového materiálu na řádku, který chceme odstranit, klikněte na červený křížek</li></ol>
<b>Očekávaný výsledek:</b>	V tabulce okna „A1001 0000“ daný řádek již není zobrazen.

Testovací scénář A. 48

## A.49 Odstranění sestavy

<b>ID:</b>	#49
<b>Název:</b>	Odstranění nákladové položky konkrétní kategorie
<b>Popis:</b>	Testovací scénář pro odstranění sestavy konkrétní oblasti. (Tento postup lze opakovat pro i pro část sestavy)
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno „UWB Racing Team Database“ se seznamem sestav oblasti „Brake System“</li><li>• Existence alespoň jedné sestavy v oblasti „Brake System“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V přehledu sestav na řádku sestavy „A1001 0000“ klikneme na křížek na konci řádku</li><li>2. V nově zobrazeném okně, které se ptá, zda chceme opravu sestavu smazat, klikněte na tlačítko „Yes“</li></ol>
<b>Očekávaný výsledek:</b>	Řádek sestavy „A1001 0000“ ze seznamu je odstraněn.

Testovací scénář A. 49

## A.50 Import souborů s typy nákladu

<b>ID:</b>	#50
<b>Název:</b>	Import souborů s typy nákladu
<b>Popis:</b>	Testovací scénář pro import souborů s typy nákladu materiálu. (Tento postup lze použít všechny kategorie nákladů: materiál, procesy, spojovací materiál, opracování)
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Otevřené okno s názvem „UWB Racing Team Database“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V horní liště okna klikněte v menu na položku „Import“</li><li>2. V nově rozbaleném menu zvolte možnost „Import Material“ (případně „Import Process, Import Fastener, Import Process Multiplier“)</li><li>3. V dialogovém okně vyberte kýžený soubor s nákladovými typy</li></ol>
<b>Očekávaný výsledek:</b>	Zobrazí se uživateli zpráva o úspěšném importování souboru.

Testovací scénář A. 50

## A.51 Export obrázků sestav

<b>ID:</b>	#51
<b>Název:</b>	Spuštění exportu obrázků sestav
<b>Popis:</b>	Testovací scénář pro spuštění exportu obrázků sestav.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V horní liště okna klikněte v menu na položku „Export“</li><li>2. V nově rozbaleném menu zvolte možnost „Export Images“</li><li>3. V nově zobrazeném okně zvolte umístění v poli „Path to directory for export“ adresáře, kam chcete obrázky exportovat</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Zobrazí se okno uživateli se zprávou „Images were successfully exported!“.</li><li>• V zadaném adresáři jsou soubory s obrázky jednotlivých sestav s názvy dle číselného označení každé ze sestav</li></ul>

Testovací scénář A. 51

## A.52 Spuštění ETL procesu

<b>ID:</b>	#52
<b>Název:</b>	Spuštění ETL procesu
<b>Popis:</b>	Testovací scénář pro spuštění ETL procesu, který zajišťuje extrakci dat z operační databáze do datového skladu.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V horní liště okna klikněte v menu na položku „Export“</li><li>2. V nově rozbaleném menu zvolte možnost „Export to DW“</li></ol>
<b>Očekávaný výsledek:</b>	Po provedení ETL procesu se zobrazí okno uživateli se zprávou „Data was successfully extracted to Data Warehouse!“.

Testovací scénář A. 52



## A.53 Export zprávy o nákladech

<b>ID:</b>	#53
<b>Název:</b>	Spuštění exportu zprávy o nákladech v aktuálním roce
<b>Popis:</b>	Testovací scénář pro spuštění exportu zprávy o nákladech v aktuálním roce.
<b>Podmínky:</b>	<ul style="list-style-type: none"><li>• Existence testovacích dat ve všech tabulkách operační databáze</li><li>• Otevřené okno s názvem „UWB Racing Team Database“</li></ul>
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. V horní liště okna klikněte v menu na položku „Export“</li><li>2. V nově rozbaleném menu zvolte možnost „Export Cost Report“</li><li>3. Do textového pole „Car number“ vepište hodnotu „60“</li><li>4. Do textového pole „Competition code“ vepište hodnotu „FSAE“</li><li>5. Do textového pole „Year“ vepište hodnotu „2019“</li><li>6. Do textového pole „Path to directory“ vepište umístění adresáře, kam chcete exportovat soubory se zprávou o nákladech</li></ol>
<b>Očekávaný výsledek:</b>	<ul style="list-style-type: none"><li>• Zobrazí se okno uživateli se zprávou „Cost report was successfully exported!“.</li><li>• V zadaném adresáři jsou soubory se zprávou o nákladech</li></ul>

Testovací scénář A. 53