

# **Aplikace pro evidenci tvorby a průběhu zakázek, skladového materiálu a prvků zabezpečení proti nechtěnému poškození a ztrátě dat**

Evidence and Process Application for Orders and Repository  
Materials with Implementation of Security Interfaces.

Radek Bejda



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2012/2013

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Radek BEJDA**  
Osobní číslo: **A09771**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Bezpečnostní technologie, systémy a management**  
Forma studia: **kombinovaná**

Téma práce: **Aplikace pro evidenci tvorby a průběhu zakázek a skladového materiálu včetně prvků zabezpečení proti nechtěnému poškození a ztrátě dat.**

Zásady pro vypracování:

1. Vytvořte aplikaci pro evidenci zakázek a skladového materiálu osvitové firmy.
2. V rámci aplikace implementujte skladové hospodářství pro potřeby uživatele a sledování průběhu spotřeby materiálu.
3. V rámci aplikace implementujte tvorbu a průběh jednotlivých zakázek v návaznosti na spotřebu materiálu plynoucí ze zakázek.
4. Vytvořte uživatelské role pro rozdílný přístup v zobrazení evidence zakázek nutné pro provoz aplikace.
5. Realizujte nutné bezpečnostní prvky nutné pro ochranu dat aplikace.
6. Aplikaci otestujte a implementujte u uživatele včetně doladění synchronizace programu v interní síti firmy.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. JILKOVÁ, Helena – RYANT, Ivan. *Tvorba aplikací v objektovém prostředí*, Praha: Grada, 1994, 328 s. ISBN 80-85623-82-X.
2. Ondřej, Čada. *Objektové programování*, 1. vyd. Praha: Grada, 2009, 200 s. ISBN 978-80-247-2745-5.
3. LIBERTY, Jesse. *Naučte se C++ za 21 dní*, 1. vyd. Praha: Computer press, 2002, 766 s. ISBN 80-7226-774-4.
4. KRUGLINSKI, David, J. *Mistrovství ve Visual C++*. 1. vyd. Praha: Computer press, 1999, 854 s. ISBN 80-7226-132-0.
5. POKORNÝ, Pavel. *Objektově orientované programování v C++*. 3. vyd. Zlín: Univerzita Tomáše Bati ve Zlíně, 2010, 94 s. ISBN 978-80-73 18-913-6.

Vedoucí bakalářské práce:

**Ing. Dalibor Slovák**

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

**25. února 2013**

Termín odevzdání bakalářské práce:

**30. května 2013**

Ve Zlíně dne 25. února 2013



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Mgr. Milan Adámek, Ph.D.  
*ředitel ústavu*

## ABSTRAKT

Cílem bakalářské práce, je vytvořit jednoduchou objektově relační aplikaci pro osvitové studio, zabezpečenou pomocí definovaných uživatelských účtů s vymezením jejich pravomocí podle stanovených uživatelských rolí. Uživatelé budou přes vytvořené rozhraní aplikace data vkládat a zpracovávat. Data budou spravována pomocí SQL Serveru a sdílena pouze v interní firemní síti. Aplikace bude dále sledovat a zaznamenávat stav spotřeby materiálu, servisních událostí a poskytovat adresář odběratelů. Budoucí výhodou této aplikace, by měla být nezávislost na jiném komerčním programu s možností rozšíření o další služby či analýzy, bez nutnosti celkové změny aplikace a lepší zabezpečení proti ztrátě nebo poškození dat. Smyslem aplikace bude po dlouhodobějším provozu, možnost vytvářet statistiky z uložených dat a poskytovat informace pro další ekonomickou strategii firmy.

Klíčová slova:

Objektově relační databáze, uživatelské účty, uživatelské role, osvitová deska, zakázka

## ABSTRACT

The aim of this bachelor thesis is to create a simple object-relational application for the exposure studio and secured by defined user accounts and their respective powers under defined user roles. Users will enter data through the application interface and process them. The data will be managed by SQL Server and shared only on an internal corporate network. The application will also monitor and record the state of material consumption, service events and provide a directory of customers. Future advantage of this application should be the independence of any other commercial program, the possibility of further service or analysis extension without changing the application and overall better security against loss or damage to data. The purpose of the application after the long-term operation will be the possibility of creating statistics from stored data for further economic strategy of the company.

Keywords:

Object-relational database, users accounts, users roles, plate imagesetter, contract

Motto:

„Know where to find the information and how to use it – That’s the secret of success.“

Albert Einstein

Poděkování

Touto cestou bych rád poděkoval vedoucímu bakalářské práce panu Ing. Daliborovi Slovákovi za odborné rady, připomínky a trpělivost při jejím metodickém vedení, za pomoc při testování programu a s tím souvisejících dalších úkonů.

Dále bych rád poděkoval i vedoucímu osvitového oddělení firmy ART-D panu Radku Homolovi, za odbornou pomoc při upřesňování základních komponentů pro zpracování projektu.

**Prohlašuji, že**

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

**Prohlašuji,**

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

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 BEZPEČNOSTNÍ TECHNOLOGIE INFORMAČNÍCH SYSTÉMU A OCHRANA INFORMACÍ</b> .....	<b>12</b>
1.1    INFORMACE, DATA .....	13
1.1.1    Informace .....	13
1.1.2    Data .....	15
1.1.3    Integrovaná omezení dat .....	16
1.1.4    Ochrana dat .....	16
1.2    PRÁVNÍ PROSTŘEDÍ – LEGISLATIVA, NORMY .....	17
<b>2 DATABÁZOVÉ SYSTÉMY</b> .....	<b>19</b>
2.1    DATABÁZOVÉ MODELY .....	19
2.2    NORMÁLNÍ FORMY .....	20
2.3    VZTAHY, RELACE A SŘBD .....	21
2.3.1    SŘBD – Systém řízení báze dat .....	22
2.4    VÝVOJOVÉ PROSTŘEDÍ .....	22
2.4.1    Platforma .NET .....	23
2.4.2    MS SQL Server, C#, Visual Studio .....	23
2.5    ŘÍZENÍ A SPRÁVA UŽIVATELSKÝCH OPRAVNĚNÍ V INFORMAČNÍM SYSTÉMU .....	25
2.5.1    Řízení přístupů na základě uživatelských rolí RBAC .....	26
<b>3 DATABÁZOVÁ APLIKACE PRO OSVITOVÉ STUDIO</b> .....	<b>28</b>
3.1    ČINNOST ODDĚLENÍ.....	28
3.1.1    Specifikace osvitového oddělení.....	28
3.1.2    Určení základních vztahů pro tvorbu databáze .....	28
<b>II PRAKTICKÁ ČÁST</b> .....	<b>30</b>
<b>4 TVORBA DATABÁZOVÉ APLIKACE - POSTUP</b> .....	<b>31</b>
4.1    SPECIFIKACE POŽADAVKŮ .....	31
4.2    UŽIVATELSKÉ ROZHRAŇÍ.....	32
4.3    STANOVENÍ PRAVIDEL DATABÁZE .....	32
4.4    KONCEPTUÁLNÍ MODEL .....	34
4.5    LOGICKÝ MODEL .....	36
4.6    TVORBA TABULEK.....	36
4.6.1    Statické tabulky pro selektivní výběry .....	36
4.6.2    Dynamické Tabulky .....	38
4.7    FORMULÁŘE A DOTAZY .....	40
<b>5 POUŽITÍ, A POHYB UŽIVATELE V APLIKACI</b> .....	<b>41</b>
<b>ZÁVĚR</b> .....	<b>48</b>

<b>ZÁVĚR V ANGLIČTINĚ.....</b>	<b>49</b>
<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>50</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>52</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>53</b>
<b>SEZNAM TABULEK.....</b>	<b>54</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>55</b>

## ÚVOD

Žijeme v moderní společnosti, ve které je závislost na informačních systémech nezbytná, a která si klade vysoké nároky nejenom na fyzickou ochranu občanů a majetku, ale i na důležité informace uložené v databázích. Cesta k informační společnosti je podporována současnou technologickou revolucí, která je založena na vzájemném propojení informačních a komunikačních technologií.

Výrazný nárůst používání informačních technologií v současném světě vede na jedné straně k vytvoření informační společnosti, urychlení komunikace a velkému rozvoji služeb a celé společnosti. Na druhé straně se vzrůstající závislosti společnosti na informačních technologiích, ale vzrůstá i riziko zneužívání těchto technologií. To má rozsáhlé dopady na činnosti subjektů, které s nimi pracují, což může vést ke značným škodám. Obecným trendem je dnes kvalitní ochrana.

Data dnes nestačí chránit před jejich ztrátou, ale ochránit je před jejich zneužitím, respektive před jejich přečtením, vytěžením a neoprávněným užitím neautorizovanou osobou.

Pracuji ve firmě, kde kromě jiného spravuji společnou databázi našeho oddělení. V průběhu několika posledních let se ve své současné podobě stává tato databáze zastaralou a nekompatibilní s novějšími verzemi komerčních programů. Nehledě na to, je tato databáze nedostatečně chráněna proti neoprávněnému přístupu do aplikace i k datům.

S ohledem na tyto skutečnosti je cílem mé práce zpracování programu a vytvoření nové databáze, která umožní bezpečné uložení a uchování dat a zajistí k nim přístup pouze oprávněným uživatelům.

Nové vytvořená objektově relační databáze se bude moci do budoucna na základě modulů rozšiřovat nebo upravovat dle aktuálních požadavků firmy v běhovém prostředí platformy „.NET“ pod operačním systémem Microsoft Windows. Aplikace nebude závislá na jiných komerčních aplikacích a bude přístupná pouze registrovaným uživatelům databáze a provozována výhradně na lokální síti firmy. Datový prostor databáze bude spravován prostřednictvím SQL serveru, na který se bude aplikace relačně dotazovat. Součástí aplikace po naplnění daty bude možné vytváření specifických přehledů v rámci náplně činností oddělení firmy, oddělení firmy jsou samostatné oddělené jednotky, a také

vytváření přehledů v rámci časových souvislostí. Tato databáze umožní efektivnější využití prostředků a zvýšení hospodárnosti v rámci tohoto oddělení.

Tímto způsobem bude možné ověřit funkčnost vlastního modelu databázové logiky na základě získaných zkušeností ve firmě (včetně specifikací) a vyzkoušet, zda navrhované řešení bude pro firmu do budoucna více vyhovující, než současné používané.

## **I. TEORETICKÁ ČÁST**

# 1 BEZPEČNOSTNÍ TECHNOLOGIE INFORMAČNÍCH SYSTÉMU A OCHRANA INFORMACÍ

V dnešní době je nedílnou součástí moderních datových sítí zabezpečení datové infrastruktury proti útokům z venčí. Většina firem si dnes již plně uvědomuje nutnost ochrany svých informačních aktiv, a proto hledají řešení bezpečnosti na úrovni, která odpovídá jak jejich nárokům na bezpečnost, tak jejich finančním možnostem. Celkové zabezpečení informačního systému se skládá z mnoha dílčích bezpečnostních technologií.

Pokud chceme bezpečnostní informační systém integrovat v nějaké konkrétní firmě, nejedná se pouze o dodávku jednotlivých bezpečnostních komponent nebo technologií, ale jde zejména o komplexní individuální řešení zabezpečení informačních a komunikačních systémů s aplikacemi norem, při zachování nebo zlepšení uživatelského komfortu a účinné, srozumitelné a jednoduché administrace a bezpečnosti správy.

Informační technologie, informační systémy a zpracovávané informace jsou heterogenní, a všudypřítomné. Mohou být zabezpečené proti odcizení jen hůře nebo lépe. Přístup k firemním informacím ať už lokálně nebo vzdáleně, přes důvěryhodné pevné nebo mobilní připojení má být umožněno jen definovaným uživatelům, nikoliv nepovolané třetí straně. Oddělení osobních a pracovních činností uživatele ve firemním digitálním světě je obtížné. Rizika v tomto prostředí informačních technologií a informačních systémů nejsou a nebudou tedy nikdy konstantní, a proto jej nelze považovat za triviální proces.

Protiopatření k preventivnímu zabezpečení informací vyžaduje aktualizaci. Největší podíl bezpečnostní incidentů jako jsou nedostupnost dat či služby, ztráta integrity nebo důvěrnosti informací se odehrává právě uvnitř podniku „mezi počítačem a uživatelem.“ Nezáleží zde přitom, jakou roli hraje uživatel v celkovém informačním systému.

Základem zabezpečení informací je, mimo souvisejícího technického, organizačně administrativního a fyzického zabezpečení, i nezpochybnitelné přidělení potřebných aplikací a prostředků konkrétnímu uživateli pro jeho přidělené role ve firmě a vytváření nezpochybnitelných auditních záznamů o přístupech uživatelů k informacím a jejich využití. To zmenšuje riziko neprůkaznosti v případě řešení bezpečnostní události.

Bezpečností integrace je zároveň také i prevencí, která snižuje počet bezpečnostních událostí a pomáhá při řešení těchto událostí, kdy umožňuje objektivně a v krátkém čase zjistit: kdo, kdy, odkud a jak, s jakými informacemi a případně i proč nakládal.

Bezpečnostní integrace bývá nejčastěji řešena optimalizací a aktualizací stávajících bezpečnostních či informačních systémů. Další možnost je dodávka zcela nového konceptu bezpečnostního řešení s využitím stávajících nebo nových bezpečnostních prvků, aby bezpečnostní správce nebo správce daného informačního systému byli dopředu včas varováni o vzniku nestandardních situací a náklady na případné odstranění důsledků byly možná co nejnižší.

## 1.1 Informace, Data

Je to pojem velmi široký a přesná definice asi nikdy existovat nebude. Záleží na tom, v jakém oboru nebo oblasti zájmů se pohybujeme pro získání informací a na základě toho jsou aplikovány specifické přístupy k jejímu zkoumání včetně způsobů definování a přenosu.

Termíny informace nebo data lze v běžném hovoru považovat za synonymní a jsou natolik příbuzné, že je prakticky nejde definovat jinak než pomocí nich samých. Pro pochopení podstaty informace stojí však za to, zamyslet se nad jejich odlišnostmi. Odlišnost mezi daty a informacemi se projeví, začneme-li uvažovat nad jejich účelem a vzájemnými souvislostmi.

### 1.1.1 Informace

Slovo informace pochází z latinského slova „In-formatio“ a znamená „uvádět ve tvar.“ Ve velmi obecném smyslu můžeme informace chápat jako údaj o reálném prostředí, o jeho stavu a procesech, které v něm probíhají.

Můžeme je členit nebo posuzovat z několika hledisek.

#### laický, pohled na informaci

- sdělení, zpráva, nebo jazykový projev, který je vybudovaný na principu informačního slohového postupu a kde se co nejobjektivněji věcně a dokumentaristicky konstatují určitá fakta;

#### filozofické pojetí informace

- poznatek o určité objektivní realitě, předmětu nebo jevu, které jsou zachyceny ve zpřístupnitelné formě a jsou využitelné při přizpůsobování se člověka životnímu prostředí;

- význam přiřazený obrazům, údajům a z nich utvořeným lidským celkům;

#### **komunikační pojetí informace**

- obsah procesu lidské komunikace, odevzdávání a přijímání oznámení, jejich přenos osobním kontaktem, zvukem, signálem a prostředky masové komunikace, který má smysl pro komunikátora i příjemce a mezi souvisejícími objekty a projevující se změnou stavů těchto objektů;

#### **kybernetické pojetí informace**

- znamená obsah toho, co se vymění s vnějším světem, když se mu přizpůsobujeme a působíme na něj;
- zpráva o objektivní realitě, která funguje jako zpětná vazba;
- část poznání, která se používá k orientaci, k aktivní činnosti, k řízení, s cílem zachovat kvalitativní specifičnost systému a tento systém zdokonalovat a rozvíjet;
- proces, kdy určitý systém předává jinému systému pomocí signálů zprávu, která nějakým způsobem mění stav přijímacího systému;

#### **matematický přístup k informaci**

- energetická veličina, jejíž hodnota je úměrná zmenšení entropie systému;
- poznatek, který omezuje nebo odstraňuje nejistotu týkající se výskytu určitého jevu z dané množiny možných jevů;
- obsah zprávy, který je definován jako záporný dvojkový logaritmus její pravděpodobnosti. [1]

Dále můžeme informace rozdělit podle různých typů:

- vědeckotechnické (např. odborné časopisy...)
- ekonomické (např. ekonomické...)
- normotvorné (např. technické formy...)
- technické (např. napětí v rozvodné síti...)
- sociální (např. výsledky sčítání lidu...)

- politické (např. denní tisk...)

Ale i forem:

- **verbálními** (např. telefonní sdělení)

- **textovými** (např. osobní formulář)

- **numerickými** (např. účet za plyn)

- **obrazovými** (např. televizní zpravodajství)

Základní jednotkou informace je BIT (binary digit – dvojkové číslo). Jedná se o nejjednodušší typ informace tvořený pouze dvěma hodnotami 0 a 1 nebo chceme-li také pravděpodobnostními hodnotami ANO-NE. [2]

### 1.1.2 Data

Podle Všeobecné encyklopedie definice zní: Data, údaje, skutečnosti – jsou vhodným způsobem vyjádřená (zakódovaná) zpráva, která je srozumitelná příjemci (člověk, počítač) a přizpůsobená k dalšímu zpracování. Bezprostředně odrážejí skutečnost a představují nejnižší prvek informačního systému – **primární data**. Procházejí dalším zpracováním – **sekundární data**, např. počítači, a po následné analýze se stávají podkladem formulace empirických tvrzení(faktů). Data se uchovávají na nosičích dat, zpracovávají se různými typy prostředků výpočetní techniky. Mohou sloužit jako vstupní údaje nutné k řízení procesů v různých typech systémů.“ [8]

Data můžeme jednoduše definovat jako libovolnou posloupnost znaků a nemusí se jednat pouze o bity či bajty. Pod touto posloupností si lze představit libovolné znaky tedy i ty, které vůbec zatím ještě neznáme nebo u kterých si nedokážeme představit, že jde o nějaké písmo či znak. Vlastníme-li data, ale nechápeme-li jejich smysl, ztrácí pro nás význam a jsou nepoužitelná. Avšak pokud nám dávají smysl, představují pro nás něco užitečného a z dat se stávají informace, které můžeme zase zpětně zpracovat do podoby dat.

Jedná se o soubor statických hodnot, které jsme získaly na základě měření nebo pozorování určité skutečnosti nebo reality. Jsou někde uloženy, dnes již nejlépe v digitální podobě na počítači nebo na serveru a připravené k dalšímu zpracování a získání nových informací.

### 1.1.3 Integritní omezení dat

V databázích se jedná o velmi důležitou vlastnost. Znamená, že data uložená v souborech vyhovují pouze určitým pravidlům a mají tedy určitá integritní omezení. Do databáze lze vkládat pouze taková data, která jsou požadována a systém sám svými nástroji odpovídá a zajišťuje jejich správné uložení a manipulaci na základě zadaných oprávnění. Můžeme je rozdělit na 3 hlavní části:

1. Referenční integrita – jedná se o zajištění bezpečnosti dat na základě vztahů mezi dvěma tabulkami tak, aby odkazy na sebe navzájem nebo mezi dalšími byli platné a odkazovali na existující tabulky a změny v jedné tabulce byly stejným způsobem zobrazeny v tabulce druhé, pokud odkazuje na ní.
2. Entitní integrita – jedná se o zajištění na základě jedinečnosti klíče tak, aby každý záznam měl nejméně jeden unikátní klíč, který nebude už nikde jinde opakován a na jehož základě je tento záznam jedinečný a také nebude obsahovat hodnotu NULL – neboli prázdnou buňku.
3. Doménová integrita – jedná se o zajištění platnosti vkládaných dat a do polí byla vkládána pouze správná data na základě definovaných datových typů.

### 1.1.4 Ochrana dat

V dnešní době nestačí data chránit pouze před jejich ztrátou, ale především je potřeba chránit je před jejich zneužitím. Jde obzvláště o jejich přečtení nebo kopírování neoprávněnou či nepovolanou osobou. Při nedůsledné ochraně může dojít ke zneužití kontaktů odběratelů, dodavatelů, účetních či technologických dat a postupů, ale také i účtů nebo se ke konkurenci dostanou data ohledně vlastních podnikatelských plánů či výrobních patentů a může dojít i k porušení Zákona o ochraně osobních údajů.

Největším problémem v této oblasti bezpečnosti jsou právě vlastní zaměstnanci, kteří přehlížejí bezpečnostní rizika spojená s citlivými firemními daty nebo internetem. Podle některých studií se uvádí až 70% zaměstnanců jako potenciální riziko trojského koně z důvodů nedostatečné vzdělanosti a nedbalosti.

Velice podceňovaným nebezpečím je uložení kopie dat na přenosné datové nosiče, notebooky nebo jiná úložiště dat, než která byla pro ně určena na bezpečném zálohovém

serveru. Tato data si pak může někdo cizí zkopírovat, a to aniž by o tom obvykle uživatel věděl. U notebooků se stávají součástí krádeže celého přístroje.

Uživatelé připojení k internetu i do interní podnikové sítě, ukládají data na stejný lokální pevný disk, mohou způsobit jejich ztrátu vlivem hardwarové závady nebo stažením a následnou aktivací virem, ale hlavně i nebezpečným průnikem do zabezpečené interní sítě. Špatně prověřeni uživatelé mohou svěřit hesla nebo data za úplatu nebo z jiných skutkových důvodů a předat je do nepovolaných rukou.

Každá organizace nebo firma by měla mít vypracovaný seznam oprávněných osob pro práci s jejich citlivými daty a jejich způsob kontroly před únikem a poškozením při případné práci mimo organizaci.

Nejvíce utajované skutečnosti (informace) je potřeba kromě hesla zajistit i případně dalšími způsoby autorizace jako jsou čtečky otisků očí, prstů, hlasové identifikátory, různé typy karet a další. Ačkoliv je heslo nejobvyklejší způsob autorizace, mohou být i přes všechna navržená bezpečnostní opatření snadno prozrazena a nemohou pouze sami zajistit maximální zabezpečení.

## 1.2 Právní prostředí – legislativa, normy

V současné době se stávají informace významnými aktivy organizací a tím zákonitě rostou požadavky na jejich důvěrnost, dostupnost a integritu. Tyto tři vlastnosti označujeme souhrnným pojmem bezpečnost informací (pro někoho informační bezpečnost).

1. **Důvěrnost:** informace musí být přístupné nebo jsou sdělovány pouze těm, kteří jsou k tomu oprávněni.
2. **Dostupnost:** informace jsou pro oprávněné uživatele přístupné v okamžiku jejich potřeby.
3. **Integrita:** zajištění správnosti a úplnosti informací.

Prověřeným návodem v oblasti bezpečnosti, jak se s těmito požadavky vypořádat jsou odpovídající zákony a normy.

Nejdůležitější normou v této oblasti je ISO/IEC 27001 – popisující systém řízení informační bezpečnosti (Information Security Management System – ISMS). Cílem ISMS je nastavit řízení procesů tak, aby byla zachována dostupnost, integrita a důvěrnost

informací, které jsou pro firmu důležité. Systém se zabývá informacemi jako takovými a to bez ohledu na to jakou mají formu (datovou, papírovou).

**Související legislativa:** obchodní zákoník (část týkající se obchodního tajemství), zákon o ochraně utajovaných informací, zákon o ochraně osobních údajů, zákon o elektronickém podpisu, zákon o některých službách informační společnosti a další.

**Související normy:** BS 17799-2 (předchůdce normy ISO/IEC 27001), ISO/IEC 1799 a její přečíslovaná verze ISO/IEC 27002, které navazují na předchozí normy a další normy zabezpečující tuto oblast. [10]

## 2 DATABÁZOVÉ SYSTÉMY

Tento pojem v dnešním světě je už zcela dobře známý. S různými typy databázových systémů se dnes můžeme setkat již na každém kroku, ať již jde o databázi např. Registru vozidel, občanů, ale i v účtů v bankovním systému, nebo i jednoduchý zakázkový systém živnostníka až po složité databázové struktury velkých společností.

### 2.1 Databázové modely

1. **Hierarchický model** – Je nejstarším typem, a vztahy podřízenosti a nadřízenosti mezi jednotlivými entitami jsou zobrazovány do stromové struktury na základě dědičnosti ve stylu rodič-potomek. Nevýhodou tohoto modelu bylo složité programování rozsáhlejších struktur a také při chybě uvnitř této struktury snadno došlo ke zamrznutí celé větve bez možnosti záchrany nebo kroku zpět.
2. **Síťový model** – tento model vyřešil do jisté míry, mnohonásobné vztahy, tím že umožňuje, aby potomek mohl mít více rodičů než pouze jednoho a díky tomu lze modelovat i složitější vztahy M:N a datové struktury. Navíc pokud dojde k ukončení procesu jednoho rodiče, nemusí to vést nutně k ukončení celé dědičné větve.
3. **Relační model** – tento model byl původně navržen E. F. Coddem a později implementován firmou IBM. Základem jsou data uložená do řádků a sloupců v definovaných tabulkách, kde řádek tvoří jednotlivý záznam, a sloupec tvoří jeho atribut. Tabulky jsou identifikovatelné pomocí tzv. primárních klíčů nebo mezi sebou provázané na základě cizích klíčů (společných atributů) a každý atribut záznamu může uchovávat pouze jednu hodnotu. Stal se standardem popsáním v normě ISO a ANSI.
4. **Objektový model** – základní ideou tohoto modelu jsou objekty, ze kterých se jednotlivé části skládají a mohou fungovat nezávisle na sobě. Ke komunikaci mezi sebou a identifikaci využívají třídy, instance a metody a podporují zapouzdření nebo mnohonásobnou dědičnost. Databáze jsou většinou programovány v objektově orientovaném jazyce v kombinaci s databázovými schopnostmi implementovanými navíc, jež poskytují plnou kontrolu nad celou strukturou

databáze. Mezi ty známější programovací jazyky patří (C#, C++, Java, SmallTalk, PHP, Perl) nebo čistě objektový jazyk Ruby.

5. **Objektově-relační model** – tento model v sobě spojuje oba dva předchozí modely. Základem je většinou objektová aplikace a relační je přístup k datům uloženým na SQL Server. Trvalé informace jsou uloženy stále v tabulkách na serveru, ale u některých lze navíc vytvořit rozšiřující datové struktury zkombinováním základních datových typů a lze je použít při indexování nebo vytváření nových kritérií záznamů. Jeho nevýhodou je, že jazyk SQL je hlavním rozhraním pro správu databází a chybí dostatečná podpora pro objektové jazyky, čímž je nutné objekty a tabulky mezi sebou překládat.

## 2.2 Normální formy

U dobře navržené databáze je nutná normalizace vkládaných dat, abychom zamezili nebo omezili případnou redundanci dat a zamezili jejich nesprávnému aktualizování. Správně navržené tabulky splňují tři základní normální formy a další dvě jsou spíše rozšířením předchozích stupňů.

### 1. Normální forma - 1NF

Tato norma (1NF) říká, že všechny atributy jsou atomické, tj. dále již nedělitelné. Jednotlivé položky jsou rozděleny do co nejjednodušších částí a v záznamu se neopakují.

### 2. Normální forma - 2NF

Tabulka splňuje 2NF, právě když splňuje 1NF a navíc každý atribut, který není jen primárním klíčem, je na primárním klíči úplně závislý. To znamená, že se v řádku tabulky nesmí objevit položka, která by byla závislá jen na části primárního klíče. Z toho vyplývá, že problém 2NF se týká pouze tabulek, kde za primární klíč je zvoleno víc položek než jedna.

### 3. Normální forma - 3NF

Relační tabulky splňují třetí normální formu (3NF), právě když splňují 2NF a žádný atribut, který není primárním klíčem, není tranzitně závislý na žádném klíči.

#### 4. Normální forma - 4NF

Tabulky splňují čtvrtou normální formu (4NF), právě když splňují 3NF, atributy primárních klíčů musí být na sobě nezávislé a popisovat pouze jeden fakt.

#### 5. Normální forma - 5NF

Tabulka splňuje 5NF, pokud zároveň splňuje i 4NF a není možné do ní přidat nový sloupec nebo skupinu sloupců, tak aby se nerozpadla vlivem skrytých závislostí na několik dílčích tabulek.

#### Boyce-Coddova normální forma – BCNF

Tabulka splňuje BCNF, jestliže pro každou netriviální funkční závislost  $X \rightarrow Y$  je  $X$  „superklíčem“. Tuto normu splňují tabulky 1NF – 3NF ve většině případů. Případná dekompozice nemusí zachovávat závislosti v rámci relací a ty pak nejsou nezávislé. [3]

### 2.3 Vztahy, relace a SŘBD

Mezi jednotlivými entitami tabulek jsou určité vztahy, na jejichž základě se relace sestavují.

#### Vztah 1:1

Představuje nejjednodušší vztah jedna ku jedné. O vztahu entit  $X$  a  $Y$  říkáme, že je typu jedna ku jedné, pokud platí, že každou libovolnou instanci entity  $X$  je možné přiřadit pouze jediné instanci entity  $Y$ .

#### Vztah 1:N

Jedná se asi o nejběžnější vztah mezi entitami představující jedna k více, v nichž můžeme jedné instanci první entity přiřadit hodnotu nula nebo více instancí druhé entity. Tento vztah se vytváří nejčastěji v normalizačních postupech. Příkladem může být Firma  $\rightarrow$  Odběratelé.

#### Vztah M:N

Tento vztah odpovídá asi nejvíce reálné podobě světa a jedná se o vztah více k více. Tento vztah ale v relační databázi implementovat bohužel nejde a proto se modeluje přes speciálního prostředníka samostatné relace, která s oběma původními entitami má vztah 1:N.

Všechny tyto vyjmenované vztahy lze nazvat také binárními vztahy, jelikož v nich vždy vystupují dvě rozdílné entity. Oproti tomu vztahy unární představují entity, který mají relace pouze sami se sebou.

### 2.3.1 SŘBD – Systém řízení báze dat

Jakýkoliv databázový systém se skládá z několika částí:

- a) data tvořená databází (báze dat)-DB
- b) systém řízení báze dat (SŘBD) – programové vybavení, které spravuje a organizuje data v určité datové struktuře
- c) aplikace nebo program, který umožňuje uživatelský a snadný přístup k uloženým datům a umožňuje s nimi pracovat.

Systém báze řízení dat pak představuje programovou vrstvu, řešící operace nad databází, umožňuje odstínění uživatelů či aplikací od technických detailů a zabezpečuje operace pro práci se soubory v databázi včetně výběru informací dle zadaných kritérií. Jde zejména o podporu těchto hlavních funkcí databáze:

- CREATE - vytvoření báze dat
- INSERT – vkládání dat
- DELETE – rušení dat
- SELECT – výběr z báze dat dle požadavků
- UPDATE – aktualizace dat

Jazykem pro komunikaci u většiny databázových systémů bývá jazyk SQL a v dnešní době se můžeme setkat se spoustou firem, které se zabývají vývojem databázových systémů ať již komerčních nebo nekomerčních. Asi neznámějším představitelem v komerční sféře je Oracle nebo Microsoft se svým SQL Server nebo z nekomerční sféry MySQL, který je volně šiřitelnou platformou databázového serveru. [4]

## 2.4 Vývojové prostředí

V dnešní době lze využít mnoho nejrůznějších vývojových nástrojů a technologií. Díky kterým lze snadno vytvářet nové aplikace a v neposlední řadě také i databázové systémy.

### 2.4.1 Platforma .NET

Platforma .NET, vyvíjená společností Microsoft, byla navržena pro zjednodušení vývoje aplikací v běhovém prostředí pod systémy Microsoft Windows. Jedná se o nadstavbu operačního systému, který má podporu pro desktopové aplikace, webové aplikace a také nově pro mobilní aplikace pod systémem Windows Mobile. Základem je .NET Framework, nabízející běhové prostředí CLR (Common Language Runtime), spouštěcí rozhraní a rozsáhlé BCL knihovny, ze kterých je možné metody a funkce vybírat. Poskytuje také podporu pro mnoho programovacích jazyků, jako jsou C#, Visual Basic, C++, Java, F#, čímž je možné psát části programů v různých programovacích jazycích a libovolně je kombinovat. Další výhodou je automatická správa paměti, objektově orientovaný přístup a vzhledem k oddělenému jádru větší bezpečnost při spouštění v systému díky několika základním částem, jako je typová bezpečnost, identita kódu, přístupová bezpečnost kódu, povolení, bezpečnost založená na rolích a kryptografické služby. Nevýhodou je, že je určena výhradně pro systémy MS Windows. Není zde možná rekompilace zdrojových kódů a většinou jsou profesionální nástroje finančně nákladnější. [7]

### 2.4.2 MS SQL Server, C#, Visual Studio

**C# (C Sharp)** - je vysokoúrovňový objektově orientovaný a imperativní jazyk, který byl vyvinut společně s platformou .NET. Vzešel z kombinace jazyků C++ a Javy, od kterých přebíral syntaxi. Jeho standardem se stala norma ISO (ISO/IEC 23270). Nabízí nativní podporu pro programování komponent a atributové programování včetně zpracování chyb pomocí výjimek, obsahuje pouze jednoduchou dědičnost. O uvolňování zdrojů se stará tzv. garbage collector, který zajišťuje typovou bezpečnost a k metodám přidává ještě vlastnosti a události. Ke svému definování používá třídy a metody, které definují objekty pomocí instancí a komunikují spolu pomocí volání svých metod.

**Visual Studio** - je profesionální vývojový software (IDE) od firmy Microsoft určený pro vývoj nových aplikací. Lze zde vytvářet konzolové aplikace a také aplikace s grafickým rozhraním jak ve strojovém kódu, tak v řízeném kódu. Poskytuje plnou podporu pro tvorbu všech typů aplikací, za použití různých programovacích jazyků díky instalované jazykové službě. [11]

**SQL a MS SQL Server** - SQL je standardizovaný strukturovaný dotazovací jazyk (Structured Query Language), který byl speciálně navržen pro práci s daty v relačních databázích. V roce 1986 byl přijat jako standard ANSI. Patří mezi deklarativní programovací jazyk, což znamená, že kód vkládáme do jiného procedurálního programovacího jazyka.

Obsahuje tyto základní části:

- **DDL** (data definition language) – jazyk pro vytváření databázových schémat a katalogů;
- **SDL** (storage definition language) – definuje způsob ukládání tabulek;
- **VDL** (view definition language – umožňuje vytváření vlastních pohledů z více tabulek;
- **DML** (data manipulation language) – obsahuje základní příkazy pro manipulaci s daty.

**Microsoft SQL Server** – je relační databázová platforma od firmy Microsoft, sloužící právě pro práci s daty. Je založený na architektuře klient/server s integrovanou nativní podporou XML jazyka. Hlavním využitím je možnost tvorby a správy vlastních datových skladů a databází. Usnadňuje aplikacím rozhodování. V dnešní době existuje několik verzí z nichž poslední je verze 2012. Od verze 2005 jsou aplikovány tzv. studia, která rozdělují systém do několika dílčích samostatných rozhraní.

*SQL Server Management Studio* – rozhraní pro vytváření vlastních databází, hlášení řešení, spravuje také služby pro analýzy a reportingové servery a řídí integrační procesy..

*Bussiness Intelligence Development Studio* – nástroj pro řešení obchodních informací jako jsou analýzy a reporty k vytváření sestav a integrační služby.

*SQL Server Configuration Manager* – nástroj pro správu služeb a konfiguraci síťových protokolů a protokolů konektivity z klientských počítačů.

V této bakalářské práci bude použita verze 2008 R2, jelikož má podporu pro 64 bitové architektury počítačů a je již prověřená s dostatečným množstvím literatury. [6]

## 2.5 Řízení a správa uživatelských oprávnění v informačním systému

Většina aplikací je dnes charakteristická tím, že do nich vstupuje více uživatelů ve stejný časový okamžik a stejně tak i oni přistupují k datům. Aby bylo možné tyto požadavky splnit a současně zajistit bezpečnost díky vícenásobnému přístupu, je třeba vytvořit a definovat pro všechny uživatele určité role. Na základě těchto vytvořených rolí, pak mohou k aplikacím přistupovat odkudkoliv, tj. z různých počítačů a dokonce i z více instalovaných verzí aplikace přes interní síť firmy nebo přes internetové prohlížeče bez nutnosti instalace, jestliže se jedná o webovou aplikaci.

Toto řešení přináší větší možnost kontroly nad softwarem i daty aplikací, ale vyžaduje implementaci bezpečnostních mechanismů, které zajišťují, že se k důležitým firemním informacím dostanou pouze určení uživatelé, kteří s nimi budou nakládat pouze definovaným povoleným způsobem. To je dosaženo nejen použitím mechanismů pro řízení přístupu uživatelů k objektům aplikace, ale i správou uživatelských a systémových oprávnění.

Smyslem vytvoření řízení přístupů uživatelů (Access Control) a jejich oprávnění je omezit uživatele tak, aby mohl pracovat jen s určitými daty, objekty nebo sdílenými síťovými zdroji, jako jsou sdílené datové sklady, tiskárny, apod., ale také abychom dodrželi základní požadavky na bezpečnost informačního systému, kterými jsou:

1. **Důvěryhodnost** – k datům a objektům mohou přistupovat pouze autorizovaní uživatelé.
2. **Integrita** – data a objekty mohou upravovat pouze autorizovaní uživatelé.
3. **Dostupnost** – omezuje možnost poškodit systém nebo aplikaci na základě špatně udělených oprávnění možným útočníkům.

Toto je docíleno uplatněním specifických zásad bezpečnostní politiky v komerčním prostředí, a sice Zásadou minimálních oprávnění, která požaduje, aby uživatelé měli pouze minimální oprávnění pro přístup k minimálnímu množství informací a zdrojů, které stačí pro činnost vyplývající z jejich pracovní pozice. A druhou zásadou je rozdělení činností a souvisejících oprávnění mezi více pracovníků s cílem zamezit možným chybám, ať už z úmyslného nebo neúmyslného chování jediného uživatele.

Každý uživatel jakékoliv aplikace chráněné přihlašovacím skriptem je nejprve zaregistrován v aplikaci přes autorizační systém. Ten slouží pro identifikaci uživatelů aplikace na základě zadaných údajů a má definovanou svojí určitou množinu práv ve smyslu „co a kdy může s daty dělat“, tedy pouze některé operace (př. čtení, mazání, přesouvání, vytváření atd..) pod svojí identitou.

V aplikacích s větším počtem uživatelů i systémových oprávnění je výhodné uživatele a jejich oprávnění seskupovat a mapovat, v případě stejných přístupů, aby se nemuseli přidělovat oprávnění zvlášť. O toto se stará zejména správa uživatelských účtů aplikace, která představuje rozhraní pro přidávání a odebírání uživatelů včetně udělování systémových práv a kontrolní funkce, které hlídají případné zásady bezpečnostní politiky.

### 2.5.1 Řízení přístupů na základě uživatelských rolí RBAC

Role-Based Access Control – je dnes nejvíce aplikovaný bezpečnostní model pro řízení přístupů uživatelů ke zdrojům a umožňuje aplikovat velké množství bezpečnostních politik nezávisle na použitém softwaru. Model se skládá ze čtyř částí, první představuje strukturu RBAC a další tři definují Uživatele, Role, Oprávnění a jejich vztahy mezi nimi. Základní ideou je zde, že Uživatelé jsou přiřazována Oprávnění prostřednictvím jeho Role.

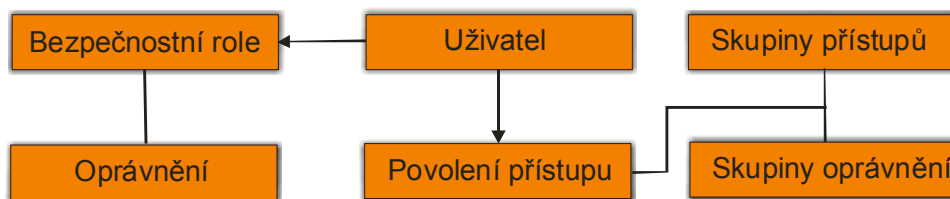
RBAC definuje tyto 3 základní pravidla:

**Přidělování rolí** – první pravidlo říká, že subjekt může vykonávat oprávnění pouze tehdy, pokud byl vybrán, nebo mu byla přidělena určitá role.

**Role povolení** – druhé pravidlo říká, že aktivní role subjektů musí být povoleny pro daný objekt a spolu s pravidlem prvním zajišťují, že uživatelé mohou používat pouze takové role, které jim byly přiděleny.

**Povolení povolení** – třetí pravidlo říká, že subjekt může vykonávat pouze taková oprávnění, která byla registrována pro jeho aktivní roli a zajišťuje spolu s pravidlem 1 a 2, že uživatelé mohou vykonávat pouze taková oprávnění, která jim byla přidělena.

V praxi řízení přístupu tak umožňuje programátorům a administrátorům spravovat datový model a zároveň kontrolovat, zda uživatelé provádí pouze k nim odpovídající operace tak, jak znázorňuje obrázek 1. Role administrátora by měla být součástí každé aplikace s plným přístupem do všech částí aplikace a případně i kódů.



Obr. 1 – Schéma řízení přístupu

V prostředí .NET je možné použít dva způsoby ochrany zdrojů a kódů. První způsob je již zmíněná bezpečnost založená na rolích a druhý způsob je chráněný přístup ke zdrojovým kódům, který je ovládán právě pomocí povolení a zákazů.

Druhý způsob ochrany kódů chrání počítačové systémy před cizími kódy, které by mohly přepsat nebo vložit svůj kód do těla aplikací případně jej poškodit. Takovým příkladem mohou být zejména viry a červy. Prostředí .NET umožňuje použití mechanismu chráněného přístupu různých stupňů důvěryhodnosti kódů, které závisí na jeho původu a dalších aspektech identity kódu.

V objektových aplikacích jsou tyto záležitosti řešeny vhodným použitím bezpečnostních systémových knihoven, správným definováním instancí a metod včetně toho, kdo je může volat nebo obsluhovat. Co se týká rolí, využívá prostředí .NET sjednocený model uživatelské identity a rolí pro autorizaci na základě procesu autentizace.

Poslední věc, která stojí za zmínku v těchto souvislostech, se týká i nastavení pravidel a zásad pro ukládání nebo definování hesel. Jejich správné dodržování může podnikům pomoci, zabránit lakačům kódů dostat se do interních podnikových sítí nebo zjistit totožnost uživatelů a zabránit tak opět prozrazení nebo poškození důležitých informací.

### 3 DATABÁZOVÁ APLIKACE PRO OSVITOVÉ STUDIO

Cílem projektu je vytvoření malé databázové firemní aplikace, přes kterou bude možné, zakládat, evidovat a sledovat průběh zpracování zakázek této firmy, ale také pohyb spotřeby osvitového materiálu ve skladu na základě požadavků plynoucí ze zakázek. Systém bude přístupný pouze pro zaměstnance osvitového oddělení této firmy, kteří budou přes uživatelské rozhraní s aplikací komunikovat a vkládat data do databáze a následně sledovat průběhy zpracování všech aktuálních zakázek současně.

#### 3.1 Činnost oddělení

Nejdříve, aby bylo možné určit požadavky a specifikace, je nutné představit náplň daného firemního oddělení, pro které bude aplikace vytvořena. Jedná se o firmu, jejímž oborem je polygrafie a služby s tímto oborem spojené. Jedno z firemních oddělení se zabývá přípravou archové montáže a následným osvitem, za využití osvitových strojů těchto elektronických montáží, na tenký kovový hliníkový list. Zjednodušeně si lze představit osvitový stroj jako velkou laserovou vypalovačku specifických elektronických dat na médium, které následně tiskárny odebírají pro tisk a kompletaci všech typů dokumentů.

##### 3.1.1 Specifikace osvitového oddělení

Osvitové oddělení má dvě povahy své pracovní náplně. První (počáteční) fází je elektronické zpracování dokumentů a kontrola úplnosti podkladů zaslaných tiskárnou, jako zakázka k přípravě pro archový nebo ofsetový tisk pro tiskařské stroje. V druhé (koncové) fází dochází k vypálení těchto „příprav“ přes specifický osvitový stroj na daný typ osvitové desky pro konkrétní tiskárnu. Zároveň dochází ke skutečné spotřebě materiálu daného typu a rozměru osvitové desky ze skladu. Následně dochází k archivaci zakázky v databázi.

##### 3.1.2 Určení základních vztahů pro tvorbu databáze

Firma využívá v současnosti databázový systém vytvořený v aplikaci MS Access 2003. Jeho životnost pomalu končí, zejména s přechodem na novější verze balíku MS Office 2007, kdy došlo ke změně některých knihoven aplikace MS Access. Tyto starší verze hůře komunikují nebo neznají nové knihovny a to včetně zobrazování jednotlivých formulářů na pracovních stanicích v závislosti na verzi systému i aplikace. Další změna během

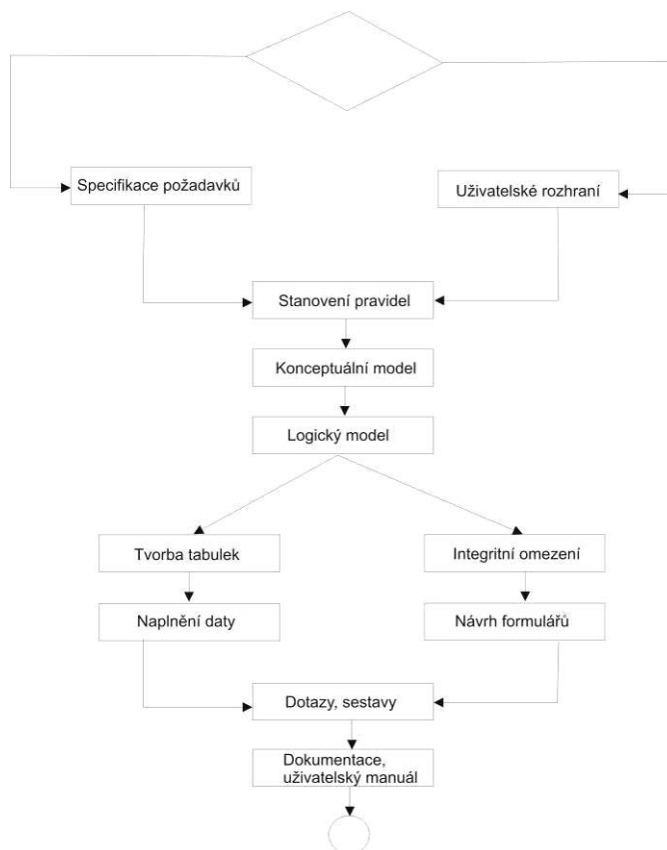
posledních let proběhla celkově v novém uspořádání chodu tohoto oddělení a správy zakázek. V neposlední řadě i změnou hlavních odběratelů, než pro které byla databáze původně navržena. Její velkou nevýhodou je to, že databáze je otevřená a přístup k ní má tedy každý, kdo zasedne k počítači. Nebezpečí úniku informací je zde vysoké, ačkoli se jedná o aplikaci provozovanou pouze v podnikové síti. Další nevýhodou je společná tabulka záznamů všech zakázek. Pokud dojde k poškození hlavního souboru je databáze prakticky zcela nefunkční, jelikož správa a údržba databáze je řízena interně aplikací MS ACCESS. Její přepracování by bylo bez chybějící dokumentace o vztazích a relacích mezi jednotlivými tabulkami časově náročné, včetně kompletní změny grafického zobrazení a do budoucna by nespĺnilo požadavky, které jsou na ní v současnosti kladeny.

Hlavní položkou databáze jsou zde zakázky jednotlivých tiskáren, které prochází určitou fází zpracování podle určitého postupu předpřipravy a technologie montáže na základě specifikace tiskárny. Databáze umožňuje tyto zakázky vytvářet, a sledovat postup zpracování přes uživatelské rozhraní, zaznamenávat a poté archivovat pro opětovné načtení a použití. Druhým artiklem jsou již hmotné osvitové desky, které se spotřebovávají v poslední fázi zpracování zakázek. Tyto osvitové desky jsou přijímány do skladu materiálu, a požadavkem ze zakázek dochází k jejich spotřebě, která je databází také sledována. Databáze pak na základě dotazů denně, měsíčně a ročně vyhodnocuje a zobrazuje celkovou spotřebu všech typů desek na skladě, ale také jednotlivých tiskáren. Zobrazení zmetkovitosti desek, u kterých dojde během osvitů k chybě mechanické ze stroje nebo jeho součástí, chybě pracovníka při elektronickém zpracování, jehož chyba se následně promítne také na desky, je řešeno selektivním výběrem typu chyby při zápisu údajů o spotřebě do té dané zakázky. Další položkou již spíše informativní, jsou záznamy o provedených údržbách osvitových strojů zaměstnanci, které slouží jako zpětná vazba komunikace mezi sebou navzájem. V neposlední řadě je využita jako adresář odběratelů a kontaktů včetně jejich specifických požadavků k tisku.

## **II. PRAKTICKÁ ČÁST**

## 4 TVORBA DATABÁZOVÉ APLIKACE - POSTUP

Pro snahu o jednodušší vysvětlení budu postupovat podle grafického modelu:



Obr. 2 – Návrhový model

### 4.1 Specifikace požadavků

Hlavním požadavkem je nezávislost aplikace na jakémkoliv komerčním programu a kompatibilita s jakýmkoliv typem PC 32, 64 - bit pod systémem Windows XP, Windows 7, jelikož na těch jsou ve firmě provozovány veškeré aplikace. Databázová aplikace musí být zabezpečena proti neoprávněnému vstupu neregistrované osoby do aplikace a data zabezpečena proti poškození a jejich ztrátě při manipulaci, editaci a případným kolizím interní sítě během komunikace s SQL serverem. Dalšími požadavky jsou :

1. zakládání jednotlivých zakázek uživatelem přes uživatelské rozhraní aplikace;
2. sledování stavů průběhu zpracování jednotlivých zakázek, od vzniku až po dokončení (konec = vyřazeno a osvětleno), a jejich editace v průběhu zpracování podle požadované fáze přes uživatelské rozhraní;

3. evidence počtů osvitových desek ve skladu;
4. sledování spotřeby osvitových desek na osvitových strojích podle požadavků ze zakázek a vyhodnocování celkových součtů pro potřeby zaměstnavatele;
5. kontaktní seznam odběratelů a jejich specifikací pro tisk;
6. seznam uživatelů, rozdělení přístupu a práv uživatelů do systému s ohledem na zabezpečení proti poškození uživatelem (omezený přístup);
7. zaznamenávání a sledování událostí spojených s údržbou osvitových strojů;
8. společný síťový přístup ke zdrojovým datům přes dotazy z dané databázové aplikace ve spojení s SQL Serverem..

## 4.2 Uživatelské rozhraní

Vývojové prostředí pro tvorbu dané databázové aplikace jsem vybral MS Visual Studio 2010 a platformu .NET Framework od firmy Microsoft. Je určeno pro tvorbu programů a běhu v systému Windows. Jazykem pro psaní programu jsem si vybral C# (C Sharp) verze 4.0, který umožňuje plné využití veškerých funkcí pro tvorbu myšlené aplikace. Zároveň tento jazyk dokáže dobře komunikovat i s jazykem SQL, který bude použit v rámci definování propojení s databázovým serverem MS SQL Server 2008.

Veškeré formuláře a okna pro komunikaci s uživatelem budou vytvořené v programu Microsoft Expression Blend 4, který nabízí příjemnější grafické rozhraní pro tvorbu formulářů, má široký výběr interaktivních komponent pro jednodušší obsluhu a nevytváří žádné další generativní kódy.

## 4.3 Stanovení pravidel databáze

Databáze bude mít několik zásadních pravidel pro obsluhu a vkládání dat. Hlavním důležitým pravidlem bude zamezení přístupu neregistrovaným osobám tak, že do aplikace kromě administrátorského přístupu bude povolen přístup pouze pracovníkům, kteří budou aplikaci využívat a které do systému bude registrovat pouze vedoucí oddělení. Vedoucího oddělení jako prvního do aplikace zaregistruje právě administrátor. Zároveň tabulka o přehledu registrovaných uživatelů, včetně jejich identifikačních údajů, bude v aplikaci zobrazena pouze vedoucímu stejně jako mazání a další úpravy záznamů. Uživatelé, kteří budou rozdělení dále na pozici montážník nebo osvitář, budou moci data pouze vkládat a

vidět jenom přehledy, které se týkají jejich pracovní náplně. Hierarchicky je možné říct, že hlavní role je administrátor společně s vedoucím, kteří budou mít kompletní pravomoci pro obsluhu aplikace, dále pak montážník s částečným omezením a jako poslední bude osvitář s nejvyšším omezením pohybu v aplikaci. Kdo, bude mít jaké práva, bude ještě rozebráno v dalších kapitolách.

Další pravidla se týkají aplikace pro manipulaci s daty a jejich vkládání. Tyto pravidla jsou:

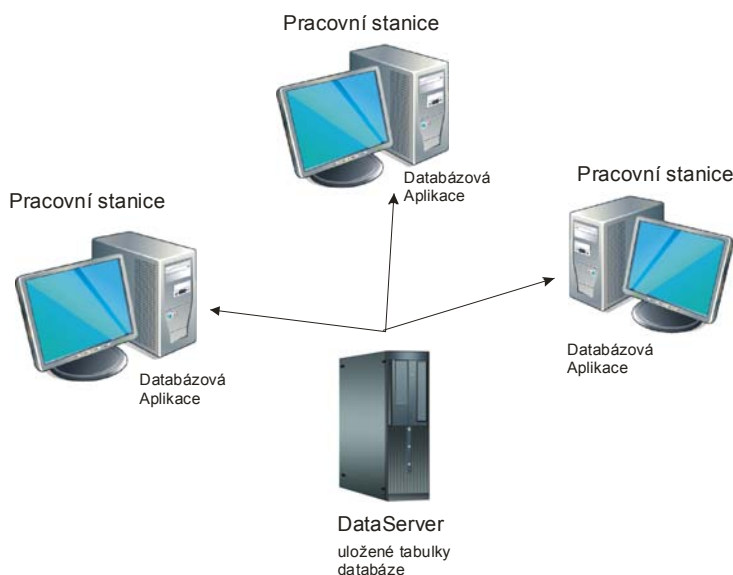
1. Nového odběratele do kontaktů, který má zároveň i nový formát osvitové desky, lze přidat do databáze až po přidání nového formátu desky do databáze, respektive do skladu. Základní úvahou je, že nelze přijímat zakázky od odběratelů, kteří mají jiný formát desek než firma používá pro ostatní odběratele. Proto bude potřeba desky nejprve objednat a zaevidovat do skladu, neboť odběratel bude mít vazbu na daný formát desek.
2. Jednotlivé zakázky budou ukládány do tabulky na SQL server. Zakázka bude v sobě obsahovat mnoho informací a aplikace bude zobrazovat jejich vybraný seznam atributů pomocí dotazů a stejně tak umožňovat jejich editaci. Naopak seznam odběratelů, osvitových desek, uživatelů a záznamů událostí o údržbě strojů budou ukládány do společné tabulky pro každý daný typ zvlášť. Pokud dojde k poškození záznamu, bude jednodušší opravit jednotlivý záznam zakázky nebo případně vytvořit novou kopii, než ztratit všechny najednou, pokud by byly uloženy v jediném společném souboru a bez správy. Také by bylo mnohem složitější zabezpečit ho pro mnohonásobný přístup ve stejném časovém úseku.
3. Na jedné zakázce může současně pracovat pouze jeden uživatel, takže bude přístup dalším, až do ukončení editace tedy uložením souboru, odmítnut.
4. Mazání souborů a jejich celkové spravování přes databázovou aplikaci bude oprávněním pouze vedoucího oddělení, aby se zamezilo nechtěné manipulaci dat a změn vstupních údajů.
5. Nejnižší pozice osvětáře, bude mít v databázi povoleno pouze vkládání dat týkajících se osvitů a událostí ohledně údržby strojů, do zbytku bude přístup odepřen podle definovaných oprávnění.

#### 4.4 Konceptuální model

Aby bylo možné aplikaci a databázi nějakým způsobem vytvořit a naprogramovat, je důležité si nejprve vytvořit modely, které budou co nejdříve popisovat námi myšlený model aplikace a databáze. Ideální případ získání prvotních informací je metoda pozorování, a typizování. Také běžný pohovor s pracovníky nebo majiteli o činnosti firmy nebo jejich náplně práce, může vést k prvotním poznatkům o podobě budoucí databáze či aplikace. K navrhované podobě databázové aplikace jsem došel na základě znalostí, které jsem ve firmě získal během pracovního procesu a postupným typizováním současné databáze. Vlastní návrh řešení vznikl také na základě zpětné analýzy při správě této databáze.

Jako první jsem hledal nový model pro uložení dat, který by nahradil ten současný, kde se o správu dat stará sám MS Access. Jak bylo popsáno už v teorii, nejlepší volbou je správa přes SQL Server, který se sám stará o bezpečnost a integritu dat a disponuje nástroji pro jejich snadnější správu. Další výhodou je možnost spravovat více databází najednou a tím lze efektivněji data strukturovat a analyzovat.

Tabulky databáze budou uloženy na SQL Serveru v interní síti firmy a instalace aplikace bude na klientských pracovních stanicích, jak znázorňuje obrázek. 3.



Obr. 3 – Uložení dat

Následující obrázek znázorňuje již celý uvažovaný model databázové aplikace včetně instancí a metod, které budou třídy volat, pomocí příkazů uživatele přes příslušné grafické rozhraní.



## 4.5 Logický model

Vychází z konceptuálního modelu, a jedná se o vyjádření vztahů a závislostí jednotlivých komponent, tříd nebo tabulek a jejich způsob komunikace mezi sebou pomocí volání svých nebo sdílených metod. Zároveň mohou poskytovat své instance jiným třídám.

Vzhledem k tomu, že se jedná o objektový model databáze, všechny součásti tedy i tabulky se zde budou chovat jako samostatné třídy objektů a vzájemně si předávat instance na základě volání metod nebo odkazů na ně.

Při programování všech vstupních hodnot je zde třeba také zmínit, jaké budou v databázi použity datové typy. V .Netu se obecně rozlišují dvě hlavní velké skupiny.

- a) **Hodnotový datový typ** – představuje skupinu všech proměnných, který již má přímo zadanou hodnotu, ať už se jedná o číselné typy, znaky a řetězce, datum nebo pravdivostní hodnotu `bool`.
- b) **Referenční datový typ** – hodnotami těchto proměnných jsou většinou pouze odkazy na hodnotu, která je uložena někde v paměti a odkazuje na nějaký jiný objekt.

V navrhované databázi budou použity všechny datové typy. Všechny veřejné hodnoty budou zadávány pomocí veřejné deklarace „`public`“, ostatní budou soukromé „`private`“.

## 4.6 Tvorba tabulek

Na základě modelování této aplikace, jsem rozhodl, že data se budou vkládat do několika hlavních tabulek, které budou dynamické. Vedle toho budou pro selektivní výběry použity tabulky statické, s hodnotami již přímo definovanými v aplikaci a s možností výběru přes selektor comboboxu. Jednotlivé záznamy tabulek se budou plnit hodnotami přes uživatelské rozhraní formulářů, které uživatel vyplní a poté stiskem tlačítka uloží do příslušné tabulky.

### 4.6.1 Statické tabulky pro selektivní výběry

Tabulka tiskový rastr, načítá své hodnoty ze souboru XML. Výběr možností je nastaven přes combobox a obsahuje všechny hodnoty, které se v tomto oboru využívají, takže není potřeba již doplňovat nové. Tyto definované hodnoty jsou: `133_kr`, `150_kr`, `175_kr`, `200_kr`, `250_kr`, `133_bo`, `150_bo`, `175_bo`, `200_bo`, `250_bo`, `300`.

Tab. 1 – Tiskový Rastr

Tiskový rast
133_kr
150_kr
175_kr

Stejně tak atribut vazba načítá své hodnoty ze souboru XML. Selektivní výběr zakázky bude obsahovat pouze tyto definované hodnoty: V1, V2, V3, V4, V5, V6, V7, V8, Plakát, Leták. Další typy vazeb už neexistují nebo je lze přiřadit k jednomu z výše uvedených.

Tab. 2 – Vazba

Vazba
V1
Plakát
Leták

Další atributem tvořený pouze jedinými definovanými hodnotami a umožňující výběr přes combobox je indikátor stavů zpracování, podle kterého se budou řídit záznamy při zobrazování v okně montáže nebo osvit. To, ve které části bude zakázka zobrazena, bude vázáno na zadanou hodnotu v poli ID vzkazu, viditelné bude pouze pole vzkazů. Má tyto definované hodnoty výběru: (1) Data email, staženo W, (2) Data FTP, staženo W, (3) Data storage, staženo W, (4) Nátisk, data W, (5) blok ok osvit, na obálku, potah, přebal chybí data, (6) obálka ok osvit, na blok, přílohu chybí data, (7) část dat ok osvit, (8) Schváleno, Ok osvit, (9) Plotry připravené k náhledu, (10) Nátisk vyhotoven a připraven k expedici, (11) chyba v datech, pošlou nové, (12) chybná technologie, upřesnit, (13) Reprint beze změn.

Tab. 3 – Tabulka stavů fáze zpracování

ID_vzkazu	Vzkazy
vk_01	data email, staženo W
vk_05	Ok Osvit
vk_13	Reprint beze změn

Combobox se selektivním výběrem budu v aplikaci využívat také k načítání propojených hodnot instancí z jiných tříd. Asi nejdůležitějším propojením bude všeobecná provázanost s tabulkou uživatelů, protože na základě hodnoty atributu „pozice“ budou zpřístupněny i určité části aplikace nebo funkce uživatelům.

#### 4.6.2 Dynamické Tabulky

Všechny dynamické tabulky budou měnit svůj obsah na základě přístupu uživatelů a jejich oprávnění k provádění změn. Tabulka Osvitové desky bude mít dva vstupní režimy pro zápis svých hodnot, první režim bude určen pouze pro vedoucího a tím je přidání nového formátu desky do systému. Druhý režim zápisu příjmu desek je povolen pro vedoucího a osvitáře, protože kompetence montážníka sem nespádají.

Tab. 4 – Osvitové desky

ID_deska	Název desky	Formát desky	Typ desky	Typ osvitovky	Počet	Zapsal
deska01	D660x530	660x530	Opti/N	osvitovka1	10	RB
deska02	D1030x785	1030x785	Fuji	osvitovka2	30	OK

Tabulka Uživatelé bude mít zásadní vliv na celou aplikaci, jednak se z ní porovnávají hesla při přihlášení, ale také do některých tabulek vkládají zkratku pracovníka ke zjištění zpětné autentizace záznamů. Hodnota pozice bude také určovat, které metody bude uživatel spouštět. Tuto tabulku uvidí v přehledu pouze administrátor systému a vedoucí oddělení. Jedině oni budou moci vkládat nové pracovníky do této tabulky.

Tab.5 – Uživatelé

Jméno	Příjmení	Heslo	Pozice	Tel. Kontakt	Email	Zkratka P.
Karel	Jindrů	xxx	osvitář	000 000 000	xxx@seznam.cz	KJ
Radek	Bejda	yyy	montážník	111 111 111	xxx@seznam.cz	RB
Jiří	Plachta	zzz	vedoucí	222 222 222	xxx@seznam.cz	JP

Záznamy o údržbě osvitových strojů budou zapisovat pouze osvitáři, neboť to spadá do jejich náplně práce. Montážník bude mít tuto možnost blokovanou.

Tab. 6 – Záznamy o údržbě strojů

ID_tech_de	Datum	Událost	Zapsal
tech_d01	4.2.2013	výměna lamp	KH
tech_d02	15.8.2015	kontrola stavu osvitovky	RB
tech_d03	23.4.2016	oprava raptoru	RH

Tabulka odběratelé bude poskytovat seznam kontaktů tiskáren nebo zákazníků. Bude provázána pomocí instancí „osvitová\_deska1, a osvitová\_deska2“ s tabulkou Osvitových desek, protože na formátu desky je odběratel závislý. Pravidlo pro zakládání bylo už popsáno v úvodu projektu.

Tab. 7 – Odběratelé, která má pole Firma a osvitové desky sdílené.

ID_Firma	Firma	Ulice,č.p.	Město	Osvitová_deska1	Osvitova_deska2	E-mail:
firma01	Tiskar1	Adresa	Praha 5	D660x530		la@seznam.cz
firma02	Tiskár2	Adresa	Praha 4	D660x530	D490x370	x@centrum.cz
firma03	Tiskár3	Adresa	Praha 3	D745x605		

Další instance „Firma“ z této tabulky včetně názvu osvitové desky bude provázána také s tabulkou zakázek. Její založení bude závislé právě na těchto instancích z tabulky odběratelů.

Tabulka zakázek nebude zobrazena vzhledem ke své rozsáhlosti, protože obsahuje velké množství instancí a přidávají se některé navíc, které do tabulky zapisuje osvitář, při spotřebě materiálu, viz obr. 10 a 12 v další části práce.

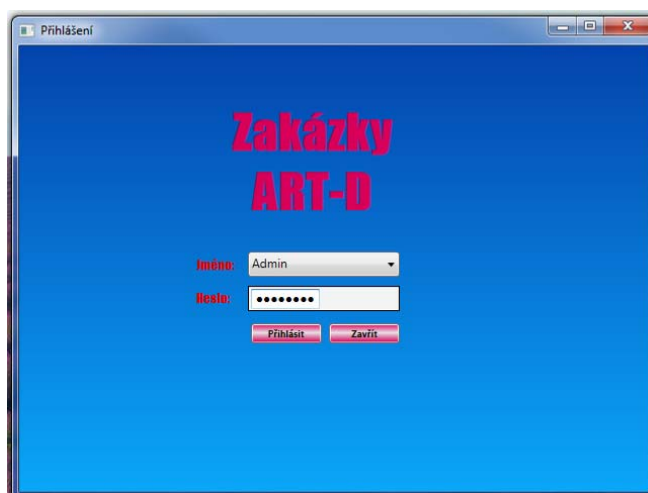
Přehledy, budou zobrazovat vybrané instance tabulek ke sledování změn a možnému editování.

#### **4.7 Formuláře a Dotazy**

Formuláře jsou tvořené pomocí Windows oken, ve kterých uživatel zadá vstupní hodnoty podle definovaného datového typu s ohledem na sledovanost položky a uloží do tabulky, nebo volbu zruší. Jak je znázorněno např. na obr. 4. Jediný formulář zakázek bude mít možnost navíc i tisku celého formuláře, který bude sloužit současně jako písemný podklad pro potřeby fakturace.

## 5 POUŽITÍ, A POHYB UŽIVATELE V APLIKACI

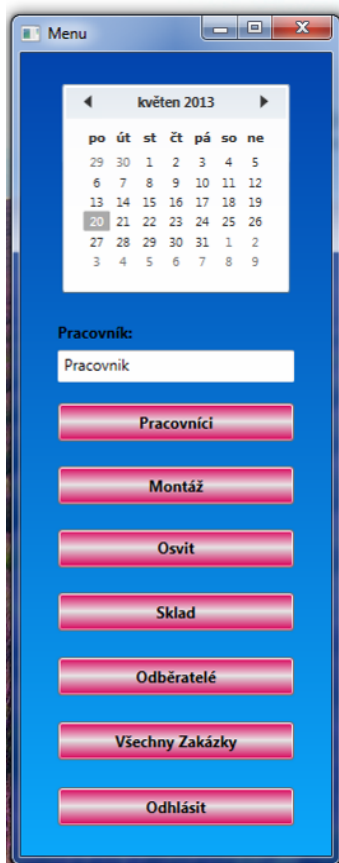
Aplikace se spustí souborem \*.exe a následně se objeví přihlašovací okno, viz obr.1. Po výběru registrovaného pracovníka a zadáním jeho hesla se poté klikne na tlačítko „Přihlásit“. Dojde k porovnání správnosti údajů s tabulkou uživatelé a potom následuje načtení zdrojových dat databáze připojením k SQL serveru. V případě chybného zadání hesla aplikace zobrazí upozornění o chybně zadaném hesle a bude čekat na zadání nového.



Obr. 5 – Okno Přihlášení

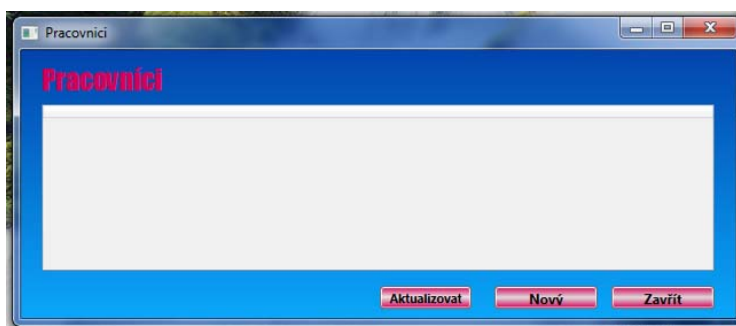
Po přihlášení do aplikace se otevře hlavní menu aplikace. Nahoře je aktuální systémový kalendář a pod ním je zobrazeno příjmení aktuálně přihlášeného uživatele.

Z Menu, které slouží hlavně jako navigační panel, se dále lze dostat do všech zbývajících částí aplikace pomocí jednotlivých tlačítek. Ty po stisku otvírají jednotlivé podokna formulářů s přehledy dat.



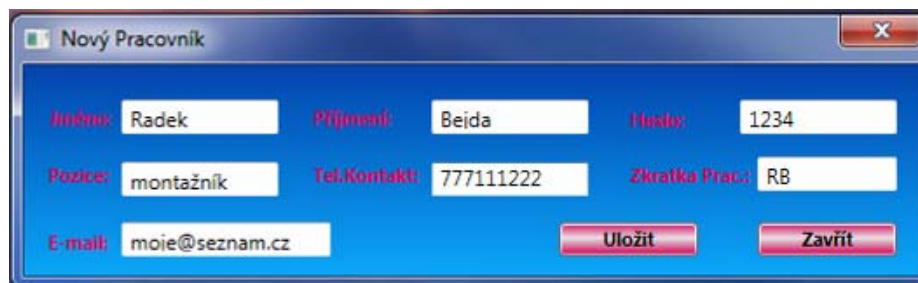
Obr 6 – Hlavní Nabídka

Tlačítko pracovníci je určeno pouze pro vedoucího, pro ostatní bude zablokováno s upozorněním, jak již bylo zmíněno dříve a slouží k zobrazení tabulky uživatelů a jejich identifikačních údajů a vypadá takto:



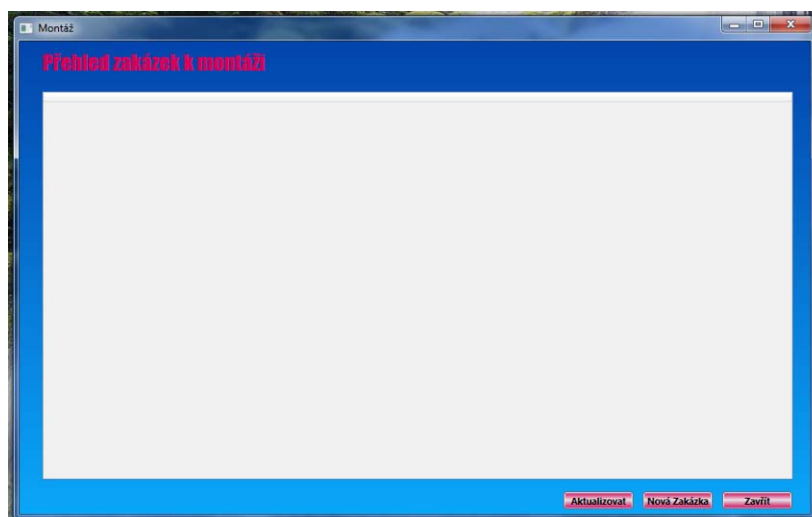
Obr. 7 – Přehled registrovaných uživatelů

Zde přes tlačítko „Nový“ se otevře formulář, kde vedoucí zadá identifikační údaje nového pracovníka do systému viz obr. 2 a následným kliknutím na tlačítko „Uložit“ dojde k zápisu a aktualizaci tabulky uživatelů. Tlačítko „Zavřít“ zavírá okna bez uložení údajů do tabulky nebo aktualizace zobrazených dat.



Obr. 8 – Založení nového pracovníka

Tlačítko „**Montáž**“ otevírá formulář pro přehled zakázek, kde v datovém poli jsou následně zobrazeny tabulkovou strukturou všechny aktivní zakázky ke zpracování a je možné do nich vstupovat a dodatečně je editovat na základě stavů zpracování.



Obr. 9 – Přehled rozpracovaných zakázek

V tomto okně se také nachází sada tlačítek pro další navigaci v aplikaci. Tlačítko „**Aktualizovat**“ slouží k aktualizaci zobrazených dat. Protože budou zakázky současně tvořené i na dalších pracovních stanicích, bude podoba přehledu dynamická. Tlačítko „**Zavřít**“ toto okno zavře, ale bez aktualizace. A poslední tlačítko „**Nová Zakázka**“ otevírá formulář pro vložení dat k vytvoření nové zakázky.

Zakázku vyplníme podle zadaných instancí, nebo případně vybereme správný údaj pomocí comboboxu. Horní polovina údajů je určena pro montážníky a vedoucího, kteří budou mít povoleno zakázku založit a editovat všechny její hodnoty, osvitář bude pouze připisovat či opravovat údaje o spotřebě materiálu a změna údajů v horní části bude pro něj blokována. Data se budou přidávat jako další instance do tabulky zakázek.

Obr. 10 – Zakázkový formulář

Další tlačítkem v Menu je osvit. Ten otevírá podobné okno jako montáž, pouze s tím rozdílem, že se zde zobrazují zakázky, které jsou určeny pro osvit, ať již kompletní nebo částečný, na základě rozlišení daného vzkazu v zakázce. Zobrazené instance jsou částečně odlišné.

Obr. 11. – Přehled Osvitu a součtů spotřeby

V tomto okně bude možné přímo vstupovat do zakázek, za účelem zapsání pouze spotřeby desek, aby bylo možné spotřebu následně odečíst i ze skladu. Instance „Osvitová deska“ bude doplněna automaticky podle odběratele zakázky a výběr typu desky pouze takový, který je definován společně v tabulce přijatých desek včetně výběru osvitového stroje. Formulář pro zápis osvitovaných desek je na obrázku 12:



Obr. 12 – Zápis Osvitu desek

ID osvitu bude generováno s každým novým zápisem. Datum je tvořen selektivním prvkem pro výběr data, a hodina bude doplněna uživatelem stejně tak jako archy. Prvek „checker“ slouží k aktivaci a zápisu oprav včetně popisu chyby. Zkratka uživatele v zápisu je provedena automaticky na základě přihlášeného uživatele.

Dalším tlačítkem na hlavním panelu je sklad desek. Slouží pro přehled skladovaného materiálu a jeho příjmu. Vytvoření nové desky je opět povoleno pouze vedoucímu, a do skladu nemůže naopak desky přijímat montážník.



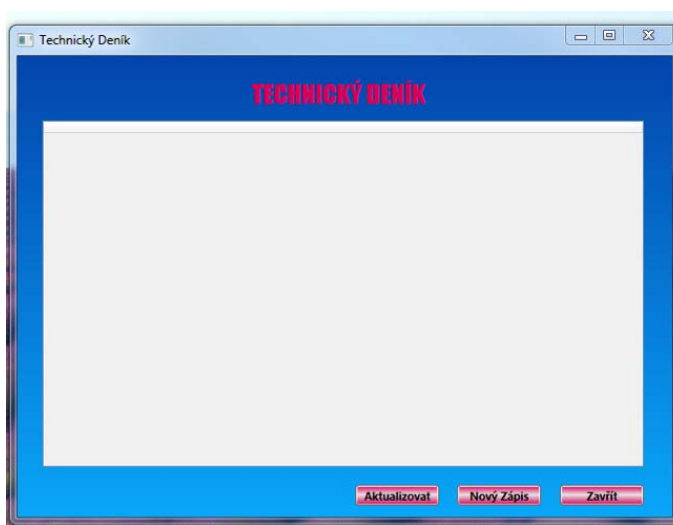
Obr. 13 – Sklad Desek

Oba způsoby zadávání desek do aplikace můžeme vidět na dalším obrázku se standardními tlačítky „Uložit“ a „Zrušit“. Oba tyto formuláře zapisují hodnoty do tabulky Osvitové desky.



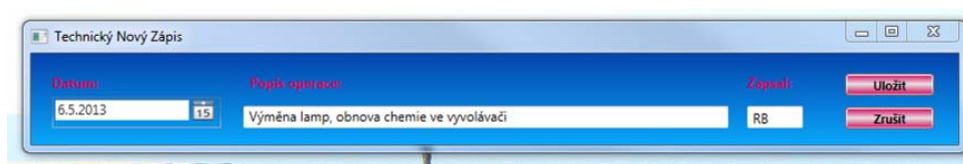
Obr. 14 – Příjem a přidání desek

Podokno technického deníku slouží osvitářům pro zápisy událostí vlastní údržby strojů a servisních techniků dodavatele. Do této sekce nebude mít přístup montážník.



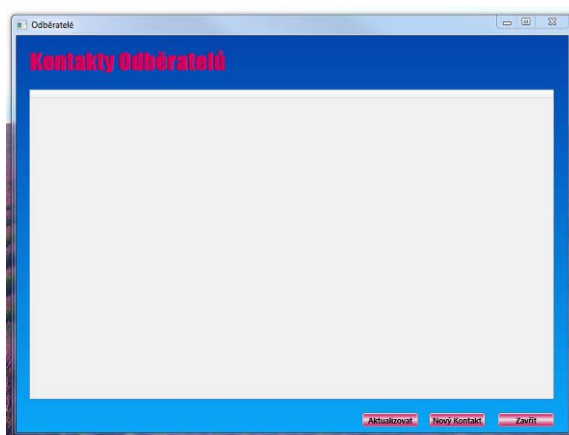
Obr. 15 – Přehled událostí technického deníku

Pro zápis nových událostí do technického deníku použijeme formulář obr. 16. Osvitář popíše událost stručně a jasně.



Obr. 16 – Zápis událostí

Předposledním navigačním tlačítkem na hlavním panelu jsou Odběratelé. Ty budou zobrazeny v podobném přehledovém okně, jako je montáž, osvit nebo všechny zakázky.



Obr. 17 – Odběratelé

Formulář pro vkládání nových kontaktů odběratelů je opět určený pouze pro vedoucího oddělení a hodnoty ukládá do tabulky odběratelů.

Obr. 18 – Formulář pro přidání nového kontaktu.

Propojené jsou zde osvitové desky, které se vybírají z tabulky Osvitových desek. Pokud nový zákazník má současně i nový typ desky je nutné nejdříve desku přidat do skladu a pak založit nového odběratele.

Poslední tlačítko „**Všechny Zakázky**“ slouží k zobrazení všech založených zakázek, tedy jak nedokončených tak dokončených a zde bude možné záznamy hromadně upravovat.

## ZÁVĚR

Bakalářská práce byla věnována nejen otázce uživatelských účtů a rolí v nové databázové aplikaci s připojením k SQL serveru, ale toto téma rozšířila o celkový náhled do informačních systémů. Nastínila současné možnosti jejich nebezpečí a zabezpečení i ochrany z několika různých pohledů. Snahou při tvorbě návrhu a vývoji projektu bylo vzít v úvahu současné požadavky osvitového oddělení na zabezpečení dat, ať už ze strany uživatelů nebo ze strany technické, a ohledem na zabezpečení proti poškození v síti. Do budoucna se tak otevřela otázka, zda nenajít společný dialog a uvažovat o sestavení kompletního návrhu na zabezpečený společný informační systému celé firmy, který by splňoval současné standardy, maximálně využil vlastních prostředků firmy při vynaložení minimálních nákladů.

## ZÁVĚR V ANGLIČTINĚ

This bachelor thesis was devoted not only to the issue of user accounts and roles in the new database application with connection to SQL Server, but this topic has expanded the overall insight into information systems. Outlined the current possibilities of risk and security and protection from several different perspectives. The effort in creating the design and development of the project was to consider the requirements of the platen separation of data security, whether by the user or by the technical, the security against damage to the network. In the future, thus, the question of whether or not to find a common dialogue and think and build a complete proposal to secure a joint information system, the company that would meet current standards, utilized the company's own funds when you insert a minimum cost.

**SEZNAM POUŽITÉ LITERATURY**

- [1] Ing. ŠENKEŘÍK, Roman. Ph.D. *Základy Informatiky – Presentace z přednášek*, zimní semestr 2010/2011, FAI UTB.
- [2] PROKOPOVÁ, Zdena. *Databázové systémy MYSQL + PHP*. Zlín: Univerzita Tomáše Bati ve Zlíně, 2006. ISBN 80-7318-486-9.
- [3] LIBERTY, Jesse. *Naučte se C++ za 21 dní*. Praha: Computer Press, 2002. ISBN 80-7226-774-4.
- [4] JILKOVÁ Helena – RYANT, Ivan. *Tvorba aplikací v objektovém prostředí*. Praha: Grada, 1994. ISBN 80-85623-82-X.
- [5] KRUGLINSKI, David, J. *Mistrovství ve Visual C++*. Praha: Computer Press, 1999. ISBN 80-7226-132-0.
- [6] KEPRT, Aleš. *Softwarová laboratoř v jazyce C#*. Olomouc: Přírodovědecká fakulta, Univerzita Palackého: Katedra informatiky, 2008 [online]. Dostupný z: <<http://phoenix.inf.upol.cz/esf/ucebni/>>
- [7] BĚHÁLEK, Marek. *Programovací jazyk C#*. [online] Dostupný z WWW: <<http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text/>>
- [8] *Všeobecná encyklopedie ve čtyřech svazcích: Encyklopedie Diderot*. Praha: Nakladatelský dům Op, 1998. ISBN 80-85841-17-7.
- [9] ČADA, Ondřej. *Objektové programování*. 1 vyd. Praha: Grada, 2009, 200 s. ISBN 978-80-247-2745-5
- [10] **Zákon č. 513/1991 Sb.**, obchodní zákoník, ve znění pozdějších předpisů, a další související zákony, Díl V (§ 17 – § 20), ze dne 5. 11. 1991. **Zákon č. 412/2005 Sb.**, o ochraně utajovaných informací a o bezpečnostní způsobilosti, ze dne 21. 9. 2005. **Zákon č. 101/2000 Sb.**, o ochraně osobních údajů a o změně některých zákonů, ze dne 4. 4. 2000. **Zákon č. 227/2000 Sb.**, o elektronickém podpisu a o změně některých dalších zákonů ze dne 29. 6. 2000. **Zákon č. 480/2004 Sb.**, o některých službách informační společnosti a o změně některých zákonů, ze dne 29. 7. 2004. **Mezinárodní norma ISO/IEC 27001:2005** – česká verze této normy je **ČSN ISO/IEC 27001**, vydaná v říjnu 2006; Česká technická norma (36 9790): *Informační technologie – Bezpečnostní techniky – Systémy managementu*

bezpečnosti informací – Požadavky; překlad zajištěn Českým normalizačním institutem (má stejný status jako oficiální verze). Mezinárodní norma **ISO/IEC 17799:2005**, zavedena v **ČSN ISO/IEC 17799:2006**; vydaná v srpnu 2006; Česká technická norma(36 9790); Informační technologie – Bezpečnostní techniky – Soubor postupů pro management bezpečnosti informací; překlad byl zajištěn Českým normalizačním institutem (má stejný status jako oficiální verze mezinárodní normy). **Britská norma BS 7799-2:2001**, má status české technické normy; **ČSN BS 7799-2**, vydaná v prosinci 2004, Česká technická norma (36 9790); Systém managementu bezpečnosti informací – Specifikace s návodem pro použití. Mezinárodní norma **ISO/IEC 27002:2005** – aktualizovaná verze normy **ISO/IEC 17799** a vydána v červenci 2007 pod označením **ISO/IEC 27002**; praktickým návodem pro implementaci jednotlivých opatření včetně doporučení pro jejich realizaci.

- [11] MSDN Knihovna: Visual Studio. [online] Dostupný z <[http://msdn.microsoft.com/cs-cz/library/52f3sw5c\(v=vs.100\).aspx](http://msdn.microsoft.com/cs-cz/library/52f3sw5c(v=vs.100).aspx)>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

BIT	Binary digit – dvojkové číslo 0 a 1.
C#	Objektový programovací jazyk, používaný při vývoji aplikací pod systémem MS Windows.
C++	Objektový programovací jazyk, na jehož základě syntaxe funguje i C#.
CLR	Common language runtime – běhové prostředí pod systémem Windows.
IBM	Americká firma zabývající se vývojem počítačů.
ISO	Mezinárodní organizace pro normalizaci
MS ACCESS	Program z kancelářského balíku MS Office, určený pro tvorbu jednoduchých nebo malých databázových aplikací.
.NET	Platforma firmy Microsoft, sloužící k vytváření aplikací pod systémy Windows.
PC 32,64 bit	Počítače s 32 nebo 64 bitovou architekturou.
PHP	Hyper Preprocessor, skriptovací jazyk pro webové aplikace.
SQL	Structured Query Language, dotazovací jazyk používaný k pořizování dat z databáze a jejich úpravě.
SŘBD	Systém báze řízení dat – představuje kompletní model databází.
Visual Studio	Vývojový software pro tvorbu nových aplikací pod systémem Windows.
XML	Označovací jazyk, který definuje soubor pravidel pro kódování dokumentu ve formátu, který je čitelný pro lidi i stroje.

**SEZNAM OBRÁZKŮ**

Obr. 1 – Schéma řízení přístupu .....	27
Obr. 2 – Návrhový model .....	31
Obr. 3 – Uložení dat.....	34
Obr. 4 – Konceptuální model návrhu databáze.....	35
Obr. 5 – Okno Přihlášení .....	41
Obr. 6 – Hlavní Nabídka.....	42
Obr. 7 – Přehled registrovaných uživatelů .....	42
Obr. 8 – Založení nového pracovníka.....	43
Obr. 9 – Přehled rozpracovaných zakázek .....	43
Obr. 10 – Zakázkový formulář .....	44
Obr. 11. – Přehled Osvitu a součtů spotřeby .....	44
Obr. 12 – Zápis Osvitu desek.....	45
Obr. 13 – Sklad Desek .....	45
Obr. 14 – Příjem a přidání desek .....	46
Obr. 15 – Přehled událostí technického deníku .....	46
Obr. 16 – Zápis událostí.....	46
Obr. 17 – Odběratelé.....	47
Obr. 18 – Formulář pro přidání nového kontaktu.....	47

**SEZNAM TABULEK**

Tab. 1 – Tiskový Rastr.....	37
Tab. 2 – Vazba.....	37
Tab. 3 – Tabulka stavů fáze zpracování .....	37
Tab. 4 – Osvitové desky .....	38
Tab.5 – Uživatelé.....	38
Tab. 6 – Záznamy o údržbě strojů .....	39
Tab. 7 – Odběratelé, která má pole Firma a osvitové desky sdílené. ....	39

## SEZNAM PŘÍLOH

CD S KÓDY A APLIKACÍ + ELEKTRONICKÁ PODOBA BAKALÁŘSKÉ PRÁCE

## **PŘÍLOHA P I:**