



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

VYUŽITÍ VBA JAKO NÁSTROJE PRO TVORBU PODPŮRNÝCH FIREMNÍCH APLIKACÍ

USE VBA AS A TOOL FOR THE CREATION OF SUPPORTING BUSINESS APPLICATIONS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Aleš Kadlec

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Dydowicz, Ph.D.

BRNO 2023

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Aleš Kadlec**
Vedoucí práce: **Ing. Petr Dydowicz, Ph.D.**
Akademický rok: 2022/23
Studijní program: Manažerská informatika

Garant studijního programu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

Využití VBA jako nástroje pro tvorbu podpůrných firemních aplikací

Charakteristika problematiky úkolu:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrh řešení, přínos práce
Závěr
Seznam použité literatury

Cíle, kterých má být dosaženo:

Cílem bakalářské práce je návrh podpůrné aplikace pomocí VBA pro zefektivnění řízení skladu náhradních dílů. Aplikace bude urychlovat práci skladníkům ve skladu náhradních dílů. Výstupem aplikace budou funkce vytvoření názvu náhradního dílu podle šablony dle kategorie konkrétního dílu, export náhradních dílů, u kterých chybí fotky či vytvořené názvy.

Základní literární prameny:

BRADEN, M. a M. SCHWIMMER. Excel 2007 VBA. Velká kniha řešení. Brno: Computer Press, a.s., 2009. 685 s. ISBN 978-80-251-2698-1.

ČIHAŘ, J. 1001 tipů a triků pro Microsoft Excel 2007/2010. Brno: Computer Press, a.s., 2011. 488 s. ISBN 978-80-251-2587-8.

KRÁL, M. Excel VBA. Výukový kurz. Brno: Computer Press, a.s., 2010. 504 s. ISBN 978-80-251-2358-4.

KRÁL, M. Excel 2010 – snadno a rychle. Praha: Grada Publishing a.s., 2010. 143 s. ISBN 80-2473-495-8.

LAURENČÍK, M. Programování v Excelu 2007 a 2010. Praha: Grada Publishing a.s., 2011. 192 s. ISBN 978-80-247-3448-4.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2022/23

V Brně dne 5.2.2023

L. S.

Ing. Jiří Kříž, Ph.D.
garant

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

Abstrakt

Tato bakalářská práce se zaměřuje na návrh a implementaci aplikace určené do skladu náhradních dílů. Hlavním cílem aplikace je tvorba názvů náhradních dílů. Tato aplikace dále umožňuje exportovat stavy skladu, zjistit seznam vytvořených fotodokumentací a nahrát data ze zdrojového souboru aplikace vyhledavače náhradních dílů. Aplikace je implementována v prostředí Microsoft Excel v jazyku VBA. Aplikace bude využívána ve skladu náhradních dílů, kde si klade za cíl zefektivnit práci a sjednotit názvy jednotlivých dílů konkrétní kategorie.

Klíčová slova

Aplikace, VBA, Microsoft Excel, náhradní díly

Abstract

This bachelor thesis focuses on the design and implementation of an application for a spare parts warehouse. The main goal of the application is to create names of spare parts. This application also allows to export the stock statuses, to get the list of created photo documentation and to upload data from the source file of the spare parts finder application. The application is implemented in Microsoft Excel environment in VBA language. The application will be used in the spare parts warehouse, where it aims to streamline work and unify the names of individual parts of a particular category.

Key words

Applications, VBA, Microsoft Excel, spare parts

Bibliografické citace

KADLEC, Aleš. *Využití VBA jako nástroje pro tvorbu podpůrných firemních aplikací*. Brno, 2023. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/152095>.
Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky.
Vedoucí práce Petr Dydowicz.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 15. května 2023

.....

Poděkování

Chtěl bych poděkovat svému vedoucímu panu Ing. Petrovi Dydowiczovi, Ph.D. za jeho odborné vedení bakalářské práce, rady a konzultace k bakalářské práci. Dále bych chtěl poděkovat firmě HARTMANN-RICO a.s. a panu Ing. Michalu Bernátovi za dlouholetou spolupráci v této firmě.

OBSAH

ÚVOD	10
VYMEZENÍ PROBLÉMU A CÍLE PRÁCE	11
1. TEORETICKÁ VÝCHODISKA PRÁCE.....	12
1.1 MICROSOFT EXCEL A VISUAL BASIC FOR APPLICATION	12
1.2 PROGRAMOVÁNÍ VE VBA.....	13
1.2.1 Zabezpečení makra a aplikací VBA.....	15
1.2.2 Uživatelské prostředí.....	15
1.2.3 Proměnné	17
1.2.4 Konstrukce jazyka VBA	19
1.2.5 Ošetření chyb.....	23
1.2.6 Funkce a procedury	24
1.2.7 Uživatelský formulář	24
1.2.8 Ovládací prvky	25
1.2.9 Debugging	26
1.3 FUNKČNÍ MODELOVÁNÍ.....	28
1.3.1 Diagram toku dat	28
1.3.2 Vývojový diagram.....	28
1.4 ANALYTICKÉ METODY	29
1.4.1 HOS 8 Analýza	30
1.4.2 SWOT analýza	31
2. ANALÝZA SOUČASNÉHO STAV	33
2.1 O SPOLEČNOSTI.....	33
2.1.1 Předmět podnikání.....	33
2.1.2 Struktura společnosti.....	34
2.1.3 Zaměření práce.....	34
2.2 METODA HOS 8	34
2.3 SWOT ANALÝZA	38
2.4 ZHODNOCENÍ ANALÝZ.....	40
2.1 EXISTUJÍCÍ ALTERNATIVY.....	41
3. VLASTNÍ NÁVRHY ŘEŠENÍ	42
3.1 DATOVÉ MODELOVÁNÍ.....	42
3.1.1 Struktura dat pro názvosloví a kategorizaci.....	42
3.1.2 Struktura dat skladových zásob.....	43
3.1.3 Seznam kategorií.....	43
3.1.4 Fotodokumentace	44
3.2 FUNKČNÍ MODELOVÁNÍ.....	44
3.3 POPIS APLIKACE.....	47
3.3.1 Hlavní list.....	47
3.3.2 Aktualizování listu s kategoriemi.....	49
3.3.3 Vytvoření nové šablony kategorie	49
3.3.4 Importování kategorie z LOTUS EXPORT	52
3.3.5 Navolení kategorie	53
3.3.6 Tvorba názvu pro vyhledavač.....	55
3.3.7 Report dat	57
3.3.8 Vymazat nepoužívané listy.....	59
3.4 ZHODNOCENÍ PRÁCE	59

3.4.1	<i>Předchozí řešení a zavádění nové aplikace</i>	59
3.5	EKONOMICKÉ ZHODNOCENÍ	59
3.6	PŘÍNOSY PRÁCE	60
3.7	VÝVOJ APLIKACE DO BUDOUCNA	61
ZÁVĚR		63
SEZNAM POUŽITÝCH ZDROJŮ		64
SEZNAM POUŽITÝCH OBRÁZKŮ		65
SEZNAM POUŽITÝCH TABULEK		66

ÚVOD

V dnešní době už si jen těžko představíme nějakou firmu nebo instituci bez toho, aniž by k řízení a plnění pracovních procesů nevyužívali informační technologie. Moderní firmy a instituce využívají informační technologie ke generování zisku, ulehčení práce, dosažení důležitých výpočtových výsledků sloužících pro analýzy manažerského rozhodování, ale také jsou tyto technologie podnikatelskými příležitostmi. V oboru informačních technologií se můžeme setkat s podnikatelskými příležitostmi zejména hardwarovými nebo softwarovými. Příležitosti hardwarové mohou profitovat na prodeji, půjčování a opravách hardwaru, kdežto softwarové nabízejí určitý program sloužící ke generování zisku a růstu společnosti.

Tato bakalářská práce se bude zabývat návrhem a vývojem aplikace sloužící k zautomatizování pracovních procesů ve skladu náhradních dílů ve výrobní společnosti HARTMANN-RICO a.s.. Aplikace bude vyvíjena v prostředí MS Excel za pomoci programovacího jazyka Visual Basic for Application.

Vymezení problému a cíle práce

Hlavním cíl bakalářské práce je návrh a implementace aplikace, která bude řešit efektivní práci s databází náhradních dílů na výrobní stroje. K vývoji aplikace bude využito prostředí MS Excel a programovací jazyk Visual Basic for Application. Aplikace bude zajišťovat konzistentnost, správnost a úplnost informací o náhradních dílech, které se za pomoci nástroje Power BI data filtrují a zobrazují ve webovém rozhraní.

Aplikace bude umožňovat vytvářet skladníkům názvy náhradní díly, protože díky velkému množství kategorií dílů je náročné, až nemyslitelné se naučit pravidla, jak vytvářet názvy pro jednotlivé kategorie. Každý díl má svoji kategorii skládající se ze sedmi úrovní. Pro každou kategorii je zapotřebí vytvořit pomocí formuláře šablonu názvu, podle které bude aplikace napovídat při tvorbě názvů. Pravidla pro tvorbu šablony názvu jsou určena ve směrnici pro tvorbu názvosloví. Další funkcí bude editování a smazání šablony zvolené kategorie. Pro výběr konkrétní kategorie bude sloužit rozklikávací formulář. Způsobem výběru konkrétní kategorie bude importováním kategorie na základě zadání objednavatele v objednávkovém systému. Druhou možností bude navolení dané kategorie ručně. S výběrem kategorií je spjata aktualizování seznamu kategorií, které se získává z externího souboru. Užitečným makrem aplikace bude makro vytvářející report dílů s informacemi o existenci fotodokumentace dílů. Pro udržování čistoty v sešitu bude použito makro určené na vymazání pomocných listů. V sešitu zůstanou pouze listy s daty potřebnými pro správný běh aplikace.

Práce se skládá ze třech kapitol, teoretická východiska práce, analýza současného stavu a vlastní řešení. V teoretické části bude popsán balíček Microsoft Office, konkrétněji Microsoft Excel a jazyk Visual Basic for Application, včetně jeho datových typů proměnných a základních konstrukcí. Druhá kapitola bude věnována aktuálnímu stavu společnosti, představení předmětu podnikání, její organizační struktura a informačních systémech využívaných ve firmě. Nedílnou součástí této části bude analýza silných a slabých stránek, SWOT, dále také analýza HOS 8. Poslední kapitole bude řešit samotné řešení aplikace pro zlepšení a zjednodušení chodu skladu náhradních dílů.

1. Teoretická východiska práce

V této kapitole práce popisuje programovací jazyk Visual Basic for Application (dále jen VBA), kde jsou popsány informace od toho, co VBA představuje a k čemu je určen. Následně jsou popsány datové typy, konstrukce, funkce a procedury, uživatelské formuláře a debugování chyb v programovacím jazyku VBA. V poslední podkapitole je charakterizováno datové modelování a následně budou vysvětleny metoda HOS 8 a SWOT analýza.

1.1 Microsoft Excel a Visual Basic for Application

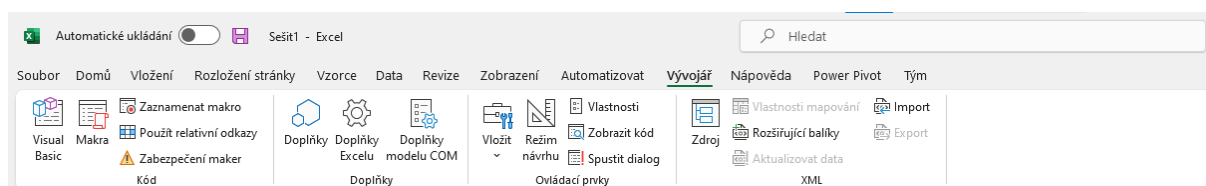
Microsoft Excel je tabulkový procesor, který umožňuje uživatelům načítat, zpracovávat, upravovat či filtrovat data. Importovaná data lze pomocí vzorců a funkcí upravovat a analyzovat. Funkce lze využít předdefinované, které jsou součástí programu Excel nebo si lze vytvořit svoje vlastní funkce. Do Excelu může uživatel importovat data, která následně může podle požadavků upravovat. Z již importovaných dat lze dopočítávat data odvozená. Nejjednodušším příkladem je výpočet roku narození z rodného čísla. Upravená a zanalyzovaná data lze pomocí grafů, tabulek, kontingenčních tabulek graficky zobrazit (9).

Program Microsoft Excel je součástí balíčků aplikací Office 365 od firmy Microsoft. Pro pokročilejší uživatele nabízí Excel tvorbu maker a programování aplikací v programovacím jazyku Visual Basic for Application, sloužícího pro zautomatizování a zefektivnění práce. Visual Basic for Application vychází z interpretu Basic, implementovaného na všech počítačích dodávaných firmou Microsoft. Následníkem interpretu Basic se později stal Visual Basic. Z tohoto jazyku poté vznikl jazyk Visual Basic for Application pro kancelářský balíček Office, dále jen jako zkratka VBA. Mezi výhody VBA se řadí stabilita a neměnné jádro programovacího jazyka. Neměnné jádro jednoduše umožňuje jednotlivé procedury a funkce přenášet v rámci jednotlivých projektů z jedné aplikace Office 365 na jinou aplikaci Office 365. Poměrně jednoduchá je i automatizace operací, kdy je makro vytvořeno zaznamenáním nebo naprogramováním kódu. Tento kód, pro zefektivnění opakujících se aktivit, se po spuštění

uživatel vykoná automaticky, namísto toho, aby uživatel operace prováděl manuálně. Uživatel pouze spustí dané makro. Kód instrukce vykoná automaticky (16).

1.2 Programování ve VBA

Programovací jazyk VBA je výkonný nástroj, sloužící zejména k zjednodušení a zautomatizování práce s aplikacemi Microsoft Office. Jazyk VBA je podporován v aplikacích Access, Excel, Outlook a Powerpoint. Pomocí jazyka VBA lze vyvíjet makra a programy, jenž ovládají tlačítka a jiné prvky v dané aplikaci balíčku Office 365. Tyto prvky mají za úkol vykonání nějakého úkolu v určité aplikaci balíčku Microsoft Office. Jediná nutná podmínka pro programování ve VBA je mít platnou licenci balíčku Microsoft Office. Tento programovací jazyk byl často nazýván „makro jazyk“, to bylo odvozeno od jeho využití, které bylo charakteristické automatizací sekvenčních příkazů v aplikacích pro zpracování textu a v tabulkových procesorech. S uvedením verze Microsoft Office 5 se možnosti VBA dostaly daleko za hranice starších programovacích jazyků. Tato verze umožňovala vytvářet a ovládat objekty v Excelu a také byl umožněn přístup k ovladačům disků a k síti. Dnes je VBA hojně využíváno především v aplikaci Microsoft Excel (3, s. 3).



Obrázek 1: Panel vývojář

(Zdroj: Vlastní tvorba)

Předchůdcem dnešního VBA byl programovací jazyk Basic, kde název představoval zkratku „Beginner’s All-purpose Symbolic Instruction Code“, což lze přeložit jako „všestranný symbolický instrukční kód pro začátečníky“. Jazyk Basic byl vyvíjen v šedesátých letech 20. století, kdy se po dvě desetiletí stal široce používaným, a to zejména pro svoji jednoduchost. Po dlouhém vývoji jazyka VBA byl v roce 1995 představen, s aplikací Excel 5, i programovací jazyk VBA (3, s. 4).

Makra nebo programy ve VBA se nejčastěji používají k úkolům, které se periodicky opakují nebo jsou příliš složité, aby je někdo dělal manuálně, což ve výsledku není příliš efektivní. Pro jazyk VBA není nutné, aby uživatel byl zdatný programátor, příkazy jsou relativně jednoduché na implementaci a customizaci dle potřeb daného úkolu. Jedním z využití

je automatizace opakovaných úkolů, kdy uživatel potřebuje data či reporty nějakým způsobem nahrát, zpracovat a zformátovat. Pro vyřešení tohoto úkolu lze vytvořit makro, kdy report, pomocí jednoduchého spuštění makra, data sám upraví a zformátuje. Když dojde ke změně zdrojových dat, uživatel nemusí opětovně data znovu upravovat, ale pouze provede spuštění makra. Dalším využitím makra je situace, kdy uživatel potřebuje stejnou činnost vykonat pro více pracovních listů nebo souborů ve stejné složce (3, s. 5).

Pro příklad lze využít i zaměření téhle práce, kde jsou zpracovávány pracovní procesy skladů ve firmě, kdy z informačního systému jsou exportovány čtyři reporty skladových zásob pro čtyři různé sklady. Každý report je potřeba upravit na stejné výsledné formátování. Tato úprava čtyř zdrojových souborů je příkladem využití VBA pro efektivní zpracování úkolů, které se opakují.

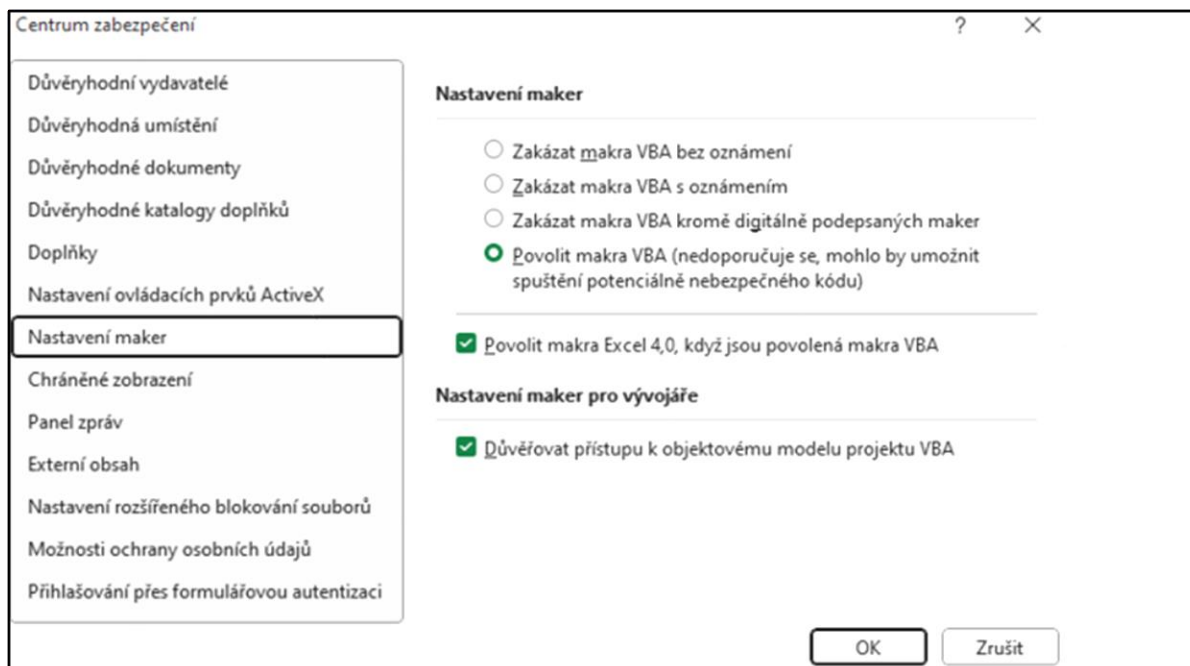
Mezi další výhody VBA můžeme zařadit situaci, kdy pokročilý uživatel naprogramuje makro vykonávající určité úkoly, které by byly pro jeho kolegu příliš těžké až neřešitelné vykonat. Méně pokročilý uživatel pomocí jednoduchého spuštění makra dokáže vykonat pro něj neřešitelné úkoly. Pokud nebude makro správně navrženo a nastaveno, může dojít k neočekávanému chování, což by mělo za následek přerušení makra a výpis chyby. V případě chyby je potřeba opravy nejlépe uživatelem, který aplikaci implementoval.

Mezi nevýhodu využití jazyka VBA lze zařadit požadavek na určitou úroveň programovacích dovedností uživatele. Jazyk VBA je řazen mezi ty méně náročné. Velmi složité je zařízení správného chodu programu za všech okolností, na všech zařízeních, v jakémkoli čase. V případě vyskytnutí se chyby běžný uživatel potřebuje technickou podporu, kdy mu zkušenější uživatel musí chybu opravit. K odhalování chyb se využívá proces ladění programu, známý pod pojmem „debugging“. Pro uvedení příkladu chyby se může vyskytnout problém, kdy dojde k přenesení makra na jiný počítač, na kterém makro nemusí fungovat. To může být

zapříčiněno špatnou kompatibilitou mezi verzemi aplikace Excel nebo napojením makra na systémové soubory a informace a uživatelích.

1.2.1 Zabezpečení makra a aplikací VBA

S verzí Office 2010 Microsoft představil důležitý krok vpřed v zabezpečení souborů s podporou maker. Tato změna se týká ověřených souborů. Ověřeným souborem je označován soubor s povolením uživatele používat makra. Po otevření sešitu se pod pásem karet zobrazí upozorňující žlutý pás, oznamující přítomnost maker v tomto souboru a s žádostí o povolení těchto maker. Akceptováním maker se stane soubor důvěryhodným. To zapříčiní to, že v budoucnu už nebude hláška dále zobrazována. V Excelu jde nastavit důvěryhodnou složku, v níž se nachází pouze důvěryhodné soubory. Po otevření těchto souborů se již nezobrazí vyzývací zpráva o povolení maker. Makra, v souborech nacházejících se v důvěryhodné složce, jsou automaticky povolována (5, s. 28).



Obrázek 2: Zabezpečení maker

(Zdroj: Vlastní tvorba)

1.2.2 Uživatelské prostředí

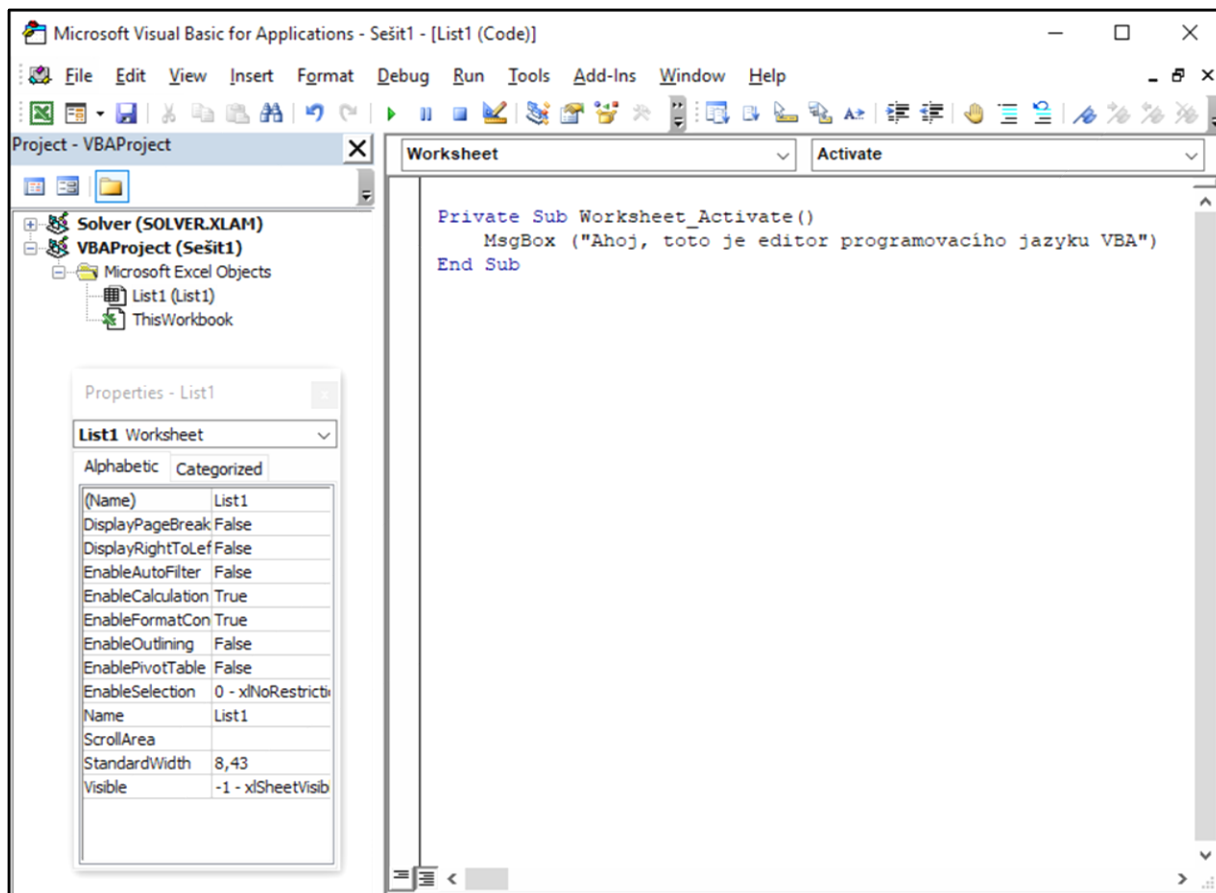
Po otevření Visual Basic for Application editoru se na levé straně obrazovky zobrazuje projektový průzkumník, ve kterém lze najít všechny součásti projektu. Zobrazení je hierarchicky uspořádáno. Nahoře je větev s pracovním sešitem a pracovními listy. Následuje

větev s objekty, které lze do projektu přidat. Jedná se o uživatelské formuláře, moduly a moduly tříd. V tomto podokně lze najít také tři tlačítka. Po kliknutí na první se zobrazí kód označené části projektu. Vedle něj je tlačítko pro zobrazení daného objektu, určené pro grafický náhled. Poslední tlačítko je pro změnu zobrazení objektů projektu v průzkumníkovi, a to buď hierarchicky nebo abecedně (6, s. 19).

Všechny Microsoft Excel objekty, uživatelské formuláře, moduly a moduly tříd se ukládají do jednoho souboru. Je zde i možnost jednotlivé objekty exportovat do samostatného souboru. Tento soubor je možné poté importovat do nového projektu. Uživatel může, už jednou napsaný kód, znovu použít, aniž by musel kód znovu psát. K této souborové práci jsou určeny souborové typy „CLS“ pro Microsoft Excel objekty a moduly tříd, „BAS“ pro běžné moduly a „FRM“ pro uživatelské formuláře (6, s. 20).

Hlavní částí okna je textový editor, podobný ostatním textovým editorům. V tomto editoru lze pohybovat kurzorem, psát, editovat, mazat, kopírovat, vložit text a mnoho dalšího. Editor po dopsání řádku kódu automaticky kontroluje syntaktické chyby. V případě výskytu syntaktické chyby, barevným zvýrazněním části kódu upozorní uživatele na její výskyt. Při psaní kódu editor zobrazuje seznam položek, které by z logiky jazyka VBA mohly následovat v psaní kódu. Pokud si ze seznamu uživatel nevybere, může pokračovat psát konkrétní věc nenacházející se v seznamu. Další funkcí je zobrazení informací o funkcích a jejich parametrech. Editor po odskočení na nový řádek zarovná text, podle odsazení řádku nad ním, popřípadě lze pomocí příslušných kláves s odsazením pohybovat. Klíčová slova jazyka VBA

jsou v tomto editoru barevně zvýrazňována. Všechny tyto funkce lze vypnout, zapnout nebo editovat v nastavení (6, s. 20-21).



Obrázek 3: MS Excel editor VBA

(Zdroj: Vlastní tvorba)

1.2.3 Proměnné

V programovacím jazyku VBA se pod pojmem proměnná rozumí dočasné uložení dat. Do definovaných proměnných můžete ukládat text, čísla, datумы a další různé typy. V praxi to pak vypadá tak, že si v paměti vytvoříme místo pro data, kterému přiřadíme charakteristický název. Pod tímto názvem můžeme přistupovat k informacím, které jsme získali nějakým výpočtem nebo interakcí od uživatele. Data je potřeba členit do datových typů, aby s nimi mohl kompilátor jazyka VBA pracovat efektivněji. Definování proměnných je možno dvěma způsoby, implicitně a explicitně. Pro deklarování explicitním způsobem lze použít výraz „Dim“, za ním následuje název proměnné a její datový typ. Druhou možností je použití implicitně deklarované proměnné. Implicitní deklarace proměnné je její použití v kódu bez předchozího deklarování. Tato možnost není ovšem doporučovaná, protože s sebou nese jistá

rizika, například použití špatného názvu proměnné. To se v programu projeví tím, že při překlepu v názvu proměnné dojde při kompilaci programu k vytvoření nové proměnné, místo toho, aby použil data uložená již v existující proměnné (7, s. 302-308).

Názvy proměnných by měly být výstižné, aby bylo možné již podle názvu určit, co se v proměnné uchovává za informace. Při tvorbě názvů uživatel nepoužívá diakritiku, a pro oddělení slov používá podtržítka. Druhou možností je, že každé první písmeno v novém slově zapíše uživatel velkým písmem. Důležitou vlastností proměnných je platnost. Platnost proměnných je určena použitím klíčových slov. Na základě umístění deklarace v kódu existují dva typy použití deklarace pomocí klíčového slova „Dim“. Podle umístění je proměnná k dispozici v jedné proceduře určitého modulu nebo v procedurách v rámci jednoho modulu. Klíčovým slovem „Private“ je platnost zaručena pro všechny procedury v daném modulu. Typem platnosti „Public“ se platnost zachovává ve všech modulech a procedurách daného projektu, a to i po skončení dané procedury. Posledním typem proměnných je typ platnosti „Static“. Tento typ platnosti je vhodný pro procedury, kdy po jejich ukončení se zachovává a při opětovném spuštění kódu procedury se hodnota například přičítá od hodnoty, na které skončila při posledním běhu (7, s. 302-308).

Programovací jazyk VBA obsahuje 13 datových typů. Další datové typy si může sám uživatel vytvářet. Níže v tabulce je seznam datových typů s rozsahem hodnot, které daný typ že uchovávat a počet bajtů v paměti, které jsou dané proměnné daného datového typu rezervovány (9).

Tabulka 1: Datové typy

(Zdroj: Vlastní tvorba podle (9))

Datové typy	Počet bajtů	Rozsah hodnot
Boolean	2 bajty	TRUE/FALSE
Byte	1 bajt	0 – 255
Integer	2 bajty	–32.768 – 32.767
Long	4 bajty	–2.147.483.648 -
Single	4 bajty	
Double	8 bajtů	
Currency	8 bajtů	
Decimal	14 bajtů	
Date	8 bajtů	1.leden 0100 – 31.prosinec
Object	4 bajty	Odkaz
String (proměnná délka)	10 bajtů + délka	0 – 2miliardy
String (pevná délka)	délka řetězce	1 až 65000
Variant (s čísly)	16 bajtů	
Variant (se znaky)	22 bajtů + délka	0 – 2miliardy

1.2.4 Konstrukce jazyka VBA

V každém programování je zapotřebí se na základě vstupních dat rozhodovat, jak se bude aplikace v dané situaci chovat, a jaké výpočty bude provádět. Ve VBA je několik základních rozhodovacích konstrukcí. Jedná se například o konstrukci „If..Then...ElseIf...Else...End If“ nebo konstrukci „Select Case“. Pro opakování jedné činnosti n-krát použijeme ve VBA cykly. Mezi nejčastější cykly se řadí „For Next“, „For Each Next“ a „Do Loop“. S těmito konstrukcemi jsou spjaty i operátory, zejména ty porovnávací a logické operátory (7, s. 39).

Tabulka 2 : Porovnávací operátory VBA

(Zdroj: Vlastní tvorba podle (9))

Operátor	Význam
Is	Porovnání proměnných odkazujících na objekty
Like	Porovnání řetězců
<	Menší než
>	Větší než
>=	Větší nebo rovná se
<=	Menší nebo rovná se
<>	Není rovno
=	Rovno
And	Logické And
Not	Logické Not
Or	Logické Or
Xor	Logické Xor

- Konstrukce If...Then...ElseIf...End If

Tato konstrukce je užívána k rozhodování s více podmínkami. Kód se provádí do doby, než dojde ke shodě v podmínce „ElseIf“, nebo kód dojde k příkazu „Else“. Dále v tomto bloku vykoná příkazy (7, s. 39-40).

```
If podminka1 Then
    'vykonej kod1
ElseIf podminka2 Then
    'vykonej kod2
Else
    'vykonej kod3
End If
```

Obrázek 4: Konstrukce If...Then...ElseIf...End If

(Zdroj: Vlastní tvorba)

- Konstrukce Select Case

Přehlednější a efektivnější volbou příkazu If..Then...ElseIf...End If je příkaz Select Case. Jedná se o alternativu If...Then...Elseif...End If , která se využívá při větším množství podmínek (7, s. 41-42).

```
Select Case vyraz
    Case podminka1
        'vykonej kod1
    Case podminka2
        'vykonej kod2
    Case Else
        'Vykonaj kod3
End Select
```

Obrázek 5: Konstrukce Select Case

(Zdroj: Vlastní tvorba)

- Cyklus For Next

Základem cyklu „For Each“ je čítač, který je součástí této smyčky a při každém průchodu cyklem dojde ke zvýšení čítače o určený krok. Většinou je čítač zvyšován o jedničku, tak, aby prošel celé pole hodnot. Jakmile je čítač větší nebo se rovná počtu opakování, cyklus je ukončen (7, s. 44).

```
Dim citac As Integer      'Čítač pro postup cyklem
Dim krok As Integer      'Velikost jednoho kroku cyklu
Dim pocet_cyklu As Integer 'Celkový počet cyklů

For citac = 0 To pocet_cyklu Step krok
    'vykonej kod
Next citac
```

Obrázek 6: Cyklus For Next

(Zdroj: Vlastní tvorba)

- Cyklus For Each Next

Cyklus „For Each Next“ je určen zejména pro práci s objekty, které se vyskytují v kolekcích. Cyklus prochází všechny prvky v dané kolekci a vykoná určený příkaz. Jedná se o bezpečnější a rychlejší variantu kódu, než je složitější alternativa přes cyklus „For Next“ (7, s. 47).

```
Dim arr As Variant      'pole typu Variant
Dim element As Variant  'Prvek typu Variant

For Each element In arr
    'kod pro operace s prvky pole typu Variant
Next element
```

Obrázek 7: Cyklus For Each Next

(Zdroj: Vlastní tvorba)

- Cyklus Do..Loop

Méně častým cyklem používaným při programování ve VBA je cyklus „Do...Loop“. Tento cyklus se odliší od předešlých dvou v tom, že nemá přesně daný počet opakování. Cyklus se vykoná nejméně jednou, a poté se testuje podmínka za příkazem While. Tento příkaz pokračuje v cyklu, pokud je platná podmínka za tímto příkazem. Příkaz While lze zaměnit za Until. V tomto případě se pokračuje v cyklu, dokud je platná podmínka za klíčovým slovem Until. Příkazy s podmínkami je možno vložit za slovo „Do“ nebo také za slovo „loop“. Jestliže podmínka není splněna cyklus je ukončen (7, s. 49-51).

```
Dim i As Integer           'Pocet cyklu
i = 1                     'zacatek cyklu
Do While i <= 10          'podminka
    'kod s inkrementací promenne i
Loop
```

Obrázek 8: Cyklus Do While

(Zdroj: Vlastní tvorba)

1.2.5 Ošetření chyb

Výše zmíněné strukturované konstrukce mají pomocí klíčových slov definovaný svůj začátek a svůj konec. Podle běhu kódu, je přesně dané, jak program do cyklu vstoupí, co přesně vykoná, a jak cyklus opustí. Poté následují příkazy, které jsou uvedeny za klíčovým slovem ukončující daný cyklus. Pro ošetření neočekávaných chyb je možné použít příkazy řídicí běh kódu. Tyto příkazy mohou obsahovat návěští. Návěští je řádek v kódu s libovolným textem a dvojtečkou označující určité místo, kam se program přesune a pokračuje dále ve vykonávání kódu. Tyto příkazy není doporučeno používat. Při použití těchto příkazů hrozí horší čitelnost kódu a ztížení odhalování chyb. Tyto příkazy by se měly používat pouze výjimečně, například pro ošetření chyb za běhu programu, tzv. Run-time errors. Příkladem těchto chyb může být situace, kdy se program pokouší otevřít soubor, který se v daném adresáři nenachází. Nebo se může jednat o situaci, kdy program mylně spoléhá na to, že list s určitým názvem je aktivní a

program v něm chce pracovat. Kromě cyklů existují ještě další příkazy řídící běh kódu:(7, s. 52-60).

1.2.6 Funkce a procedury

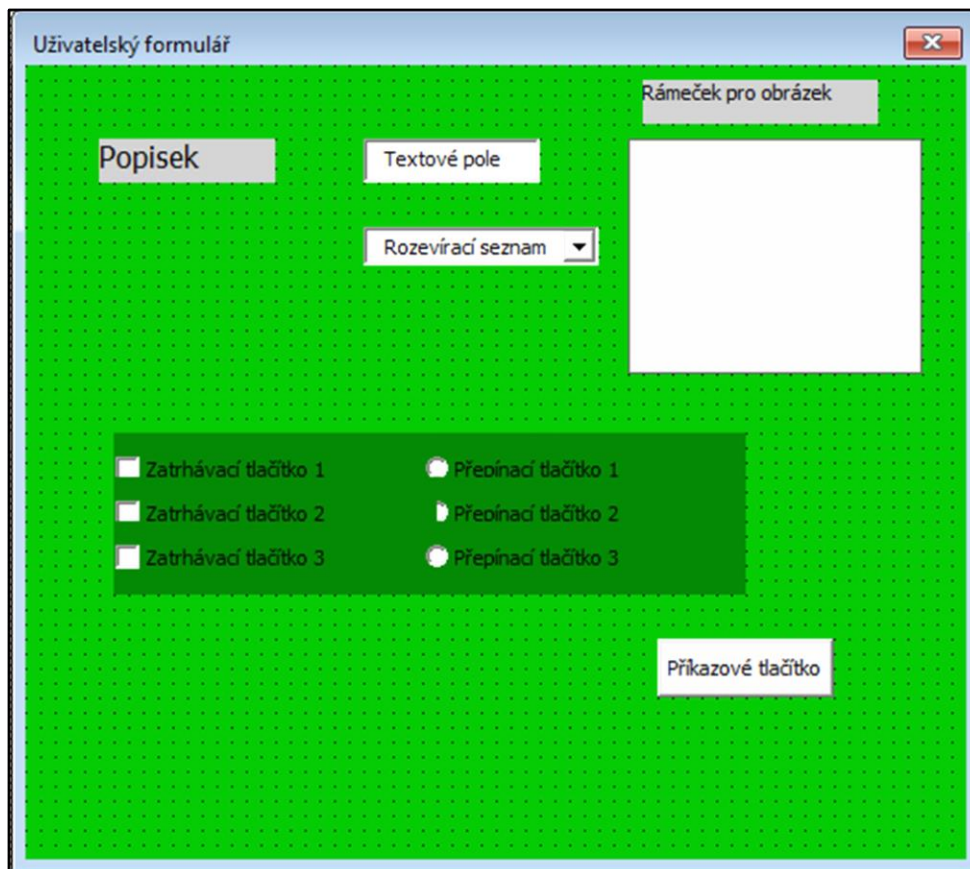
Pro spuštění kódu ve VBA je nutné mít k dispozici proceduru. Procedura v jazyku VBA zastává roli programu. Procedura může, ale nemusí, přijímat parametry. Procedura žádnou hodnotu nevrací. Deklarace procedur může mít charakter „Private“ nebo „Public“. Deklaraci „Private“ lze spustit pouze pomocí funkcí nebo jiných procedur v tomtéž modulu, ve kterém byla deklarována a definována. Proceduru s platností „Public“ jde spustit kromě modulu v kterém byla deklarována také v ostatních modulech, a přes uživatelské rozhraní. Hlavní rozdíl mezi funkcemi a procedurami je ten, že funkce vrací hodnotu, s kterou program může, ale nemusí dále pracovat (7, s.231).

Argumenty funkcím nebo procedurám jsou předávány pomocí odkazu či hodnotou. K tomuto účelu jsou určeny klíčová slova „ByRef“ a „ByVal“. Pokud je argument předán odkazem na původní proměnnou „ByRef“, funkce po provedení příkazů hodnotu původní proměnné změní. V opačném případě, případě předání hodnotou „ByVal“, dojde k vytvoření kopie a původní hodnota v proměnné zůstane zachována (7, s. 233).

1.2.7 Uživatelský formulář

Uživatelskou interakci lze v prostředí aplikace Excelu zachytit pomocí uživatelského formuláře. Uživatelský formulář se skládá ze třech hlavních částí. Hlavní částí je dialogové okno, podobající se dialogovým oknům z operačních systémů Windows. Toto okno je reprezentováno objektem uživatelského formuláře. Součástí formulářů jsou ovládací prvky, které plní vizuální a interakční funkci. Poslední částí uživatelského formuláře je kód jazyka VBA. Ve většině případů mají uživatelské formuláře svůj kód v jazyku VBA. V jednodušších případech se obejdou i bez kódu. Objekt uživatelský formulář, ale i jeho ovládací prvky jsou určeny vlastnostmi a metodami. Pomocí těchto vlastností a metod můžeme určit, jak se tento objekt či data na něj vázána budou chovat nebo jak se změní jejich grafická podoba. Většina těchto objektů dokáže detekovat události, které jsou většinou vyvolány uživatelskou interakcí. Pomocí kódu VBA dokáže formulář na tuto interakci zareagovat anebo přizpůsobí chování formuláře.

To v praxi může znamenat změnu dat na listu, změnu prvků ve formuláři nebo překopírování dat z jednoho listu na druhý (6, s. 239-240).



Obrázek 9: Uživatelský formulář

(Zdroj: Vlastní tvorba)

1.2.8 Ovládací prvky

V aplikaci Microsoft Excel existují dva typy ovládacích prvků, které nelze zaměňovat. Jedná se o prvky typu „ActiveX“ a „Forms“. Už ve verzi Excel 5.0 Microsoft představil ovládací prvky typu „Forms“, a i toto byl jeden z důvodů proč Microsoft z trhu vytlačil své konkurenty. Excel dodnes umožňuje na list vložit tyto ovládací prvky, které mohou integrovat s buňkami. Tyto prvky jsou stále používány a nesou označení „Forms Controls“. Většinou reagují na jednu interakci a mají kratší seznam vlastností. Tyto vlastnosti může uživatel upravovat. Pomocí vlastnosti „OnAction“ může uživatel prvku přiřadit proceduru. Ovládací prvky „ActiveX“ reagují na více událostí, na různou událost mohou reagovat různým

způsobem a také mají více upravovatelných vlastností. Pro vyvolání reakce po kliknutí na tlačítko musíte kód zapsat ručně (7).

Počet prvků „ActiveX“ je nekonečný, závisí na tom, které jsou na počítači nainstalované. Mezi nejčastěji užívané prvky „ActiveX“ lze zařadit:

- CommandButton

Po kliknutí na toto příkazové tlačítko program vykoná danou proceduru, která je definována v kódu (8).

- ComboBox

Tento prvek umožňuje výběr jedné položky z určitého seznamu položek. Seznam může definovat uživatel pomocí kódu nebo odkazem na buňky určitého listu (8).

- CheckBox

Zaškrťovací políčko má tři základní hodnoty, podle toho, zda je zaškrtnuto (True), zda není zaškrtnuto (False) nebo zda odkazuje na prázdnou buňku (Null) (8).

- OptionButton a RadioButton

Přepínače se většinou používají na zatrhnutí pouze jedné hodnoty v rámci jedné skupiny těchto prvků (8).

- ListBox

Prvek seznam je podobný prvku pole se seznamem, ovšem u seznamu lze vybrat jednu, ale i více hodnot najednou povolením vlastnosti MultiSelect (8).

- ScrollBar

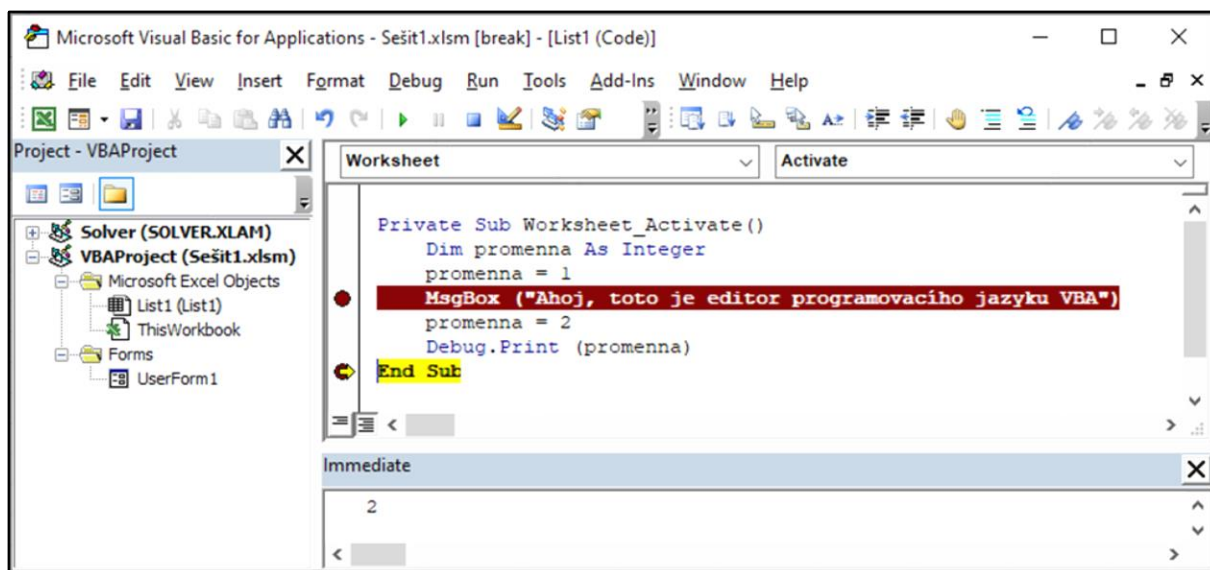
Dalším prvkem je posuvník, u kterého lze pomocí tažením myši měnit hodnotu. Pro změnu hodnoty lze využít i šipky, kde velikost skoku lze nastavit ve vlastnosti „SmallChange“. Tuto hodnotu si pak můžete někde uložit nebo zobrazit (8).

1.2.9 Debugging

Pro ladění chyb lze v prostředí Microsoft Excel jazyka VBA využít více metod. Jednou z metod je zkoumání kódu. Uživatel, který kód napsal, tak musí vědět, co a proč v jaký moment

kód vykonává. Pokud je kód složitější je vhodné postup kódu okomentovat v komentářích. Pokud uživatel chybu neodhalí zkoumáním, může pokračovat k metodě vložení „MsgBox“ funkce do kódu. Pomocí funkce „MsgBox“ si uživatel může nechat vypsát proměnné, u kterých má podezření, že způsobují selhání programu. Pokud se touto funkcí vypíše neočekávaná hodnota, uživatel pomocí klávesové zkratky „Ctrl+Break“ přesune do VBA editoru, kde kód může zkoumat. Alternativní možností je použít příkaz „Debug.Print“, kde si opět může uživatel nechat vypsát hodnoty některých proměnných. Výhodou tohoto příkazu je nepřerušování programu, možnost oddělovat proměnné čárkami a vypsání hodnot do ladícího okna. Dalším způsobem je použití nástroje VBA Debugger.

Důležitým nástrojem je bod zlomu, „Breakpoint“. Tento bod uživatel nastaví na určitý řádek, kód se tam zastaví a uživatel si může zkontrolovat jaké hodnoty proměnných mu do daného příkazu vstupují nebo z jiného příkazu byly vypočítány. „Breakpoint“ je v kódu zvýrazněn tmavě červenou barvou v šedém sloupci před začátkem daného řádku. Jednou z mnoha dalších možností je pro uživatel procházet řádek po řádku pomocí klávesy F8. Ideální je tuto možnost kombinovat s bodem zlomu (4, s. 208).



Obrázek 10: VBA Debugging

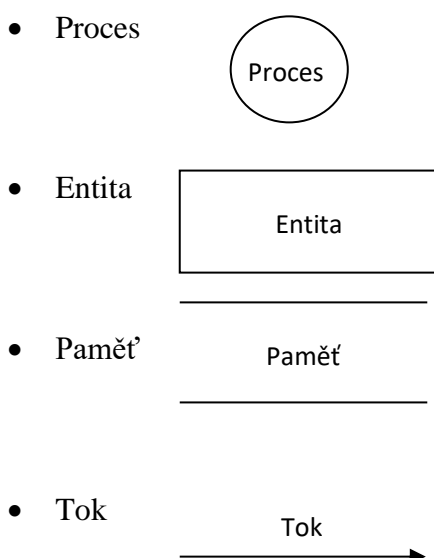
(Zdroj: Vlastní tvorba)

1.3 Funkční modelování

Funkční modelování zobrazuje toky dat mezi systémem a jeho okolím, a mezi funkcemi systému a daty ukládanými v něm. Funkčním modelováním lze zobrazit funkce systému. Nejčastějším modelem využívaným pro znázornění funkcí systému je Data Flow Diagram.

1.3.1 Diagram toku dat

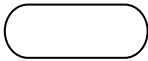
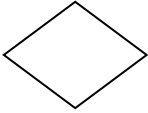


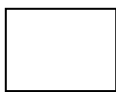
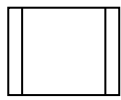

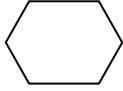
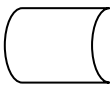

Data Flow Diagram je řazen mezi nejčastěji využívané modely funkčního modelování. V českém jazyce je označován diagramem datových toků. Pomocí něj lze modelovat chování systému. Diagramu datových toků může být doplněn kratším textem. Tento text popisuje znázorněnou funkcionalitu. Lze modelovat celý systém, ale také jeho podprocesy (10, s. 84).



1.3.2 Vývojový diagram

Vývojový diagram je určen pro modelování procesu, postupu, algoritmu nebo posloupnosti kroků. Lze ním graficky modelovat slovní popis toku procesů. Modelování je prováděno na základě jednoduchých geometrických tvarů. Každý tvar představuje různý aspekt procesu.

Praktickými příklady vývojového diagramu jsou popis procesu, výrobního procesu nebo algoritmu (11).

- Začátek/konec 
- Rozhodovací blok 
- Dokument 
- Propojení 
- Proces 
- Podproces 
- Vstup/Výstup dat 
- Cyklus 
- Uložení dat 
- Uložení dat / databáze 

1.4 Analytické metody

V rámci této kapitoly budou pospány analytické metody HOS 8 a SWOT analýza. Analytickou metoda HOS 8 lze hodnotit efektivnost informačních systémů. Analýza SWOT je metoda sloužící ke zhodnocení konkrétního záměru.

1.4.1 HOS 8 Analýza

S informačními systémy firem souvisí jejich efektivita a efektivnost. Informační systémy ve firmách mají za úkol podporovat firemní procesy a zefektivňovat je. To vše za účelem naplňování strategických cílů firmy. Aby byly cíle firmy naplněny je potřeba dělat správné věci správně. S tímto souvisí vlastnosti efektivita a efektivnost. Tyto vlastnosti jsou základem metody HOS 8. Metoda HOS 8 vychází z hodnocení jednotlivých složek informačních systémů. Každá jedna složka je hodnocena na základě čtyřbodové stupnice zobrazující: 1- špatná, 2- spíše špatná, 3- spíše dobrá, 4- dobrá. Pokud nějaká složka výrazně vybočuje k horšímu hodnocení, má to za následek horší úroveň systému jako celku. Cílem metody HOS 8 je určit slabá místa informačního systému. Výsledné hodnocení celého systému odpovídá k horšímu hodnocení, má to za následek horší úroveň systému jako celku. má to za následek horší úroveň systému jako celku. Cílem metody HOS 8 je určit slabá místa informačního systému. Výsledné hodnocení celého systému odpovídá hodnocení nejslabší části systému. Oblasti hodnocení informačního systému byly určeny na základě dlouhodobého zkoumání jejich důležitostí (13, s. 51-52).

- Hardware

V této části je zkoumáno technické vybavení firmy (13, s. 52).

- Orgware

Zkoumá pravidla, pracovní postupy a normy pro používání informačních systémů (13, s. 52).

- Software

Oblast software zjišťuje, zda programové vybavení firmy je efektivní, zkoumá i jeho funkce a jednoduchost použití (13, s. 52).

- Peopleware

Zkoumá pracovníky, kteří s informačním systémem přijdou do kontaktu. Důraz je kladen na povinnosti pracovníků vůči informačnímu systému (13, s. 52).

- Dataware

Hodnotí správu, zabezpečení a dostupnost dat (13, s. 52).

- Customers

Zkoumání jsou zákazníci, například uživatelé e-shopů, nebo pracovníci firmy, kteří potřebují data ze systému pro svoji práci (13, s. 52).

- Suppliers

Existují dvě možnosti dodavatelů informačního systému. Rozlišují se podle toho, kdo se o informační systém stará. Mohou to být interní zaměstnanci firmy. V druhém případě může být informační systém provozován a podporován externí firmou (13, s. 52).

- Managment

Oblast managmentu je hodnocena ve vztahu k informační strategii a vnímání koncových uživatelů systému (13, s. 52).

Výsledné hodnocení metody HOS 8

Metoda má tři předem stanovené úrovně výsledného hodnocení. Vyvážený systém má optimální poměr účinnosti. Ve výsledku metody to znamená, že všechny hrany mají stejné hodnocení nebo nejvýše tři z nich vybočují od ostatních o maximálně o jeden bod. Nevyvážený systém je systém, kde alespoň jedna oblast je na nejvyšší úrovni, a zároveň alespoň jedna jiná oblast je na nejnižší úrovni. V tomto případě je celkové hodnocení dáno nejslabším článkem. Pokud je na některé ose dosažena určitá hodnota a rozdíl mezi touto hodnotou a celkovou úrovní je větší než 1, tak se jedná o velmi nevyvážený systém (13, s. 52).

1.4.2 SWOT analýza

SWOT analýza je nástroj určený k analýze vnitřního a venkovního prostředí za pomoci silných a slabých stránek. Silnými stránkami SWOT analýzy je stručnost, přehlednost a komplexnost. Z hlediska vnitřního prostředí popisuje SWOT analýza silné (Strengths) a slabé

(Weaknesses) stránky. Externí vlivy jsou popsány za pomoci příležitostí (Opportunities) a hrozeb (Threats) (14).

Silné stránky

Silné stránky by měly zobrazovat nějakou konkurenční výhodu, kterou může být například nějaký technologický postup, patent či inovace (14).

Slabé stránky

Slabé stránky představují věci, v čem firma zaostává oproti konkurenci. Může se jednat o absenci inovací, výzkumu, omezená výrobní kapacita, málo kvalifikované pracovní síly a další (14).

Příležitosti

Příležitosti musí firma umět správně identifikovat a ve správný čas je využít. Například produkt, který firma vyrábí bude dotován státem, tak firma musí udělat vše pro to, aby mohla svého produktu prodat co nejvyšší množství (14).

Hrozby

Negativní událost, která může firmu nějak poškodit označujeme termínem hrozba. Z hlediska SWOT analýzy lze za hrozby považovat situace, kdy dojde k nějakému jevu, který firmu poškodí, zapříčiní snížení zisku nebo zapříčiní nenaplnění strategických cílů. Může se jednat o zdražení vstupních surovin k výrobě produktů, krach obchodních partnerů nebo vznik nové konkurence na trhu (14).

2. Analýza současného stav

V druhé kapitole se bude práce věnovat popisu společnosti, předmětu podnikání a organizační struktuře ve společnosti. Dále si určíme příležitosti, hrozby, silné a slabé stránky problematiky řešené v této bakalářské práci pomocí analýzy SWOT, a poté pomocí metody HOS 8 provedeme analýzu informačního systému a softwaru používaného ve skladu náhradních dílů.

2.1 O společnosti

Firma HARTMANN-RICO a.s. je akciová společnost zaměřující se na výrobu zdravotnických potřeb. Firma Rico a.s. se ve Veverské Bítýšce formovala od počátku 20. století, nejdříve jako úpravna kaolínu, později se přeorientovala na výrobu zdravotnických potřeb, které se věnuje až dodnes. V druhé vývojové linii se začala formovat již v ranné době 19. století v německém Heidenheimu, jako společnost Paul Hartmann. V roce 1991 došlo ke spojení výše zmíněných společností Rico a.s. a Paul Hartmann a to zapříčinilo, že během další dekády se společnost stala největším českým výrobcem zdravotnických potřeb. Společnost do dneška nese název HARTMANN-RICO a.s., který vznikl při sloučení německé a české vývojové větve. Společnost má také další závody na území České republiky, a to v Havlíčkově Brodu a Chvalkovicích, logisticky rozmístěné sklady v Rajhradu u Brna, Tuchoměřice nacházející se nedaleko Prahy (1).

2.1.1 Předmět podnikání

Firma HARTMANN-RICO a.s. se zabývá výrobou zdravotnického materiálu, od nejrůznějších operačních setů a Medisetů. Medisetý a operační sety se používají na operačních sálech, kdy je důraz kladen na správné a úplné sestavení všech komponent v tomto setu, aby na operačním sále něco nechybělo nebo se v setu nacházela kontaminovaný předmět. I proto je ve výrobě kladen důraz na hygienu, kdy při vstupu do výroby pro zaměstnance platí striktní hygienická pravidla. Společnost produkuje i hygienické potřeby, náplasti Cosmos, tlakoměry, teploměry a další diagnostické přístroje. V dnešní době nedílnou součástí sortimentu jsou také

dezinfekční prostředky, rukavice, autolékárničky, tělová kosmetika, ale i dámská hygiena a mnoho dalšího (2).

2.1.2 Struktura společnosti

Tato společnost na území České republiky zaměstnává zhruba 1700 zaměstnanců, kteří pracují buď ve výrobních závodech (Veverská Bítýška, Havlíčkův Brod, Chvalkovice), ve skladech (Rajhrad u Brna, Tuchoměřice u Prahy) a v kancelářských komplexech v Praze a Brně. V čele firmy jsou tři představitelé firmy. Dva zastupují roli jednatele firmy a třetí představuje roli technického a výrobního ředitele. V nižší úrovni organizační struktury společnosti jsou dva ředitelé pro závody v rámci Veverské Bítýšky, jeden pro závod Chvalkovice a jeden pro závod Havlíčkův Brod. Pod touto úrovní už jsou dále manažeři jednotlivých oddělení či výrobních provozů (1).

2.1.3 Zaměření práce

Tato práce se zabývá optimalizováním procesů ve skladu náhradních dílů. K tomu bude sloužit aplikace vytvořená v prostředí MS Excel pomocí programovacího jazyka VBA. Níže se práce bude věnovat SWOT analýze z hlediska procesu tvorby materiálových čísel, jejich názvosloví, kategorizací a nástrojů využívaných v rámci pracovních úkonů skladu náhradních dílů.

2.2 Metoda HOS 8

V této kapitole bude provedeno zhodnocení stavu informačního systému a dalších oblastí pomocí metody HOS 8.

Hardware

Sklad je vybaven stolními počítači a monitory od firmy Dell. Pro účely skladníků jsou počítače vybaveny dvěma monitory, které zajišťují pro efektivnější práci, protože jejich práce vyžaduje většinou vykonávat více úkolů najednou nebo čerpat data z více zdrojů. Ostatní stolní počítače pro vyhledávání náhradních dílů nejsou příliš výkonné, ovšem díky tomu, že vyhledávač je provozován ve webovém prohlížeči je toto vybavení dostačující. Ve skladu jsou používány dále notebooky, taktéž od firmy Dell, které jsou dostatečně výkonné, ale jedná se o čtrnáctipalcové notebooky. Jejich velikost má výhodu ve své mobilitě, ovšem při práci na psacím stole jsou povětšinou nedostačující pro příliš malou obrazovku. Ve firmě je

provozována 1Gbit síť s reálnou rychlostí 100Mbit. Tato síť je dostatečně nadimenzovaná, protože většina pracovních procesů se provádí přes informační systém SAP nebo se zapisují data do souborů za pomoci nejrůznějších MS Office programů, které nemají příliš vysoký nárok na rychlost sítě. Veškeré připojení na internet je zprostředkováno za pomoci Virtual Private Network (VPN), za účelem vyšší bezpečnosti informační infrastruktury firmy.

Software

V této firmě je hlavní informační systém SAP, který je hlavní informační systém pro většinu procesů, od výrobních, přes mzdové, až po procesy skladové nebo logistické. Z tohoto systému lze data exportovat do vyhledavače, ovšem některá data je potřeba doplnit či pozměnit. Protože v systému SAP, jsou například exportovány pouze krátké názvy, které mají maximální délku čtyřicet znaků. Jen zřídka je pro správné a úplné pojmenování dostačujících čtyřicet znaků. A proto je dále využíván program MS Excel, v kterém se názvy dotváří a doplňují se tam další informace. Dále se v tomto sešitu doplňují kategorie až do sedmé úrovně.

Orgware

Ve skladu náhradních dílů je vyžadováno, aby na každý nově vzniklý problém či unikátní situaci bylo vytvořeno pravidlo, jakým se má daná situace vyřešit. Všechny operace s díly musí být promyšleny do detailu, aby se později nevyskytl problém, například v účetnictví při spotřebování náhradního dílu na stroj. Pro každou SAP transakci je potřeba se držet vytvořených návodů, například při přeskladnění, kdy při nezadání jednoho parametru může dojít k nedokončení procesu přeskladnění a blokaci dílu. Při příjmu nových náhradních dílů se skladníci řídí podle směrnice, která udává, v jakém obalu musí být konkrétní díl ve skladu zaskladněn, aby na něj nepůsobily vnější vlivy a díl nedegradoval. Skladníci se řídí podle směrnice, která udává pravidla pro tvorbu názvů pro jednotlivé kategorie dílů.

Peopleware

Každý nový zaměstnanec je zaškolen na informační systém a práci v něm. Dále má možnost v případě jakýchkoliv dotazů se zeptat kolegů nebo nahlédnout do směrnic či návodů k dané problematice. V případě nějakých jiných problémů je zaměstnanec zaškolen na to, jak

si podat požadavek na IT oddělení, kde mu na helpdesku většinou během jednoho pracovního dne odpoví a problém se společně pokusí vyřešit.

Dataware

Každý zaměstnanec má svůj vlastní účet na MS Office 365, a v rámci tohoto předplatného má i přístup ke cloudu Onedrive, kde si může ukládat svá pracovní data a mít k nim odkudkoli bezpečný přístup. Pro data, která jsou ve firmě sdílena je využíván síťový firemní server s ochranou firewall.

Customers

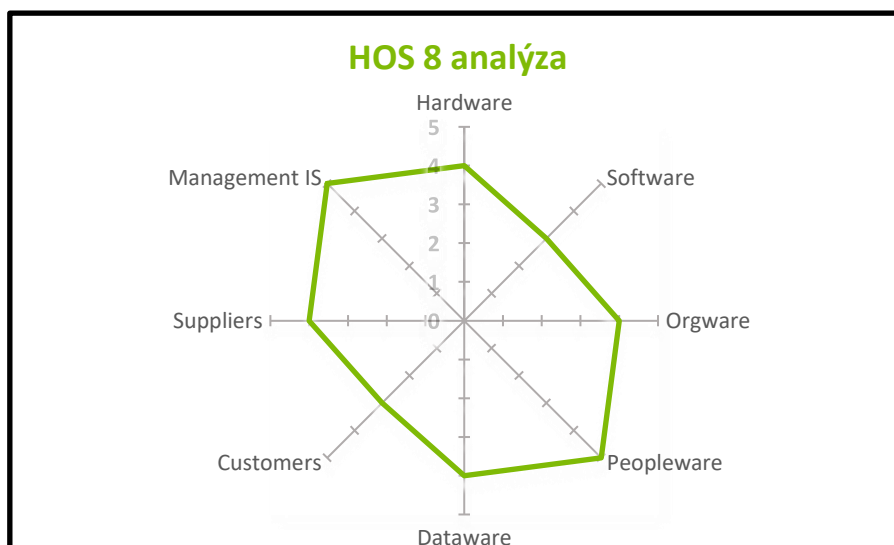
Jako zákazníci jsou v tomto případě zaměstnanci firmy, přesněji skladníci a ti mají do informačního systému přístup a pracují s ním. Zaměstnanci jsou v tomto systému zaučeni, když nastane nějaká méně častá situace, řídí se podle návodů, popřípadě je potřeba takový návod vytvořit.

Suppliers

V případě informačního systému si firma provoz a údržbu systému zajišťuje sama skrze pracovníky IT oddělení. Pokud budeme hovořit o vyhledávači, tak k jeho údržbu zařizuje dodavatelská firma.

Management IS

Management informačního systému a informačních technologií má v této firmě na starosti IT oddělení, kde každý využívaný software má svého specialisty.



Obrázek 11: Diagram HOS 8

(Zdroj: Vlastní tvorba)

Tabulka 3: Tabulka HOS 8

(Zdroj: Vlastní tvorba)

Oblast	Číselné hodnocení	Slovní hodnocení
Hardware	4	Vysoká úroveň
Software	3	Střední úroveň
Orgware	4	Vysoká úroveň
Peopleware	5	Velmi vysoká úroveň
Dataware	4	Vysoká úroveň
Customers	3	Střední úroveň
Suppliers	4	Vysoká úroveň
Management IS	5	Velmi vysoká úroveň

2.3 SWOT analýza

V této metodě je analyzován současný stav tvorby názvů, údržby databáze materiálových čísel, konzistentnosti, úplnosti a správnosti. Databáze je vedena v sešitu MS Excel, do kterého má přístup každý zaměstnanec skladu. Tento sešit je uložen na cloudu Onedrive, v kterém jsou veškerá data týkající se vyhledavače náhradních dílů. Nástroj vyhledávač náhradních dílů je vytvořen v programu MS Power BI, který si všechny potřebná data nahrává z cloudu. Jedná se o exporty z programu SAP, které zahrnují data:

- seznam všech SAP čísel náhradních dílů
- aktuální zásoby SAP čísel s počty kusů a pozicemi ve všech skladech
- nastavené minimální držené zásoby na skladě
- jednotkové ceny náhradních dílů
- celková peněžní suma konkrétní skladové zásoby konkrétního náhradního dílu
- kusovníky náhradních dílů na stroje

Vyhledavač umožňuje pracovníkům technického oddělení, rychlé vyhledání potřebného náhradního dílu. Techničtí pracovníci potřebují co nejrychleji náhradní díl najít, aby mohli stroj opět uvést do provozu s minimální dobou odstávky. K tomuto požadavku je nutná konzistentnost, úplnost a aktuálnost dat v databázi.

Silné stránky

Mezi silné stránky řadíme zejména jednoduchost rychle aplikovat změny, protože sešit Excel je multiplatformní a je uložen na cloudu, ke kterému má přístup více uživatelů v jeden moment a mohou tam provádět změny nebo přidávat nové náhradní díly. A proto k tomuto souboru mají udělené osoby přístup odkudkoli a prakticky z jakéhokoli zařízení. Tento excelovský sešit, v kterém se nachází seznam materiálových čísel, názvů a jednotlivých kategorií daného náhradního dílu, je pro skladníky přehledný a umožňuje provádět změny velice rychle a jednoduše. Vyplňování kategorií je zde umožněno za pomoci napovídání již zadaných hodnot na předchozích řádcích v rámci aktuálního listu. Každý náhradní díl má mít i fotodokumentaci, aby technik ve vyhledavači rozpoznal, zda se opravdu jedná o díl, který hledá. Jestliže díl foto nemá, tak ve vyhledavači je k zobrazení seznam všech náhradních dílů bez

fotodokumentace. Další výhodou tohoto vyhledavače je možnost rychle reagovat na požadavek navýšení minimální zásoby, který zadá technik na základě vlastního uvážení, kdy bere v potaz i cenu dílu, která je ve vyhledavači zobrazena. Vyhledavač potřebuje každý den stahovat aktuální data z informačního systému SAP. K tomuto je vytvořena aplikace ve VBA, kdy se všechny reporty stahují automaticky na základě skriptu. Tím odpadá každodenní zdlouhavá práce stahovat každý jednotlivý report zvlášť. Za účelem minimalizace odstávky strojů z důvodů poruch je potřeba, aby data v aplikaci byly aktuální, konzistentní a úplná. Aktuálnosti dat dosáhneme každodenním poloautomatickým aktualizováním reportů z informačního systému SAP.

Slabé stránky

Mezi slabé stránky řadíme nemožnost exportu reportu seznamu dílů, u kterých chybí fotodokumentace. Fotodokumentace je důležitá pro techniky, kteří fotky využívají při hledání konkrétního dílu. Tento problém se může vyskytnout v moment, kdy přijde nový díl, který ještě nebyl vyfocen nebo díl vyfocen byl, ale fotky nebyly nahrány do příslušného adresáře. Ve skladu se nenachází firemní telefon určený čistě pro fotodokumentaci, jenž bude mít přímé zálohování fotek do adresářové cesty v cloudu určené pro fotky pro vyhledávač. Další slabou stránkou je tvorba názvů, jeden skladník může například použít originální a předem domluvené správné slovo. Pro příklad lze uvést kategorii snímáčů, kdy může skladník použít slovo „Snímáč“, ale pokud název bude tvořit někdo, kdo nemá přehled ve tvorbě názvů, tak může ten stejný díl nazvat synonymem „čidlo“. Také může dojít prohození pořadí atributů v názvu. Jistou prevencí této chyby je vyhledání pravidel pro tuto danou kategorii ve směrnici pro tvorbu názvosloví. Skladník může ve vyhledavači najít již existující názvy a podle těch dílů vytvořit správný název. Tato metoda je ovšem zdlouhavá a v některých případech se může jednat i o těžko proveditelný úkol.

Příležitosti

Aplikace pro tvorbu názvosloví může zefektivnit proces vytváření nových názvů. Přínosem této funkce bude úspora času. Další příležitostí je zautomatizovaný proces tvorby a zpracování reportu skladových zásob jednotlivých materiálů s názvy, kategoriemi, informacemi, zda je vytvořena fotodokumentace a upozorněním, zda nějaká data v datové struktuře vyhledavače chybí. Vytvořením makra pro zpracování reportu všech fotodokumentací

všech materiálových číslech lze dosáhnout časové úspory a možnost tento proces zpřístupnit i méně zdatným uživatelům.

Hrozby

Hrozbou, která toto může ohrozit je přetížení databázového serveru sloužícího pro ukládání dat informačního systému SAP, to by způsobilo, že by vyhledavač nenačítal aktuální data, data by byla ze dne poslední aktualizace. Dalšími možnými hrozbami jsou bezpečnostní hrozby informačních technologií, ať už se jedná o malware či jiný hackerský útok, který by mohl způsobit ztrátu či modifikaci dat. Hrozba nekonzistentnosti a neúplnosti dat se může projevit výskytem duplicity nebo nákupem dílu od dodavatele stroje, na který je díl určen. Dodavatel stroje obvykle nabízí vyšší cenu než výrobce. Eliminací duplicit a zajištěním konzistentnosti a úplnosti dat může dojít k optimalizaci skladových zásob a snížení nákladů na sklad.



Obrázek 12: SWOT analýza

(Zdroj: Vlastní tvorba)

2.4 Zhodnocení analýz

Z provedených analýz bylo zjištěno, že se ve firmě používá jeden hlavní informační systém, ze kterého pak zaměstnanci data exportují. Tyto data využívají k uživatelsky intuitivnějšímu zobrazování, a poté i k dalším pracovním procesům s daty. Z analýzy vyplynulo, že ve skladu je velké množství dílů a není jednoduché umět každý díl správně

pojmenovat. K tomuto procesu tvorby názvosloví chybí nástroj, který by skladníkům radil, jak názvy pro náhradní díly vytvářet, a jaké všechny atributy, a v jakém pořadí je do názvu zadávat. Za účelem zlepšení tohoto procesu je doporučeno vyvinout aplikaci, která bude na základě kategorie náhradního dílu uživateli napovídat, jaké atributy a v jakém pořadí má zadávat. Tímto způsobem bude možno mít data úplná a konzistentní, údržbáři díky tomu zkrátí čas opravy stroje. Zamezí se tvorbě duplicit, což bude mít za následek optimalizaci zásob a tím dojde k snížení nákladů na sklad. V neposlední řadě to také bude mít přínos v pořizovací ceně náhradního dílu. Díky úplnému názvu dojde k jednodušší identifikaci výrobce daného dílu a zamezí se koupi dílu za vyšší cenu od dodavatele daného stroje. Jako další důsledek lze vyzdvihnout zefektivnění práce skladníků.

2.1 Existující alternativy

V této kapitole bude popsán návrh na tvorbu aplikace, která by sloužila jako šablona pro tvorbu názvů nových náhradních dílů. Aplikace by byla implementována v programovacím jazyku VBA. Na základě stanovených pravidel, která jsou popsána ve směrnici pro tvorbu názvů náhradních dílů by aplikace vytvořila formulář, do kterého by uživatel zapisoval konkrétní atributy, konkrétního náhradního dílu, řazeného do konkrétní kategorie. To by znamenalo značnou úsporu času uživatele, zaručení konzistentnosti a úplnosti názvů náhradních dílů.

Jelikož by byla tato aplikace vytvářena již na základě vytvořeného datového modelu, který je dán aplikací vyhledavače náhradních dílů, tak by, při implementaci jakéhokoliv již existujícího řešení, muselo dojít k úpravám implementace softwaru pro tento konkrétní případ. V informačním systému SAP a objednávkovém systému používaném v této společnosti je vedena veškerá agenda týkající se skladových zásob a objednávek náhradních dílů. Nabízelo by se zde řešení implementovat nový modul systému SAP. V tomto modulu by se dalo navrhnout i další funkcionality, například by se tento modul za pomoci strojového učení učil a později by dokázal navrhnout nové názvy náhradních dílů. K této funkcionalitě by bylo vyžadováno velké množství dat, na kterých by se nový modul mohl učit. I při implementaci nového SAP modulu na úrovni této práce by se jedlo o nákladnější řešení, než tomu bylo v případě vývoje aplikace ve VBA.

3. Vlastní návrhy řešení

Třetí kapitolou jsou vlastní návrhy řešení. V této kapitole budou popsány datové a funkční modely, existující alternativy, podrobný popis vlastní aplikace, ekonomické zhodnocení a návrh vývoje aplikace do budoucna.

3.1 Datové modelování

Správná struktura dat je důležitá pro efektivní zpracovávání dat v aplikaci. Nejprve bude popsáno datové modelování.

3.1.1 Struktura dat pro názvosloví a kategorizaci

Hlavní strukturou aplikace je tabulka obsahující materiálové číslo, které je také nazýváno „sapové číslo“. Toto číslo je přiřazeno při zakládání nového náhradního dílu v ERP systému, konkrétně tedy systému SAP. Základním pojmenováním každého materiálového čísla je krátký název materiálu, který je automaticky exportován z ERP systému. Jeho maximální délka je 40 znaků. Dlouhý název je název, který by měl obsahovat všechny potřebné atributy a označení. Tento název bude je vytvářen pomocí této aplikace. Dále je zde 7 úrovní kategorií, podle kterých je vytvářen dlouhý název pro vyhledavač náhradních dílů. Tento název by měl mít v dané kategorii jednoznačně určené atributy, které název musí obsahovat. Důvodem je snadnější vyhledávání náhradních dílů a rychlejší opravy strojů či infrastruktury. Tyto data jsou uchovávány v sešitu aplikace Microsoft Excel typu „xlsx“, ve kterém data uživatele upravují a přidávají nová. Každý nový náhradní díl je zapisován na nový řádek do prvního sloupce. V dalších sloupcích jsou zadávána další data, včetně vytvořeného názvu a jednotlivých úrovní dané kategorie.

Tabulka 4: Datová struktura pro kategorizaci a dlouhé názvy

(Zdroj: Vlastní tvorba)

Materiál	90001234
Závod	1234
Krátký název	Snímač Sick WT18
Dlouhý název	Snímač optický WT18-3P431 1026032 1824 Sick
Název kategorie	Snímače
Úroveň 1	Elektro
Úroveň 2	Snímače
Úroveň 3	Optické
Úroveň 4	Optické
Úroveň 5	Optické
Úroveň 6	Optické
Úroveň 7	Optické

3.1.2 Struktura dat skladových zásob

Skladové zásoby lze zjistit pomocí funkce v ERP systému. Pomocí textového souboru je možno vyexportovat aktuální skladové zásoby. Pro zpracování dat v aplikaci jsou pro nás důležitá čísla materiálu, skladové pozice, počet jednotek skladem, měrná jednotka a typ skladu. Firma má více skladů, každý má svoje typové označení. Proto je při zpracovávání exportovaných dat textového souboru nutné všechny soubory sjednotit a zpracovat do jednoho listu.

3.1.3 Seznam kategorií

Pro objednávkový systém firmy je uchováván soubor s veškerými kategoriemi náhradních dílů. Jedná se o „xlsx“ soubor, ve kterém jsou kategorie uchovávány. I přesto, že pro vyhledavač existuje až sedmi úrovně hierarchie kategorií, tak objednávkový systém pracuje s maximálně 5 úrovněmi. Z důvodu relativně nízké rozmanitosti náhradních dílů je dostačující

používat pětiúrovňovou hierarchii kategorie. Na základě tohoto zjištění, že objednávkový systém pracuje s 5 úrovněmi bylo rozhodnuto, že i v této aplikaci se bude používat pouze 5 úrovní.

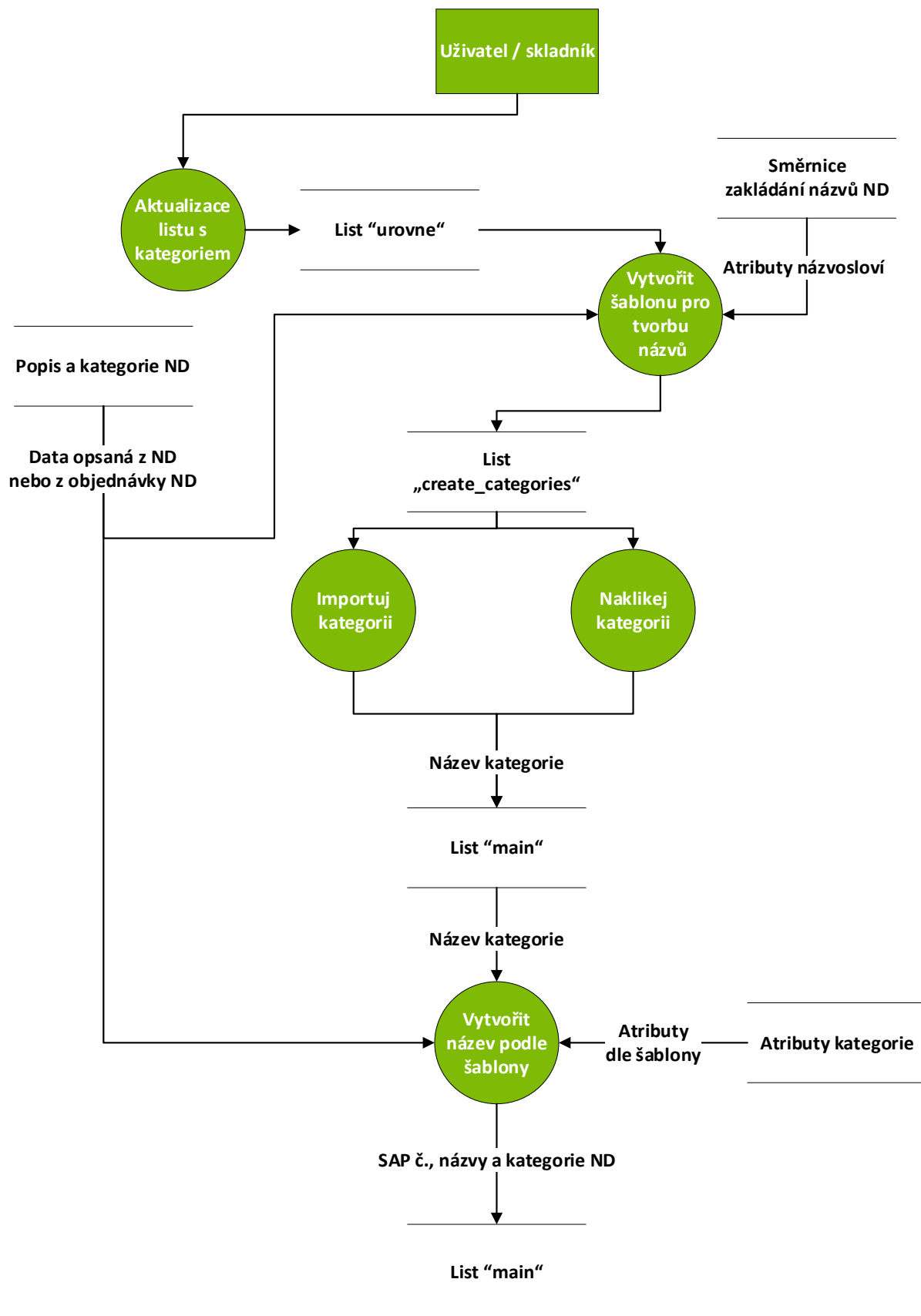
Objednávkový systém umožňuje zadavateli zadat úroveň objednávaného dílu. Uživatel může z tohoto systému vyexportovat „.xlsx“ soubor, ve kterém se nachází materiálové číslo a úroveň kategorií tohoto dílu. Tyto údaje může aplikace zpracovat a uživateli zjednodušit zakládání nových názvů pro vyhledavač. Pokud by zde zadavatel objednávky kategorii nezadal, musí uživatel kategorii zjistit sám a navolit ji pomocí jedné z funkcionalit aplikace.

3.1.4 Fotodokumentace

Všechny soubory fotodokumentace jsou uloženy v jedné složce. Pro zpracování seznamu fotodokumentace je nutné spustit skript v příkazové řádce systému Windows. Tento skript na základě přístupu do určité složky zjišťuje seznam souborů v dané složce a ukládá jej do textového souboru. S tímto souborem se poté pracuje při analýze chybějících fotodokumentací náhradních dílů. Identifikační atribut je název fotky, který musí být totožný s materiálovým číslem, podle kterého se fotka přiřadí k danému dílu.

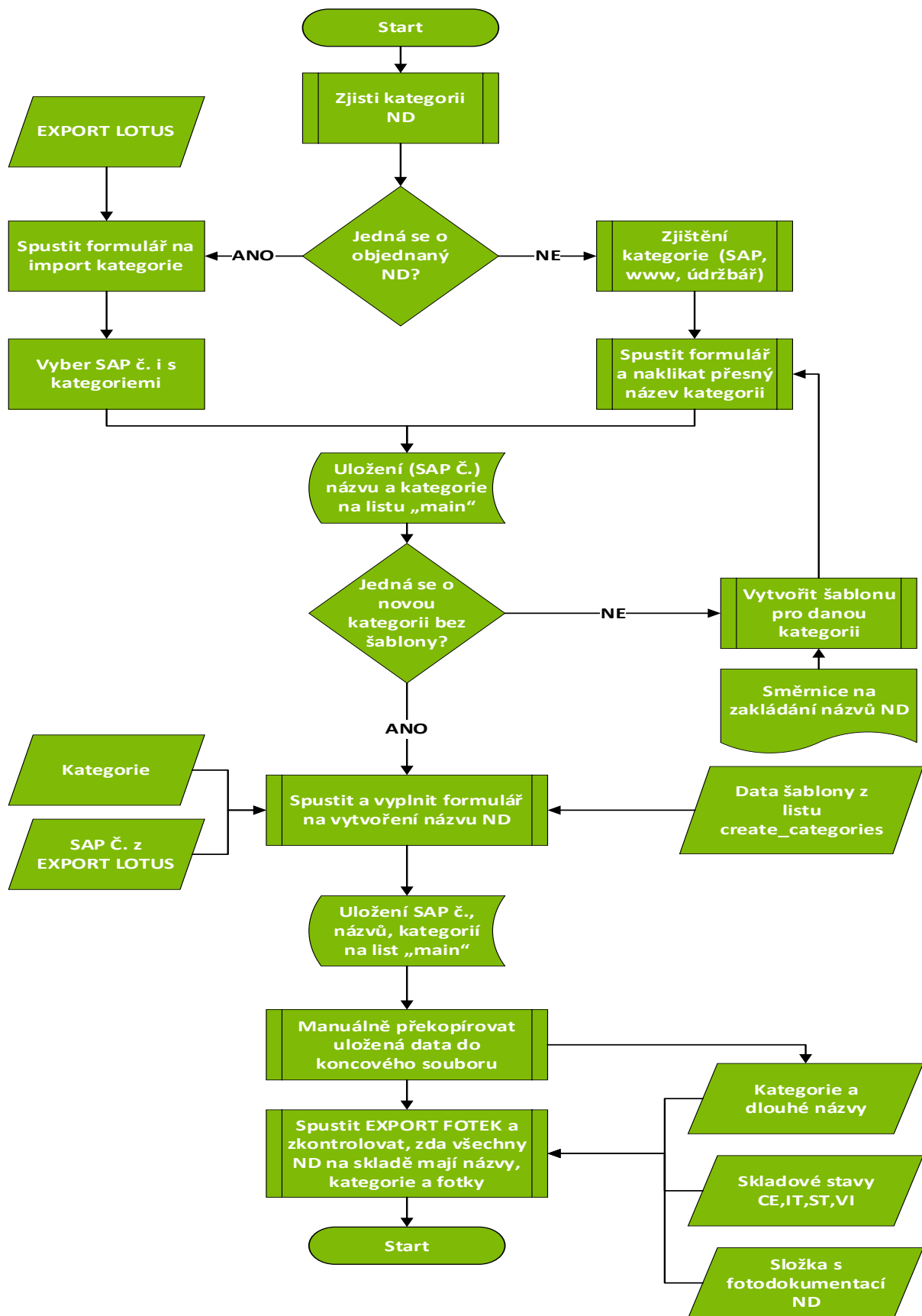
3.2 Funkční modelování

V této kapitole bude znázorněna funkční stránka aplikace. V prvním případě pomocí diagramu toku dat pro tvorbu názvů náhradních dílů pro nově založenou kategorii. Tento případ se týká založení nové kategorie dílů do skladu náhradních dílů, může se jednat například o díly spravované IT oddělením. Dalším diagramem v této kapitole zakresleným je vývojový diagram tvorby nových názvů náhradních dílů. Důraz je zde kladen na zpracování externích dat, využívané pro report obsahující všechna materiálová čísla s příslušnými údaji.



Obrázek 13: Datová flow diagram

(Zdroj: Vlastní tvorba)



Obrázek 14: Vývojový diagram

(Zdroj: Vlastní tvorba)

3.3 Popis aplikace

Vyvíjená aplikace pro zjednodušení pracovních procesů se skládá z několika různých případů užití. Jedná se o příkazová tlačítka, která spustí dané makro. Většinou dojde k otevření formuláře, do kterého uživatel zapisuje data, vybírá z „Option button“ tlačítek, anebo makro importuje a zpracovává externí data.

3.3.1 Hlavní list

Výchozím stavem aplikace je list „main“. Tento list se skládá ze záhlaví, které je podbarveno zeleně. V této oblasti se nachází skupina buněk potřebných ke správnému běhu aplikace. V pravé horní části je skupina ovládacích tlačítek. Jedná se o příkazové tlačítka, pomocí kterých se spouští jednotlivá makra. Makra v této aplikaci zpracovávají data z externích zdrojů, provádějí s nimi operace a generují uživateli tabulární výstup. Zbylá část listu je věnována datům, které uživatel zpracovává při vytváření nového názvu náhradního dílu. Jedná se o tabulku se stejným formátováním a obsahem, jako je tabulka zdrojových dat pro vyhledavač. Každý řádek v této tabulce zaznamenává data k jednomu náhradnímu dílu. Kvůli omezenému přístupu do souboru se zdrojovými daty pro vyhledavač, uživatel vytvořená data do tohoto externího souboru kopíruje ručně. Dalšími základními listy aplikace jsou listy „urovne“ a „create_categories“. Jiné za běhu programu vytvořené listy mají funkci pomocnou při výpočtu nebo zobrazení dat.

The screenshot shows a Microsoft Excel spreadsheet. The top part (rows 1-13) has a green background and contains settings for the application, such as file paths and folder names. The bottom part (rows 14-22) is a table with a header row in orange. The header row has columns: 'Materiál', 'Závod', 'Kód textu materiálu', 'Dílňový název', 'Název kategorie', and seven columns labeled 'Úroveň 1' through 'Úroveň 7'. The data rows below are empty.

Obrázek 15: Aplikace - hlavní list aplikace

(Zdroj: Vlastní tvorba)

Skupina buněk v záhlaví listu je důležitá pro správný běh aplikace. Kompilátor si odsud zjišťuje adresářová cesta k jednotlivým souborům a složkám. K těmto souborům a složkám poté přistupuje a jejich data zpracovává. Dále zde je buňka na vepsání zkratk jednotlivých

skladů, které chceme v aplikaci zpracovávat. Poslední dvojice buněk je určena pro vložení konkrétní kategorie.

Název tohoto souboru:	aplikace.xlsm
Adresář aplikace:	C:\Users\Aleš.DESKTOP-2QRJ8SC\OneDrive - Vysoké učení technické v Brně\Aplikace\aplikace.xlsm
Adresář stromu:	C:\Users\Aleš.DESKTOP-2QRJ8SC\OneDrive - Vysoké učení technické v Brně\Aplikace\urovne_lotus\urovne_lotus.xlsx
Adresář fotek:	C:\Users\Aleš.DESKTOP-2QRJ8SC\OneDrive - Vysoké učení technické v Brně\Aplikace\fotky
Adresář stavby skladu:	C:\Users\Aleš.DESKTOP-2QRJ8SC\OneDrive - Vysoké učení technické v Brně\Aplikace\stavy_skladu
Adresář kategorie a dlouhé názvy:	C:\Users\Aleš.DESKTOP-2QRJ8SC\OneDrive - Vysoké učení technické v Brně\Aplikace\kategorie
Adresář Desktop	C:\Users\Aleš.DESKTOP-2QRJ8SC\Desktop
Adresář lotus export	C:\Users\Aleš.DESKTOP-2QRJ8SC\OneDrive - Vysoké učení technické v Brně\Aplikace\export_lotus\EXPORT_LOTUS.xlsx
Sklady náhradních dílů	CE IT VI ST
Překopírovaná kategorie z LOTUSU:	
Naklikaná kategorie:	Elektro - Elektro - Snímače - Optické

Obrázek 16: Aplikace - konfigurační buňky aplikace

(Zdroj: Vlastní tvorba)



Obrázek 17: Aplikace - Ovládací tlačítka aplikace

(Zdroj: Vlastní tvorba)

3.3.2 Aktualizování listu s kategoriemi

Pro správné fungování aplikace je zapotřebí mít soubor propojený s externím souborem z objednávkového systému, který určuje veškeré ve skladu existující kategorie. Pomocí VBA skriptu je naimportován seznam všech existujících kategorií do listu se jménem „urovne“. Seznam kategorií je vstupem pro vytváření šablon. Podle těchto šablon jsou vytvářeny názvy konkrétních náhradních dílů. Na obrázku máme příklad struktury dat na listu „urovne“, kde jsou v jednotlivých sloupcích uvedeny konkrétní kategorie.

V tomto případě se jedná o seznam všech možných druhů snímačů. Mezi kategoriemi druhé a třetí úrovně se nachází čísla, která jsou pro tuto aplikaci nepodstatná. Pokud již není určena další podkategorie, tak je sloupeček vyplněn znakem „-“. V posledním sloupečku je pak název kategorie. Tento název je vytvořen spojením názvů všech úrovní dané kategorie. Pokud kategorie obsahuje méně jak pět úrovní, tak je název ukončen posledním znakem v názvu poslední dostupné úrovní.

Elektro	Snímače	2060205	Indukční	-	-	Elektro - Snímače - Indukční
Elektro	Snímače	2060205	Optické	-	-	Elektro - Snímače - Optické
Elektro	Snímače	2060205	Bezpečnostní	-	-	Elektro - Snímače - Bezpečnostní
Elektro	Snímače	2060205	Magnetické	-	-	Elektro - Snímače - Magnetické
Elektro	Snímače	2060205	Magnetické na píсты	-	-	Elektro - Snímače - Magnetické na píсты
Elektro	Snímače	2060205	Ostatní	-	-	Elektro - Snímače - Ostatní
Elektro	Snímače	2060205	Rotační	-	-	Elektro - Snímače - Rotační
Elektro	Snímače	2060205	Teplotní	-	-	Elektro - Snímače - Teplotní
Elektro	Snímače	2060205	Teplotní a vlhkostní	-	-	Elektro - Snímače - Teplotní a vlhkostní
Elektro	Snímače	2060205	Tlakové	-	-	Elektro - Snímače - Tlakové
Elektro	Snímače	2060205	Ultrazvukové	-	-	Elektro - Snímače - Ultrazvukové
Elektro	Snímače	2060205	Příslušenství	Kabely	-	Elektro - Snímače - Příslušenství - Kabely
Elektro	Snímače	2060205	Příslušenství	Ostatní	-	Elektro - Snímače - Příslušenství - Ostatní

Obrázek 18: Příklad kategorií snímačů

(Zdroj: Vlastní tvorba)

3.3.3 Vytvoření nové šablony kategorie

Pro vytvoření šablony dané kategorie je naprogramováno makro spustitelné pomocí tlačítka „Vytvořit šablonu kategorie“. Po kliknutí na příkazové tlačítko se spustí uživatelský formulář, se základními sedmi textovými poli. První krok spočívá v identifikaci kategorie, kterou chce uživatel založit. K tomuto je určené malé textové pole, kde uživatel zadá číslo řádku z listu „urovne“, na kterém se nachází konkrétní kategorie. Druhou možností je vybrání kategorie pomocí rozevratelné nabídky „ComboBox“. Do textového pole vedle čísla řádku aplikace vepíše informaci, zda je šablona pro konkrétní kategorii již vytvořená nebo ji je možno

založit. Do pole „Kategorie s ID“ aplikace vepíše číslo ID, které je unikátní v rámci seznamu založených šablon. Poté uživatel dle směrnice pro tvorbu názvosloví vepíše číslo udávající počet zadávaných parametrů pro danou kategorii. Obvykle se jedná o atributy technické specifikace dané kategorie. Dále jsou zde textová pole pro vyplnění podstatného jména a přidavných jmen, která se budou nacházet na začátku každého dílu patřícího do této kategorie. V levém spodním rohu je skupina příkazových tlačítek. Tlačítko „Načíst další parametry kategorie“ načítá skupinu textových polí s popisky. Počet textových polí je roven dvojnásobku, který uživatel zadal v textovém poli s popiskem „Zadejte počet atributů dané kategorie“. Ke každému atributu je zapotřebí mít možnost zadat nápovědu, proto bude vytvořen dvojnásobný počet textových polí s příslušnými nadpisy.

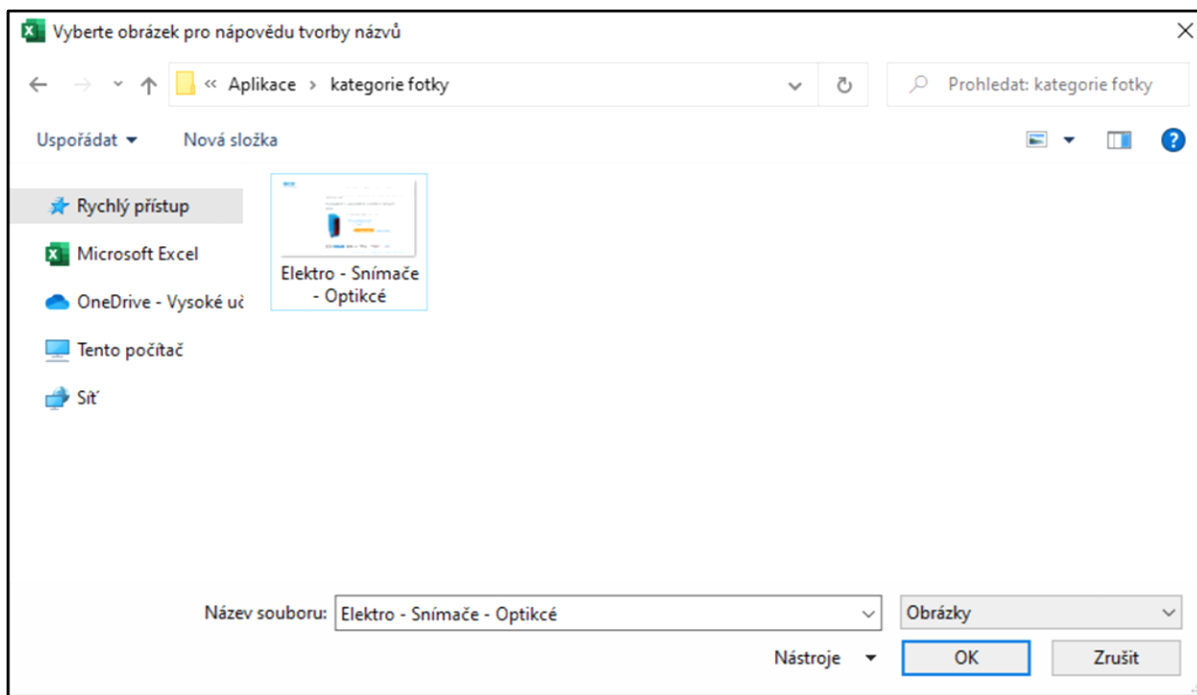
The screenshot shows a window titled "Tvorba kategorie" with a light blue background. The form contains the following elements:

- A label "Zadejte číslo řádku, na kterém je kategorie" followed by a text input field.
- A label "Vytváříme novou šablonu pro kategorii:" followed by a dropdown menu.
- A label "Zadejte počet parametrů kategorie navazujících na přidavné jména" followed by a text input field.
- A label "Kategorie s ID" followed by a text input field.
- A label "Podstatné jméno" followed by a text input field.
- A label "Přidavné jméno první podkategorie V případě vynechání nic nevyplňujte" followed by a text input field.
- At the bottom, three buttons: "DALŠÍ (Načíst počet zadávaných parametrů kategorie)", "Přidat další soubor", and "Uložit novou šablonu kategorie".

Obrázek 19: Aplikace - vytvoření šablony kategorie - výchozí okno

(Zdroj: Vlastní tvorba)

Následně uživateli umožní vybrat adresářovou cestu s „.jpg“ souborem, jenž by měl znázorňovat nebo specifikovat danou kategorii. V případě potřeby lze přidávat až další čtyři „.jpg“ soubory. Podporované jsou pouze soubory typu „.jpg“.



Obrázek 20: Aplikace - vložení fotky

(Zdroj: Vlastní tvorba)

Po načtení skupiny ovládacích prvků uživatel vyplní daná textová pole. Pro každý atribut jsou vytvořena textová pole pro popis atributu a nápovědu. Nápověda by měla napovídat, jak daný atribut zjistit, jakou má fyzikální jednotku nebo z jaké informace ho vyčíst.. V nápovědě lze zadat například nejčastější hodnoty napětí pro baterie, což při tvorbě názvu uživateli usnadní identifikaci daného parametru na konkrétním dílu. Poslední textová pole uchovávají adresářovou cestu k fotodokumentaci pro danou šablonu. V posledním kroku je nutné kliknout

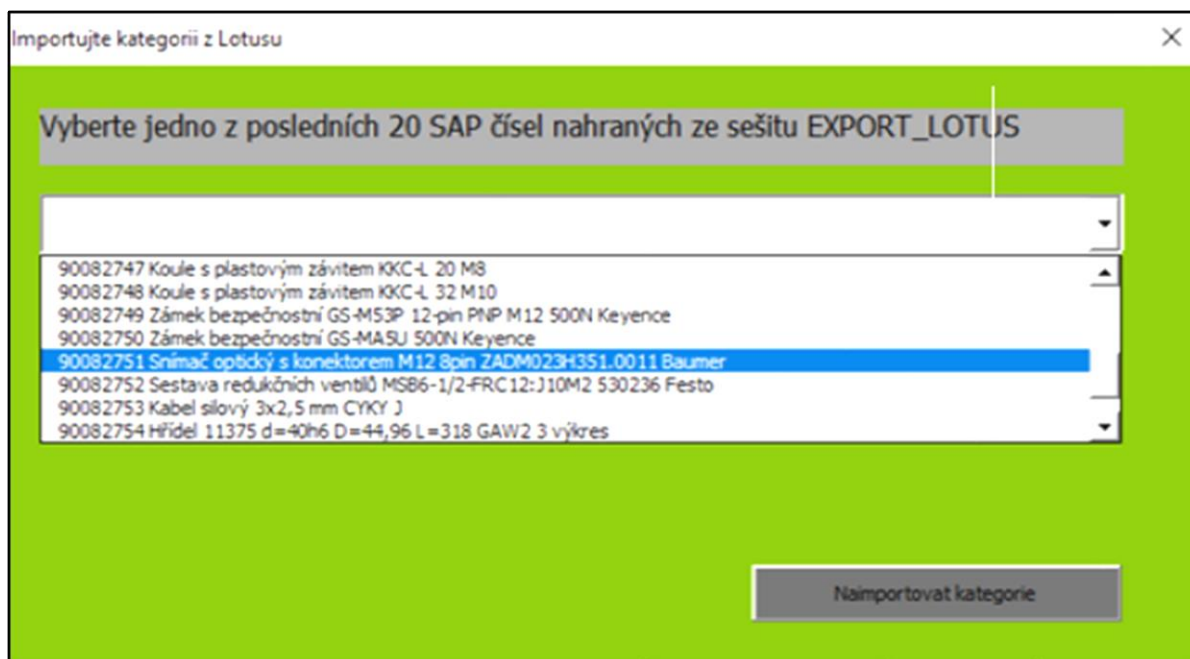
na tlačítko „Uložit šablonu nové kategorie“. Poté dojde k zapsání všech potřebných údajů šablony do listu „create_categories“.

Obrázek 21: Aplikace - vytvoření šablony kategorie - příklad

(Zdroj: Vlastní tvorba)

3.3.4 Importování kategorie z LOTUS EXPORT

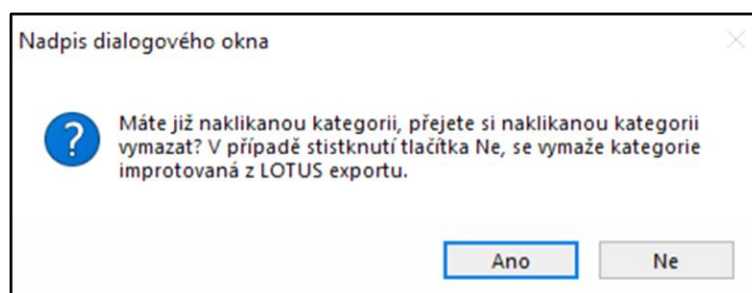
Před vytvořením názvu podle šablony, je nutné importovat do formuláře tvorby názvu název kategorie. Prvním nástrojem pro import kategorie je makro importující posledních dvacet záznamů ze souboru exportovaného z objednávkového systému. Makro načte formulář, ve kterém uživatel vybere daný díl. Výběr probíhá z rozevíracího seznamu „ComboBox“, kde jsou načteny spojená materiálová čísla s názvy náhradních dílů. Po výběru daného dílu uživatel potvrdí svoji volbu příkazovým tlačítkem „Naimportovat kategorii“. Výsledkem je zapsání dané kategorie do pomocné buňky na hlavním listu „main“.



Obrázek 22: Aplikace - Import kategorie z objednávkového systému

(Zdroj: Vlastní tvorba)

Pokud je již kategorie naimportována z formuláře pro ruční navolení dané kategorie, tak aplikace nabídne uživateli, která kategorie se má zachovat, a která vymazat.



Obrázek 23: Aplikace - import kategorií zvolení kategorie

(Zdroj: Vlastní tvorba)

3.3.5 Navolení kategorie

Druhou možností importu kategorie je ručně navolení kategorie. K tomuto účelu je uzpůsobeno makro s dynamickou tvorbou ovládacích prvků ve formuláři. Vstupní data jsou

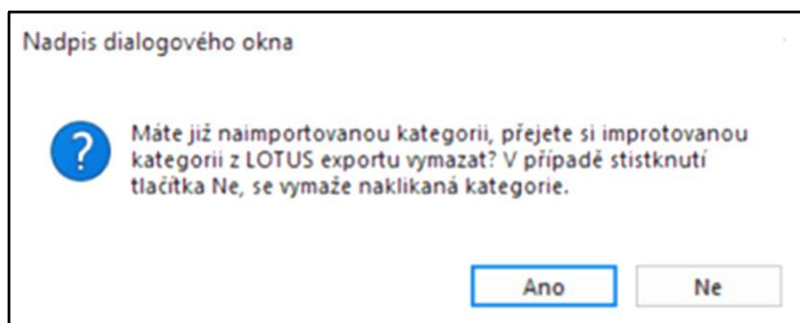
brána z listu „urovne“. Uživatel zatrhuje postupně všechny úrovně od první až po pátou. Při ukládání aplikace kontroluje, zda již není naimportována kategorie z objednávkového systému.



Obrázek 24: Aplikace - formulář na zvolení kategorie

(Zdroj: Vlastní tvorba)

Pokud je kategorie z objednávkového systému naimportována, tak aplikace nabídne uživateli, která kategorie se má zachovat, a která vymazat. Výsledkem je zápis názvu kategorie do dané buňky na listu „main“.



Obrázek 25: Aplikace - zvolení kategorie

(Zdroj: Vlastní tvorba)

```
'first occurrence of the specific first category
'last occurrence of the specific last category
'row_count - number of nonempty cell in column "A" for first_cat
Erase Ar_2()
Worksheets("urovne").Activate
row_count = Range("A:A").Cells.SpecialCells(xlCellTypeConstants).Count
first_occurrence = first_row(ctlOptionButton_1.Caption, row_count, "A", 5)
last_occurrence = last_row(ctlOptionButton_1.Caption, first_occurrence, row_count, "A")

'Load all unique values text to Array
subtr = 0
subtr = last_occurrence - first_occurrence + 1
ReDim Ar_2(subtr)
Ar_2() = create_list(first_occurrence, last_occurrence, "B") ',5

'creates option buttons second_cat
Call creat_opt_btn(Ar_2, "second_cat", 160, RGB(21, 113, 75))
```

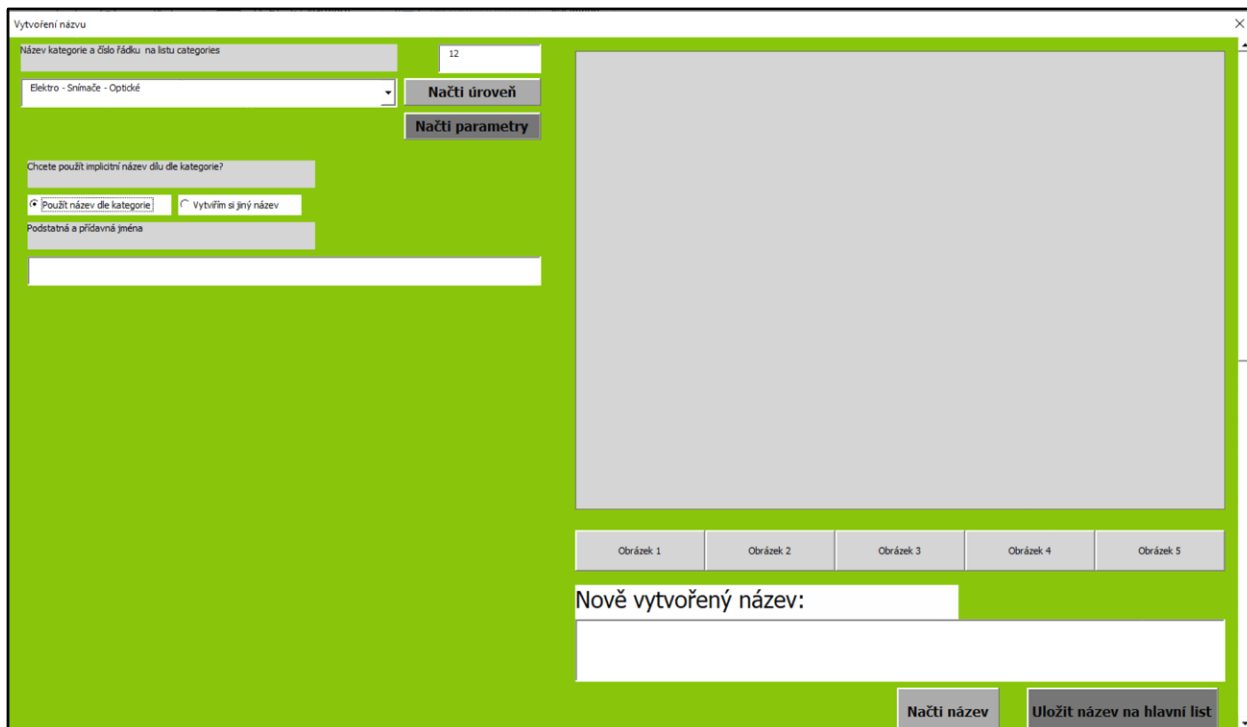
Obrázek 26: Aplikace - ukázka kódu pro vytvoření prvku „OptionButton“ první úrovně kategorie

(Zdroj: Vlastní tvorba)

3.3.6 Tvorba názvu pro vyhledavač

Pomocí tlačítka na spuštění makra pro vytvoření názvu náhradního dílu se spustí napovídající formulář. Tento formulář je navrhnut tak, aby byl pro uživatele co nejvíce intuitivní, přehledný a lehce pochopitelný pro méně zkušené uživatele. V prvním kroku je za pomoci tlačítka „Načti kategorii“ načtena kategorie. Tento název kategorie byl automatizovaně vytvořen již v předešlých krocích za pomoci jiných maker a zapsán do příslušné buňky na listu „main“. V dalším kroku uživatel klikne na tlačítko „Načti parametry“. Zde se formulář spojí s již vytvořenou šablonou pro tvorbu názvu vybrané kategorie, dojde k vytvoření příslušných formulářových objektů pro zadávání parametrů uživatelem. Pokud šablona obsahuje

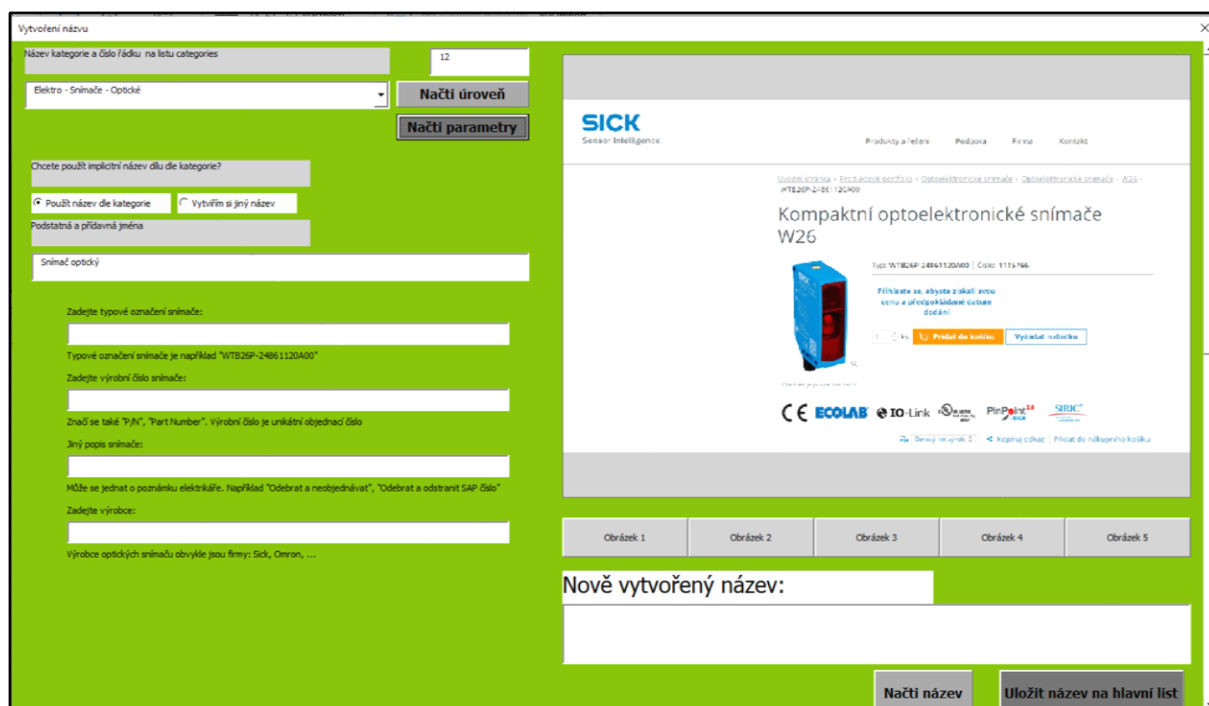
fotodokumentaci, zobrazí se zde i fotky. Nejvyšší možný počet fotek zobrazovaných v rámci jedné kategorie je 5 fotek.



Obrázek 27: Aplikace - výchozí okno tvorby názvu

(Zdroj: Vlastní tvorba)

V posledním kroku uživatel načte název po moci tlačítka „Načti název“. Název se propíše do příslušného textového pole tohoto formuláře. Po kontrole názvu je název uložen do tabulky na listu „main“.



Obrázek 28: Aplikace - načtené okno tvorby názvu

(Zdroj: Vlastní tvorba)

3.3.7 Report dat

Pro zpracování informací o náhradních dílech může uživatel použít zautomatizované makro. Toto makro sjednocuje skladové zásoby s pozicemi ze všech skladů do jednoho listu. Z externího souboru naimportuje názvy a kategorie. Vygeneruje a spustí soubor se skriptem v příkazové řádce Windows. Tento skript reportuje seznamu všech fotek v daném adresáři. Tento seznam je uložen ve formátu textového souboru. Každá fotodokumentace k materiálovému číslu musí mít název nesoucí materiálové číslo a koncovku „.jpg“ souboru. Vytvořený textový soubor potom nahraje do pomocného listu a spojením s příslušným listem za pomoci funkce „SVYHLEDAT“ zapíše do příslušného listu, zda fotodokumentace existuje nebo ne. Pomocí této funkce dojde k ověření, zda se materiálové číslo nachází v zdrojovém souboru pro vyhledavač a zda obsahuje i vytvořený název nebo ne. Pomocí této funkce dojde k

ověření, zda se materiálové číslo nachází v zdrojovém souboru pro vyhledavač a zda obsahuje i vytvořený název.

SAP číslo	Logio název	FOTO	Požice	Sklad	Chyba	Počet	Jednotka
90002616	Snímač opt... Sick	FOTO NENALEZENO	T-02-04-02	CE1	CHYBA SAP číslo nemá dlouhý název v souboru: C:\Users\	1	KUS
90006319	Snímač opt...2 Sick	OK	T-02-04-02	CE1	CHYBA SAP číslo se nenachází v souboru: C:\Users\xxxxyy\	4	KUS

Obrázek 29: Aplikace - chybové hlášky reportu dat

(Zdroj: Vlastní tvorba)

```
'Copy data from all inventory files from all subwarehouses to sheet "photo"
'dt_sfx - date format for substrning in inventory txt filename
'arr_u - type of subwarehouses
'Range("D5").Value - cell with path name of inventory files

Function upload_data(dt_sfx As String, arr_u() As String)
    Dim first_empty_row, count_copy_rows, count_empty_copy, y, x As Integer
    Dim copy_range, copy_range_to, inv_name As String
    Dim wb_name, wb_in As Workbook
    Dim s As Variant
    Const xlToUp As Long = -4162
    y = 10

    For Each s In arr_u()
        inv_name = ThisWorkbook.Sheets("main").Range("D5").Value & "\Stavy_Skladu_" & s & "_" & dt_sfx & ".txt"
        ThisWorkbook.Sheets("photo").Activate
        first_empty_row = first_empty_cell("A")
        Set wb_in = ThisWorkbook
        wb_name = 0

        'Open file
        On Error Resume Next
        Set wb_name = Workbooks.Open(inv_name)
        'copy data
        wb_name.Sheets(1).Activate
        count_copy_rows = first_empty_cell("E")
        copy_range = "E7" & ":" & "E" & count_copy_rows - 1
        wb_name.Sheets(1).Range(copy_range).Copy
        count_empty_copy = first_empty_row + count_copy_rows - 1
        copy_range_to = "A" & first_empty_row & ":" & "A" & count_empty_copy
        wb_in.Sheets("photo").Range(copy_range_to).Insert

        'Error MessageBox if file does not exists
        If wb_name Is Nothing Then
            ThisWorkbook.Sheets("photo").Cells(10, y) = "CHYBA v souboru - " & s
            MsgBox ("Problém s otevřením souboru urovne_lotus. Aplikace se snaží otevřít soubor: " & vbCrLf & inv_name & vbCrLf _
                & vbCrLf & "Zkontrolujte zda je soubor správně zadán nebo zda se ve složce nachází.")
        Else
            ThisWorkbook.Sheets("photo").Cells(10, y).Locked = True
        End If

        y = y + 1

        'Close file
        wb_name.Close SaveChanges:=False
    Next
End Function
```

Obrázek 30: Aplikace - Zjednodušená verze funkce pro kopírování dat z inventurních souborů

(Zdroj: Vlastní tvorba)

3.3.8 Vymazat nepoužívané listy

Méně podstatným, ale užitečným makrem je makro, které uvede sešit do původního stavu. Makro projde všechny vytvořené listy a vymaže všechny kromě třech hlavních. Těmito hlavními listy jsou: „main“, „urovne“ a „create_categories“.

3.4 Zhodnocení práce

V následující kapitole bude zhodnocen vliv této aplikace na práci ve skladu. Nejprve si shrneme předchozí řešení a implementaci aplikace, poté ekonomické zhodnocení a přínosy.

3.4.1 Předchozí řešení a zavádění nové aplikace

Předchozí řešení tohoto problému bylo založeno na zejména vědomostech uživatele nebo na manuálním vyhledávání pravidel pro zakládání názvů ve směrnici popisující tyto pravidla. Pro nové uživatele nebo nové hardwarové vybavení je aplikace uzpůsobena na rychlé nastavení uživatelských parametrů. To se provádí vyplněním základní skupiny buněk na listu „main“. Pro nasazení aplikace do provozu je zde důležité, že zaměstnanci skladu jsou již podobné aplikace vytvořené ve VBA používají. Velkou výhodou pro údržbu aplikace je i faktor, že na oddělení IT je specialista, který vytvářel již zmiňované aplikace vytvořené ve VBA. Důraz při implementaci byl kladen na kvalitní dokumentaci kódu, aby byla co nejjednodušší jeho údržba a rozvoj o nové funkčnosti.

3.5 Ekonomické zhodnocení

Vývoj aplikace probíhal v prostředí Microsoft Excel v jazyku VBA. Jelikož kancelářský balíček Office 365 je v této firmě přístupný každému zaměstnanci, tak náklady na softwarové vybavení k tomuto projektu lze považovat za nulové. V projektu budeme jako programátora uvažovat, zaměstnanec, který má na starosti podporu aplikací a specializuje se na tvorbu aplikací ve VBA má hodinovou mzdu 260,- Kč. Jedná se o hrubou hodinovou mzdu. Zaměstnavatel musí za zaměstnance odvést pojištění. Celková náklady na jednu hodinu práce programátora jsou tedy po zaokrouhlení 348,- Kč. Práce programátora lze rozdělit na tři části. První částí je zjišťování požadavků na aplikaci a její návrh. Zjišťování požadavků prováděl na základě sledování pracovních procesů skladníků a diskuse s nimi. Následovně zjištěné požadavky konzultoval s manažerem skladu. Na základě zjištění vytvořil návrh a prvotní prototyp aplikace. Tato část programátorovi zabrala 10 hodin. Po akceptaci návrhu a prototypu

následovala etapa vývoje aplikace. Na vývoj aplikace bylo nutno vyčlenit 40 hodin. Nejméně časově náročnou fází bylo zprovoznění aplikace na všech počítačích a účtech a školení zaměstnanců skladu. Tato fáze zabrala 8 hodin. Z důvodu absence důkladného testování aplikace je nutno počítat s údržbou a opravou chyb, které se neodhalily při vývoji, testování a uvedení do provozu. Na tento úkon bude nutno po dobu 9 měsíců vyčlenit programátorovi každý měsíc 2 hodiny práce na projektu.

Tabulka 5: Ekonomické zhodnocení

(Zdroj: Vlastní tvorba)

Úkon	Cena
Specifikace požadavků a návrh (hrubá mzda)	2 600 Kč
Implementace aplikace	10 400 Kč
Zprovoznění aplikace a školení	2 080 Kč
Údržba a oprava chyb v průběhu 9 měsíců	4 680 Kč
Celkové pojištění odvedené zaměstnavatelem	6 679 Kč
Celkové náklady	26 439 Kč

Celková částka vynaložená na tento projekt provedený programátorem by činila 26 439,- Kč. Tento program byl vyhotoven autorem této práce v rámci dohody DPP a ve volném čase.

3.6 Přínosy práce

Hlavními přínosy aplikace byly očekávány rychlejší tvorba názvů a konzistentnost dat v databázi. Konzistentnost dat je důležitá při vyhledávání náhradních dílů. Formulář, který funguje jako šablona uživateli napovídá, jaké atributy, a v jakém pořadí je nutné atributy zadávat. Po zadání všech dat, je využita funkce „Trim“, která z řetězce vymaže nadbytečné mezery, tabulátory a ostatní bílé znaky. Pro správné zvolení kategorie je naprogramován formulář, ve kterém si uživatel příslušnou kategorii zvolí. Aplikace eliminuje uživateli zadat chybně sestavený nebo neúplný název kategorie. Může se jednat o příklad, kdy uživatel nemá dostatek znalostí. Na základě chybějících znalostí uživatele může být špatně zvolena kategorie

„Elektro - Snímač – Ostatní“, namísto správné kategorie „Elektro - Snímač – Indukční“. Uživatel v příslušném formuláři vidí přehledně všechny podúrovně kategorie Snímačů, čímž by mělo dojít k eliminaci chybovosti v názvech kategorií. Zjišťování seznamu všech souborů ve složce obsahující fotodokumentaci byl náročný proces, kdy se spouštěl manuálně skript z příkazové řádky. Tento proces vyžadoval pokročilou uživatelskou znalost systému Windows. Poté uživatel musel manuálně data upravovat, aby se na listu vyskytovaly pouze data, která chce a nepotřebná data vymazat. Toto makro zpřístupní uživateli seznam materiálových čísel, s příslušnými názvy, skladovými zásobami, pozicemi ve skladu, informacemi o existenci fotodokumentace a informacemi o neúplnosti dat. Tento seznam obsahuje jen materiály, které jsou skladem. Pouze materiály skladem lze s téměř stoprocentní jistotou identifikovat, upravit názvy a pořídit kvalitnější fotodokumentaci. Přínos tohoto makra je menší náročnost na uživatelskou znalost počítačů a automatická úprava dat. Nejvíce znatelným přínosem je úspora času, kdy se změnila časová náročnost tohoto procesu na zlomek původní časové náročnosti.

3.7 Vývoj aplikace do budoucna

Aplikace v této verzi nedosahuje svého plného potenciálu. Některé funkce by do aplikace mohly být ještě implementovány dodatečně. Firmě lze doporučit dvě možnosti na budoucí vývoj aplikace. Tyto možnosti se od sebe odlišují zásadně rozdílným odhadem na rozpočet. Jedním je zdokonalovat nynější aplikaci a zachovat ji ve VBA a prostředí MS Excel. Výrazně dražší možností by bylo rozšíření vyhledavače a doplnění již existujících nebo dalších navrhovaných funkcí. Výhodou by bylo přesunutí se do prostředí online a jednotnost systému. Uživatel by měl vše na jednom místě ve webové aplikaci. Další možností by mohlo být vytvoření samostatného SAP modulu s novými funkcemi. I toto řešení nového SAP modulu by bylo oproti aktualizování nynější verze aplikace ve VBA nákladné.

Vzhledem k již existující aplikaci budeme předpokládat upgrade nynější aplikace vytvořené ve VBA. První nabízenou novou funkcí by mohlo být přidání objektu „CheckBox“ do formuláře pro tvorbu šablony nové kategorie. Toto políčko by mělo za funkci vytvoření textového pole, do kterého by se dal zapsat předem definovaný text důležitý pro danou kategorii. V situacích, kdy by se do názvu zapisoval rozměr, konkrétně délka, tak by se do textového pole při tvorbě šablony vložil řetězec „L=“. Tento řetězec s příslušným názvem parametru a nápovědou by se uživateli zobrazil i při zakládání názvu. Změna by byla v tom, že

by uživatel nemusel zapisovat ručně do textového pole řetězec „L=“. To by zrychlilo proces tvorby názvů a přispělo k snížení pravděpodobnosti, že se uživatel přepíše v názvu.

Vzhledem k již existující aplikaci VBA, v které dochází k automatickému zakládání nových materiálových čísel se zde nabízí možnost integrovat tyto dvě aplikace do sebe. Zde by ovšem muselo být vše řádně navrženo a zabezpečeno. Obávám se, že integrace by mohla způsobit menší přehlednost již existující aplikace pro zakládání materiálových čísel.

Práci by šlo ulehčit i za pomoci automatické konverze „.png“ nebo „.pdf“ souborů na soubor „.jpg“. Tato konverze by byla efektivní v případě, kdy by byl vytvořen snímek obrazovky, který je ve Windows ukládán do souboru „.png“. Tento soubor poté musí skladník manuálně konvertovat na „.jpg“ soubor. Tato konverze je nutná pro správné zobrazení fotodokumentace dané kategorie ve formuláři. Proces konverze formátu fotek by se mohl uskutečnit za předpokladu, že by ve firemních zařízeních byl příslušný program, který by umožnil pomocí příkazu v příkazové řádce formát fotodokumentace konvertovat.

U makra vytvářející report náhradních dílů by se nabízelo zdokonalit toto makro o filtrování skladů nebo regálů v rámci jednoho skladu. Příslušný filtr by uživatel definoval pomocí uživatelského formuláře. V tomto formuláři by měl rozevírací seznamy. V jednom seznamu by si vybíral sklad a v druhém konkrétní regály daného skladu.

Závěr

Cílem této práce bylo navrhnutí aplikace v prostředí MS Excel, která by zautomatizovala procesy ve skladu náhradních dílů ve společnosti HARTMANN-RICO a.s.. Hlavním bodem celého procesu implementace aplikace bylo porozumění tvorby názvů těchto dílů. Za cíl bylo implementovat aplikaci, která zjednodušila proces vytváření názvů materiálových čísel. Další požadovanou vlastností bylo, aby tyto názvy byly v dané kategorii jednoznačně určeny, byly konzistentní, úplné a aktuální.

V první části práce byl popsán program Microsoft Excel a jazyk VBA, ve kterém byla aplikace vyvíjena. Pro metody zkoumání informačního systému a procesů ve skladu náhradních dílů byly využity SWOT analýza a metoda HOS 8. Pomocí těchto analýz byly definované silné a slabé stránky, příležitosti a hrozby firmy. Tato analýza byla prováděna na pracovních procesech ve skladě náhradních dílů. Metodou HOS 8 byl zkoumán informační systém a jeho okolí.

V kapitole vlastního návrhu řešení bylo popsáno datové, funkční modelování aplikace a její vývoj. Cílem aplikace bylo zefektivnění tvorby názvů náhradních dílů, jejich konzistentnost a úplnost. Formulář pro tvorbu konkrétního názvu tyto kritéria splnil. Na základě konzistentnosti a úplnosti názvů náhradních dílů bude možno analyzovat, jestli se ve skladu nachází nějaké duplicitní materiálové čísla. Tato aplikace dokázala vytvořit šablony, pomocí které uživatelé mohou vytvářet názvy se všemi potřebnými parametry. Správné a úplné názvy náhradních dílů v budoucnu pomohou nákupčí k efektivnějšímu nákupu dílů. Díly bude možno kupovat přímo od výrobce za levnější cenu, než by tomu mohlo být při koupi přes dodavatele. V závěru kapitoly je zhodnocena ekonomická stránka projektu a popsány případné návrhy na doplnění dalších funkcionalit aplikace.

Seznam použitých zdrojů

- [1] HARTMANN | HARTMANN Česká republika [online]. Veverská Bítýška: HARTMANN – RICO, 2023 [cit. 2023-01-31]. Dostupné z: <https://www.hartmann.info/cs-cz/>
- [2] Zdravotnické potřeby...LékarnaHARTMANN [online]. Veverská Bítýška: HARTMANN – RICO [cit. 2023-01-31]. Dostupné z: <https://www.lekarnahartmann.cz/>
- [3] URTIS, Tom. Excel VBA 24-Hour Trainer [online]. 2nd edition. Indianapolis: John Wiley & Sons, 2015 [cit. 2023-04-30]. ISBN 978-1-118-99137-4. Dostupné z: http://club.futureinhands.com/assets/books/Excel_VBA_24-Hour_Trainer_2nd_Edition.pdf
- [4] ALEXANDER, Michael a John WALKENBACH. Excel VBA Programming For Dummies. 5th edition. Hoboken: John Wiley & Sons, 2019. ISBN 978-1-119-51823-5.
- [5] ALEXANDER, Michael a Dick Kusleika KUSLEIKA. Excel 2019 Power Programming with VBA. Indianapolis: John Wiley & Sons, 2019. ISBN 978-1-119-51492-3.
- [6] AITKEN, Peter G. Excel Programming Weekend Crash Course. New York: Wiley Publishing, 2003. ISBN 0-7645-4062-9.
- [7] KRÁL, Martin. Excel VBA: výukový kurz. Brno: Computer Press, 2010. ISBN 978-80-251-2358-4.
- [8] Overview of forms, Form controls, and ActiveX controls on a worksheet. Microsoft - Cloud, Computers, Apps & Gaming [online]. Washington: Microsoft, c2021 [cit. 2023-05-15]. Dostupné z: <https://support.microsoft.com/en-us/office/overview-of-forms-form-controls-and-activex-controls-on-a-worksheet-15ba7e28-8d7f-42ab-9470-ffb9ab94e7c2>
- [9] Microsoft EXCEL ... ať pracuje za Vás ... [online]. c2004-2021 [cit. 2023-04-30]. Dostupné z: <https://office.lasakovi.com/excel/>
- [10] Kočí, R., Křena, B.: Úvod do softwarového inženýrství. Studijní opora, VUT v Brně, 2010
- [11] Vývojový diagram (Flow chart). ManagementMania.com [online]. 2017 [cit. 2023-04-30]. Dostupné z: <https://managementmania.com/cs/vyvojovy-diagram-flow-chart>
- [12] KOCH, Miloš a Bernard NEUWIRTH. Datové a funkční modelování. Vyd. 4., rozš. Brno: Akademické nakladatelství CERM, 2010. ISBN 978-80-214-4125-5.
- [13] KOCH, Miloš. Posouzení efektivnosti informačního systému metodou HOS. Trendy ekonomiky a managementu [online]. Vysoké učení technické v Brně, Fakulta podnikatelská, 2013, (16), 49-56 [cit. 2020-04-19]. ISSN 1802-8527. Dostupné z: <http://hdl.handle.net/11012/24462>
- [14] PETRYL, Jan. SWOT analýza. Marketing Mind [online]. České Budějovice: Marketing Mind, 2017 [cit. 2023-04-30]. Dostupné z: <https://www.marketingmind.cz/swot-analyza/>
- [15] SWOT analýza. ManagementMania.com [online]. 2020 [cit. 2023-04-30]. Dostupné z: <https://managementmania.com/cs/swot-analyza>
- [16] MARVAN, Luboš. Lekce 1 - Úvod do VBA. Itnetwork.cz Učíme národ IT [online]. Praha: itnetwork.cz Učíme národ IT, 2023 [cit. 2023-04-30]. Dostupné z: <https://www.itnetwork.cz/ms-office/microsoft-vba-zaklady/uvod-do-vba>

Seznam použitých obrázků

Obrázek 1: Panel vývojář.....	13
Obrázek 2: Zabezpečení maker	15
Obrázek 3: MS Excel editor VBA.....	17
Obrázek 4: Konstrukce If...Then...Elsef...End If	21
Obrázek 5: Konstrukce Select Case	21
Obrázek 6: Cyklus For Next	22
Obrázek 7: Cyklus For Each Next.....	22
Obrázek 8: Cyklus Do While	23
Obrázek 9: Uživatelský formulář.....	25
Obrázek 10: VBA Debugging.....	27
Obrázek 11: Diagram HOS 8	37
Obrázek 12: SWOT analýza.....	40
Obrázek 13: Datová flow diagram	45
Obrázek 14: Vývojový diagram	46
Obrázek 15: Aplikace - hlavní list aplikace.....	47
Obrázek 16: Aplikace - konfigurační buňky aplikace.....	48
Obrázek 17: Aplikace - Ovládací tlačítka aplikace	48
Obrázek 18: Příklad kategorií snímačů	49
Obrázek 19: Aplikace - vytvoření šablony kategorie - výchozí okno.....	50
Obrázek 20: Aplikace - vložení fotky.....	51
Obrázek 21: Aplikace - vytvoření šablony kategorie - příklad.....	52
Obrázek 22: Aplikace - Import kategorie z objednávkového systému	53
Obrázek 23: Aplikace - import kategorií zvolení kategorie	53
Obrázek 24: Aplikace - formulář na zvolení kategorie	54
Obrázek 25: Aplikace - zvolení kategorie.....	55
Obrázek 26: Aplikace - ukázka kódu pro vytvoření prvku „OptionButton“ první úrovně kategorie.....	55
Obrázek 27: Aplikace - výchozí okno tvorby názvu	56
Obrázek 28: Aplikace - načtené okno tvorby názvu	57
Obrázek 29: Aplikace - chybové hlášky reportu dat.....	58
Obrázek 30: Aplikace - Zjednodušená verze funkce pro kopírování dat z inventurních souborů	58

Seznam použitých tabulek

Tabulka 1: Datové typy.....	19
Tabulka 2 : Porovnávací operátory VBA	20
Tabulka 3: Tabulka HOS 8.....	37
Tabulka 4: Datová struktura pro kategorizaci a dlouhé názvy.....	43
Tabulka 5: Ekonomické zhodnocení.....	60