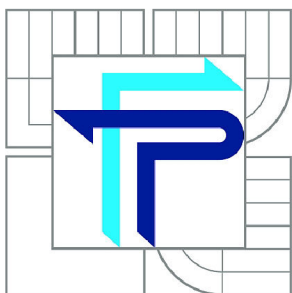


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH SYSTÉMU PRO SPRÁVU KLIENTŮ SPOLEČNOSTI TANDEM CRC CZ, S.R.O.

DESIGN OF SYSTEM FOR MANAGING THE COMPANY'S CLIENTS FOR TANDEM CRC CZ,
S.R.O.

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ DLOUHÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN LUHAN, Ph.D.

BRNO 2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Dlouhý Ondřej

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh systému pro správu klientů společnosti Tandem CRC CZ, s.r.o.

v anglickém jazyce:

Design of System for Managing the Company's Clients for Tandem CRC CZ, s.r.o.

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení a jejich přínos

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

BUREŠ, I. 10 zlatých pravidel péče o zákazníka aneb CRM v digitálním věku. 2. vyd. Praha: Management Press, 2006. 158 s. ISBN 80-7261-149-6.

GILMORE, W. J. Velká kniha PHP 5 a MySQL: Kompendium znalostí pro začátečníky i profesionály. 3. vyd. Brno: Zoner Press, 2011. 736 s. ISBN 978-80-7413-163-9.

GUTMANS, A. Mistrovství v PHP 5. 2. vyd. Brno: Computer Press, 2007. 655 s. ISBN 978-80-251-1519-0.

CHLEBOVSKÝ, V. CRM: Řízení vztahů se zákazníky. 1. vyd. Brno: Computer Press, 2005. 190 s. ISBN 80-251-0798-1.

LEHTINEN, J. Aktivní CRM: Řízení vztahů se zákazníky. 1. vyd. Praha: Grada, 2007. 158 s. ISBN 978-80-247-1814-9.

Vedoucí bakalářské práce: Ing. Jan Luhan, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2012/2013.

L.S.

doc. RNDr. Bedřich Půža, CSc.
Ředitel ústavu

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
Děkan fakulty

V Brně, dne 31.05.2013

Abstrakt

Bakalářská práce je zaměřena na návrh systému pro správu vztahu se zákazníky pro společnost Tandem CRC CZ s.r.o. Analyzuje konkrétní potřeby společnosti a na základě jejich požadavků je vytvořen návrh systému.

Abstract

This bachelor thesis focuses on design of system for managing clients of Tandem CRC CZ company. It analyzes specific requirements of the company and according to them proposes design of system.

Klíčová slova

vztah se zákazníky, databáze, e-business, e-managment, CRM, funkční modelování, datové modelování

Key words

Managing clients, e-business, databases, e-managment, CRM, data modelling, functional model.

Bibliografická citace mé práce

DLOUHÝ, O. Návrh systému pro správu klientů společnosti Tandem CRC CZ, s.r.o.. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2013. 55 s. Vedoucí bakalářské práce Ing. Jan Luhan, Ph.D..

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 31. května 2013

.....

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu práce, Ing. Janu Luhanovi, PhD, za pomoc a odborné vedení při zpracování této práce. Dále bych rád poděkoval své partnerce Evě a celé své rodině za podporu během studia. Děkuji!

Obsah

| | |
|---|----|
| Úvod | 11 |
| Cíle a metodika práce | 12 |
| 1 Teoretická východiska | 13 |
| 1.1 Customer Relationship Management (CRM) | 13 |
| 1.1.1 Co je to CRM | 13 |
| 1.1.2 Vývoj chápání CRM..... | 13 |
| 1.1.3 Cíle CRM | 14 |
| 1.1.4 Varianty / druhy CRM..... | 14 |
| 1.1.5 Provázanost CRM s databázemi (a datovými sklady)..... | 15 |
| 1.2 Softwarové inženýrství | 16 |
| 1.2.1 Pojem softwarového inženýrství | 16 |
| 1.2.2 Stručná historie a důvody vzniku | 16 |
| 1.2.3 Cíle softwarového inženýrství | 17 |
| 1.2.4. Softwarový proces | 17 |
| 1.3 Databáze | 18 |
| 1.4 Datové modelování | 18 |
| 1.5 Datový modely | 19 |
| 1.5.1 Lineární | 19 |
| 1.5.2. Hierarchický | 20 |
| 1.5.3. Síťový | 20 |
| 1.5.4 Relační model..... | 21 |
| 1.5.5 Objektový model | 22 |
| 1.5.6 Objektově relační model | 22 |
| 1.6 Relační datový model | 23 |
| 1.6.1 Relační datová struktura..... | 23 |

| | | |
|--------|---|----|
| 1.6.2 | Vlastnosti relačních tabulek | 23 |
| 1.6.3 | Pravidla relační databáze..... | 23 |
| 1.6.4 | Klíče relace..... | 25 |
| 1.6.5 | Integrita relačního modelu | 26 |
| 1.6.6 | Normalizace | 27 |
| 1.7 | Entitně-relační modelování | 29 |
| 1.8 | Funkční modelování | 30 |
| 1.8.1 | Vývojový diagram..... | 30 |
| 1.8.2. | Diagram datových toků | 31 |
| 1.9 | Technologie předpokládané pro implementaci..... | 32 |
| 1.9.1 | Databáze MySQL | 32 |
| 1.9.2 | Jazyk SQL | 32 |
| 1.9.3. | Jazyk PHP | 33 |
| 2 | Analýza a popis problému | 34 |
| 2.1 | Základní informace o společnosti | 34 |
| 2.2 | Stručný popis společnosti | 34 |
| 2.2 | Popis současné situace..... | 35 |
| 2.2.1 | Popis vztahu se zákazníkem..... | 35 |
| 2.3 | Analýza problému..... | 37 |
| 2.3.1 | Evidence klientů..... | 37 |
| 2.3.3 | Nahrávání souborů | 38 |
| 3 | Vlastní návrhy řešení | 39 |
| 3.1 | Konceptuální model..... | 39 |
| 3.1.1 | Identifikace entit..... | 39 |
| 3.1.2 | Identifikace relací..... | 39 |
| 3.1.3 | Identifikace atributů | 40 |

| | | |
|-------|--|----|
| 3.2 | Logický model | 41 |
| 3.3 | Popis relací, atributů a vztahů v logickém modelu..... | 42 |
| 3.4 | DFD - Funkční model..... | 45 |
| 3.5 | Popis nejdůležitějších procesů..... | 45 |
| 3.5.1 | Vytvoření záznamu zákazníka | 45 |
| 3.5.2 | Přidání události..... | 46 |
| 3.5.3 | Nahrání souboru | 48 |
| 3.6 | Skladba aplikace..... | 49 |
| 3.6.1 | Modul zákazník..... | 49 |
| 3.6.2 | Modul správa..... | 50 |
| 4 | Závěr | 51 |
| | Seznam použitých zdrojů | 52 |
| | Seznam obrázků. | 54 |
| | Seznam tabulek | 54 |
| | Seznam příloh..... | 55 |

Úvod

Vzhledem k rostoucí hustotě informačních toků v celé lidské společnosti se logicky tento nárůst projevuje i v obchodních vztazích. Více než kdy dříve je tedy potřeba, aby byly údaje o obchodních partnerech centrálně ukládány, zpracovány a v případě potřeby rychle vyvolány.

Metodiky správy těchto vztahů jsou často klíčové pro budování úspěšných obchodních partnerství – vzhledem k množství podkladů generovaných při použití umožňuje efektivní komunikaci i s větším množstvím klientů.

Ve své práci jsem se rozhodl zpracovat návrh systému pro správu zákazníku pro společnost Tandem CRC cz s.r.o., která se zabývá dodáváním interiérů prodejen – převážně regálových systémů.

Vzhledem ke své infromatické profilaci jsem se rozhodl zanalyzovat potřeby pro vztah se zákazníky ve společnosti Tandem CRC CZ, s.r.o. a navrhnout systém, který by tyto vztahy zachycoval a umožňoval by s nimi dále pracovat.

Poměrně častým termínem, který je často spojován s databázemi klientů je pojem CRM – Customer Relation Managment (správa vztahu s klienty). Tato metodika byla dříve spojovaná především s velkými společnostmi, ale v poslední době se principy přejímají i do menších společností. Dá se předpokládat, že do budoucna se tato metodika rozšíří ještě více.

Cíle a metodika práce

Vzhledem k faktu, že v současné době absentuje ve společnosti Tandem CRC cz s.r.o. ucelená metodika pro správu klientů, bude provedena analýza firemních procesů zaměřené na klienty – tedy především obchodní kontakt, akvizice a reporting.

V této počáteční analýze bude věnována zvláštní pozornost procesů styku se zákazníkem (zákaznickým procesům – navázání kontaktu, zaznamenání komunikace).

Na základě analýzy se provede návrh samotného systému, který by měl výsledky analýzy reflektovat. Tento návrh bude tedy východiskem pro následnou implementaci.

Jak jsem již zmínil, výsledkem bakalářské práce tedy bude **analýza a návrh systému** (se zaměřením na datový model) ve stavu, který umožní případnou implementaci. Zvýšený důraz bude věnován návrhu databáze (fázím konceptuálního a logického návrhu).

V analýze a v návrhu budou zohledněny některé části metodiky CRM.

Pro následnou implementaci se předpokládá použití skriptovacího jazyka PHP, relační databáze MySQL . Samotný systém bude provozován on-line s využitím protokolu HTTP.

Mým cílem je tedy vytvořit návrh systému správy klientů pro malou společnost, který bude pokrývat její individuální potřeby.

1 Teoretická východiska

1.1 Customer Relationship Management (CRM)

1.1.1 Co je to CRM

Pod anglickou zkratkou CRM se rozumí Customer Relationship Management (řízení vztahu se zákazníky) – soubor praktik a postupů sloužící k optimalizaci vztahu s klienty většinou s podporou počítačových nástrojů (CRM nástroje). Systém zpracovává různorodé zákaznické údaje (například jejich kontakty, počty zaměstnanců) a především obchodní procesy s nimi a dosahované tržby. Tyto systémy se používají ke sledování a vyhodnocování – jejich součástí je většinou i statistický výstup. Hlavním záměrem používání CRM je zlepšování služeb, analyzování potřeby zákazníků za účelem budování dlouhodobého synergetického vztahu se zákazníky. Tento optimalizovaný vztah se pozitivně projevuje na především na dlouhodobé prosperitě podniku. Je poměrně logické, že CRM není pouze systémový rámeček (např. v podobě software), ale opravdu celá škála praktik v oblasti vytěžování dat, komunikace, propagace, obchodu, statistiky, předprodejního a poprodejního servisu.

Obšírnost konceptu CRM lze dobře demonstrovat na prvcích, které jsou považovány za základní pilíře CRM¹:

- Lidé – jejich aktivní participace je nezbytná.
- Procesy - procesy, které zefektivní CRM – jedná se tedy o inovaci procesů v marketingu, prodeji a službách.
- Technologie – nástroje, které umožňují moderní řízení kontaktu.
- Data – převážně jejich sběr a jejich uchování.

1.1.2 Vývoj chápání CRM

Přestože termín Customer Relationship management se začal více používat až od 90. let tak jeho počátky můžeme sledovat do doby dřívější. V literatuře můžeme najít rozdělení na 5 vln / stádií CRM.

¹ Chlebovský, V. CRM: Řízení vztahu se zákazníky, s. 23-24

- Takzvaná první vlna proběhla v letech 1980 – 1990, kdy docházelo k **automatizaci služeb pro zákazníky** – především zavádění call-center a rozvoji telekomunikačních systémů
- 2. stádium se datuje k období mezi roky 1990-1995, kdy dochází k **tvorbě centralizovaných systémů pro řízení dat za účelem porozumět zákazníkům.**
- Ve třetí fázi (1995-2000) vrcholila snaha (na základě informací o porozumění) tvořit **hodnoty pro zákazníky,**
- Čtvrtá fáze (2000-2005) již pokrývá **vytvoření specializovaného managmentu,** zaměřeného na vztah se zákazníky
- Pátá fáze (2005 – současnost) nyní již dochází k vytváření **společnosti, zaměřené na zákazníky.**

Je pochopitelné, že jednotlivé vlny se prolínaly a postupy zavedené v dřívějších vlnách částečně přetrvaly.²

1.1.3 Cíle CRM

Obecně se uvádí, že smyslem CRM je změna celého paradigmatu přístupu k zákazníkovi. Jedná se především o změnu koncepce z uzavírání jednotlivých kontraktů (transakční přístup) na **dlouhodobý, perspektivní vztah se zákazníkem** (relační přístup). Taktéž se snaží o maximálně věnování pozornosti **zákaznickových potřeb.**³

1.1.4 Varianty / druhy CRM

² Lehtien, J. Aktivní CRM: řízení vztahů se zákazníky, 20

³ Chlebovský, V. CRM: Řízení vztahu se zákazníky, s. 23

Vzhledem i k poměrně volné definici CRM lze použít mnoha taxonomií pro klasifikaci tohoto druhu systémů. I zde samozřejmě je možný a pravděpodobný překryv jednotlivých činností CRM v rámci globálního systémů.

Operativní CRM

Slouží především pro přímou podporu prodeje. Smyslem je uchovávat veškerou komunikaci se zákazníkem (např. při prodeji a její další zpřístupnění pro další procesy (např. jako podklad pro další marketingovou kampaň).⁴

Analytické CRM

Zde již jedná o vytěžování dat v pravém slova smyslu – za použití statistických či ekonomických metod zde můžeme měřit hodnotu zákazníka, segmentaci zákazníků, analýzu kanálů atp. Slouží jako podklad pro strategické rozhodování⁵

Kolaborativní CRM

Je do jisté míry kombinací operativního a analytického CRM. Jedná se o nejmladší druh CRM a slouží především k poskytování komplexních informací o zákazníkovi napříč celou firmou. Tedy například prodejce při opětovném oslovení klienta by měl mít k dispozici veškeré jednání daného klienta s např. reklamačním oddělením společnosti⁶

1.1.5 Provázanost CRM s databázemi (a datovými sklady)

Vzhledem k tomu že pro efektivní používání analytického CRM (a následně tedy i kolaborativního CRM) potřebujeme mít data utříděna (ve smyslu organizačním i statistickým) a uložena. Za předpokladu splnění této podmínky nám tedy vzniká databáze, která pak může pomoci vytvořit i datový sklad. Tento datový sklad nám umožňuje s případnou minimální změnou i využívat Business Intelligence (např. OLAP

⁴ DOHNAL, J. Řízení vztahů se zákazníky: procesy, pracovníci, technologie. 2000, s. 60

⁵ Tamtéž, s. 63

⁶ Tamtéž, s. 63

kostku) a další analytické nástroje. Je tedy nanejvýše vhodné, aby se data uspořádávala v co nejpomyslnějších strukturách.⁷

1.2 Softwarové inženýrství

1.2.1 Pojem softwarového inženýrství

Softwarové inženýrství je disciplína, která se zaměřuje na veškeré aspekty tvorby počítačového vybavení. Těmito aspekty jsou především **vývoj, nasazení a údržba softwaru**.

Softwarové inženýrství je součástí (podmnožinou) odvětví nazývaného **systemové inženýrství**, které se zabývá především komplexními systémy.

Vzhledem k značné šíři odvětví se pro účely této práce se budu především zabývat problematikou správného navrhování a modelování počítačového softwaru.

1.2.2 Stručná historie a důvody vzniku

Vznik této inženýrské vědy je datován v 60. letech v reakci na tzv. „softwarovou krizi“, která vznikla v následku ohromného nástupu výpočetní techniky, kdy docházelo ke snižování cen hardwarového vybavení, ale cena softwaru exponenciálně rostla. Samotné programové vybavení však bylo často velmi nevykonné a funkčnost byla často omezená.⁸

Obecně se předpokládá, že tento jev byl způsoben především změnami způsobenými přechodem od jednoúčelových strojů k vysoce výkonným, obecně použitelným, mainframům, které již technicky umožňovaly vytváření velkých a obsáhlých systémů. Brzy nastalo zjištění, že dosavadní přístupy k vytváření softwaru nejsou již nadále aplikovatelné, a je potřeba je změnit.

⁷ HABANĚ, J., SODOMKA, P. Efektivní tvorba a provoz datových skladů [online]. 2003 [cit. 2013-04-12]. Dostupné z: <http://si.vse.cz/archive/proceedings/2003/efektivni-tvorba-a-provoz-datovych-skladu.pdf>.

⁸ KIMBLE, Ch., The Software Crisis [online]. 2005 [cit. 2013-04-12]. Dostupné z http://www.chris-kimble.com/Courses/World_Med_MBA/Software_Crisis.html

V rámci této situace se v roce 1968 let vytvořil v souvislosti s konferencí NATO v Německu relativně volný pojem **softwarové inženýrství**.

Následující vývoj byl zaměřený na vyřešení problému softwarové krize – významný je především rozvoj strukturovaného programování. Taktéž se počíná významněji rozvíjet objektově-orientované programování (např. vznik jazyku SMALLTALK) ⁹

1.2.3 Cíle softwarového inženýrství

Cílem softwarového inženýrství je aby výsledný produkt měl veškeré následující vlastnosti¹⁰

- Čitelnost
- Správnost
- Spolehlivost
- Znovu použitelnost
- Rozšiřitelnost
- Flexibilita
- Efektivita

1.2.4. Softwarový proces

Je posloupností činností, které je zapotřebí postupně splnit, aby bylo možné správně vyvinout software. V současnosti existuje několik hlavních modelů činnosti, ale všechny obsahují následující souslednost obecných činností:

1. **Specifikace** (Specification) – určení budoucí funkcionality softwaru a identifikace možných omezení.

⁹ SOMMERVILLE, I. Software engineering history [online]. ©2008 [cit. 2013-04-12] Dostupné z <http://www.cs.st-andrews.ac.uk/~ifs/Books/SE9/Web/History/>

SOMMERVILLE, I. Software Engineering, 2008, s. 2

¹⁰WILLIAMS, K. Notes on Software Engineering Goals [online]. [2006]. [cit. 2013-04-12] Dostupné z <http://williams.comp.ncat.edu/segoals.htm>

2. Vývoj (Development)

- a. Analýza a návrh – zde je s ohledem na požadavky vytvořena analýza a návrh
 - b. Implementace – samotná implementace kódu software
3. **Kontrola** (Validation) – v této fázi dochází ke kontrole, zdali software splňuje požadavky zákazníka.
 4. **Rozvoj** (Evolution) – dochází k modifikaci softwaru dle měnících se požadavků zákazníka / uživatele

Vzhledem k faktu, že se v této práci zabývám pouze návrhem systému, jsou tedy pro mě klíčové následující činnosti: **specifikace + analýza a návrh.** ¹¹

1.3 Databáze

Databáze je datovou strukturou, která uchovává organizované informace. Obecně se však pod pojmem databáze většinou myslí i systém, který se stará o interakci s uživatelem a umožňuje definovat, vytvářet a udržovat databázi. Dále se tento systém stará o řízený přístup k databázi (tedy především autorizaci a autentizaci).

V technickém hledisku je tedy třeba odlišovat mezi databázemi (ve smyslu množiny uložených dat) a takzvaným Systém řízení báze dat (SŘBD), který se stará o práci s daty. ¹²

1.4 Datové modelování

Je specifickou částí návrhů v rámci softwarového inženýrství. Nejenom u databází se používá především tři různých schémat, která po sobě následují :

- **Konceptuální schéma** – slouží k „zachycení“ dat používaných organizací do podoby modelu. Nebereme přitom ohled na podrobnosti (fyzickou implementaci či podkladový model databáze).

¹¹ SOMMERVILLE, I. Software Engineering, 2008, s.64

¹² CONOLLY, T., C. BEGG a R. HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází.* 2009, s. 37

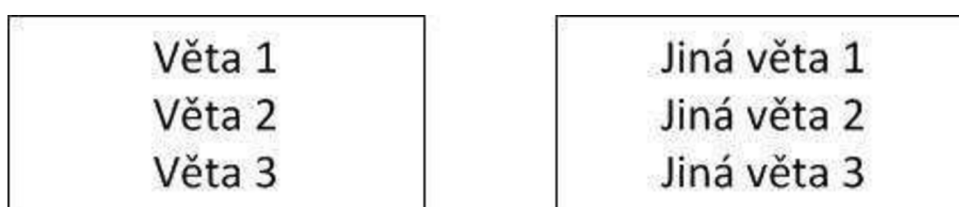
- **Logické schéma** - model dat vytvořených v tomto schématu je již postaven na určitém modelu databáze (například relační model), ale nezabývá se fyzickou implementací v konkrétním ŠRBD.
- **Fyzické schéma** – tento proces slouží k vytvoření popisu implementace databáze, popisuje tabulky, organizaci souborů, indexy, integritní a bezpečnostní omezení.¹³

1.5 Datový modely

Na data a datové modely je možné se dívat z více pohledů. Jeden z pohledů se zabývá způsobem uložení sbírek dat, realizace vztahů mezi daty a omezení vlastních dat. V tom případě mluvíme o datovém modelu uložení dat¹⁴

1.5.1 Lineární

Jedná se o datový model, kdy jsou záznamy uloženy ve formě jednotlivých souborů, ve kterých jsou uloženy datové věty o pevně dané struktuře. Tyto soubory nemají žádnou vazbu mezi sebou. O vztazích mezi samotnými větami v souboru nemáme žádné informace s výjimkou, že víme informace o předcházející větě a o větě následující. Tento typ si můžeme představit například jako papírovou kartotéku v knihovně (s knihovnickými lístky). V současné době se již téměř nepoužívá.¹⁵



Obrázek 1: Lineární datový model¹⁶

¹³ CONOLLY, T., C. BEGG a R. HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. 2009, s. 206-207

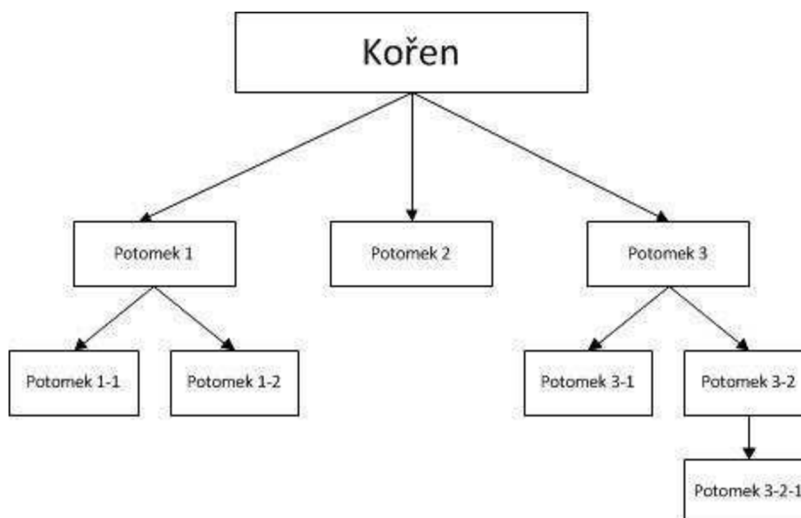
¹⁴ Tentýž, s. 62-63

¹⁵ KOCH M., B. NEUWIRTH. *Datové a funkční modelování*. 2010, s. 20-21

¹⁶ Zdroj: vlastní zpracování

1.5.2. Hierarchický

Tento model je postavený na stromové hierarchii, kdy kromě předchůdce a následovníka známe též vztah rodin-potomek. Struktura je postavena na tzv. „kořenu“, který je hlavním rodičem celého systému.



Obrázek 2 : Hierarchický datový model¹⁷

Problémem tohoto vztahu je takový že ač jeden rodič může mít mnoho potomků, jeden potomek může mít pouze jednoho rodiče. Model tedy umožňuje pouze vztah 1:N, nikoliv N:M.¹⁸

Model má uplatnění především v popisu situací, které mají přirozenou strukturu právě stromově seřazenou.

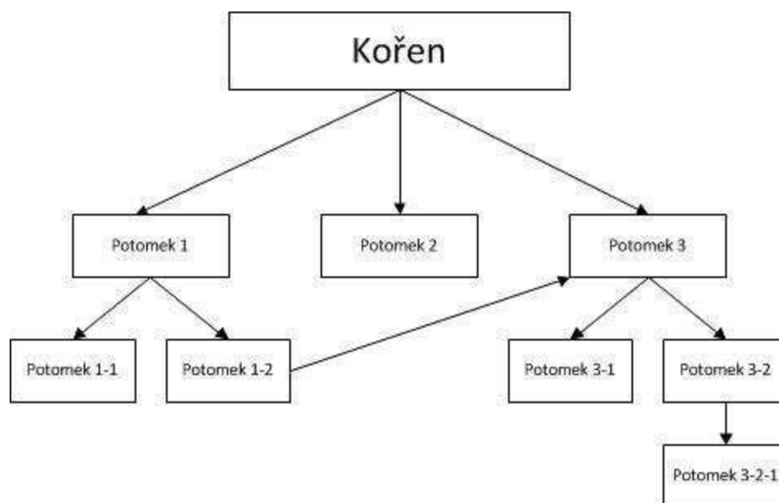
1.5.3. Síťový

Vzhledem k omezením hierarchického modelu byl vytvořen model síťový, který umožňuje realizovat vztahy N:M (byť u složitějších vztahů to obnáší vyšší režii systému). Vzhledem k charakteru modelu začít hledání v podstatě v kterémkoliv uzlu.

¹⁷ Zdroj: vlastní zpracování

¹⁸ KOCAN M., *Databáze nejsou jen relační, díl druhý* – HDM.[online] 2001. [cit. 2013-04-12]. Dostupné z : <http://www.dbsvet.cz/view.php?cisloclanku=2001112601>

Vztahy mezi položkami jsou zajišťovány ukazateli (pointery). Kořenový bod, zde již být nemusí (ale často bývá)¹⁹



Obrázek 3: Sít'ový datový model²⁰

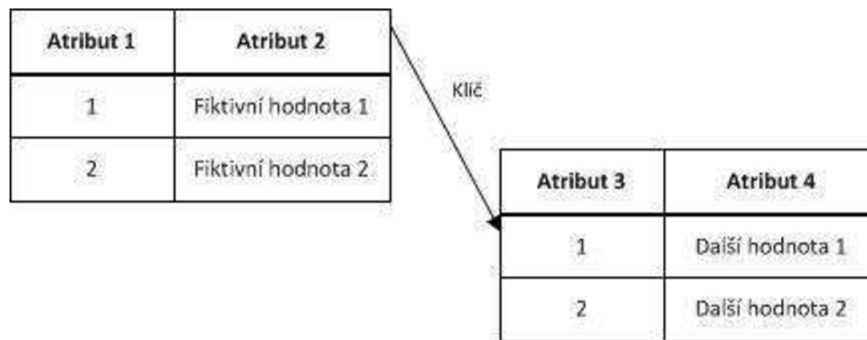
1.5.4 Relační model

Jedná se o v současnosti nejrozšířenější datový model, který je postaven na teorii relací. Jeho počátky sahají do konce 60. let a vznik především díky článku Edgara F. Codd – A Relational Model of Data for Large Shared Data Banks (Relační model dat pro velké sdílené banky dat) z roku 1969. Celá koncepce relačního modelu vychází z matematického pojetí teorie relačních množin. Umožňuje zachycovat nejen samotná data, ale i vztahy mezi nimi. Jedná se v současnosti o nejrozšířenější datový model, který se od konce 70. let se stal de facto standardem pro databázová řešení. Tento model bude ještě podrobněji rozveden. Vztahy jsou zachyceny pomocí tzv. relačních klíčů.²¹

¹⁹ KOCAN M., *Databáze nejsou jen relační, díl třetí – SDM*. [online] 2001. [cit. 2013-04-12]. Dostupné z: <http://www.dbsvet.cz/view.php?cisloclanku=2001120303>

²⁰ Zdroj: vlastní zpracování.

²¹ KOCH M., B. NEUWIRTH. *Datové a funkční modelování*. 2010, s. 23



Obrázek 4: relační datová model²²

1.5.5 Objektový model

Objektový model je modelem nejmodernějším – vychází z (a je mu velmi podobný) paradigmatu objektově-orientovaného programování. Entitu skutečného světa reprezentuje jako objekt, který má své vlastnosti a určité metody – dochází zde tedy k jisté kombinaci uložení dat a vykonatelného kódu. Vzhledem k této skutečnosti podporuje některé vlastnosti pro objektově přístup typické – například dědičnost, zapouzdření, polymorfismus, jednoznačné identifikátory objektů atp.²³

Problémem objektového modelu je však relativně vysoká výpočetní a prostorová náročnost, kdy především u jednoduchých záznamů nedosahovala příliš dobré efektivity. Taktéž pochopitelnost záznamů nebyla příliš dobrá. Taktéž již trh relačních databází byl již poměrně rozsáhlý a výrobci neměli velkou motivaci relační model opustit. Z tohoto důvodu se nakonec objektově orientovaný model příliš nerozšířil.

1.5.6 Objektově relační model

Snaha zkombinovat výhody objektového a relačního přístupu k datům umožnila vznik objektově-relačního modelu. V tomto modelu dochází k rozšíření relačního modelu o objekty a konstrukty, které umožňují manipulaci s novými datovými typy.²⁴

²² Zdroj: vlastní zpracování

²³ KOCH M., B. NEUWIRTH. *Datové a funkční modelování*. 2010, s. 21-22

²⁴ BATKO, M. *Relační vs. objektově-relační vs. objektové databáze* [online] 2002. [cit. 2013-04-12]. Dostupné z <http://www.fi.muni.cz/~xbatko/oracle/compare.html>

V současnosti jsou tyto přístupy podporovány například v databázích Oracle.

1.6 Relační datový model

1.6.1 Relační datová struktura

Relační model ukládá data v **relacích** (které jsou označeny jako tabulky). Pojmenované sloupce jsou **atributy**, dohromady na jedné tabulce tvoří **schéma relace**. Jednotlivé datové věty (někdy též záznamy) jsou zde označeny jako **datové n-tice** (řádek tabulky) – množina všech datových n-tic v dané relaci je **tělem relace**. Pojmem **doména** označujeme množinu možných hodnot, které mohou nastat v rámci jednotlivých atributů. Kolekce všech relací tvoří relační databázi.^{25 26}

1.6.2 Vlastnosti relačních tabulek

Relační tabulka by měla mít následujících 6 vlastností²⁷:

- Tabulka má v rámci příslušné databáze jedinečné jméno.
- Údaj v buňce tabulky obsahuje pouze jedno hodnotu a ta by již neměla být dále strukturovatelná (atomickou hodnoty).
- Každá sloupec (atribut) má jedinečné jméno v rámci tabulky.
- Pořadí řádků je irelevantní.
- Pořadí sloupců je irelevantní.
- Každý záznam je jedinečný.

1.6.3 Pravidla relační databáze

Tato pravidla byla zformulována (již se záměrem na systém řízení báze dat) samotný doktorem Coddem. Specifikují podmínky, které by měl plně relační databázový systém splňovat:

²⁵ KOCH M., B. NEUWIRTH. *Datové a funkční modelování*. 2010, s. 23

²⁶ CODD, E. F. A Relational Model of Data for Large Shared Data Banks [online]. 1970 [cit. 2013-04-12]. Dostupné z <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>

²⁷ CONOLLY, T., C. BEGG a R. HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. 2009, s. 65

1. Informační pravidlo
Všechny informace v relační databázi jsou vyjádřeny pouze jediným způsobem – jako hodnoty v tabulkách
2. Pravidlo jistoty
Všechna data musí být přístupná přes název tabulky, název sloupce a hodnotu primárního klíče
3. Systematické zpracování hodnoty „null“
SŘBD musí umožnit hodnotu údaje „null“ či prázdnou. Při interpretaci ji pak musí identifikovat jako chybějící či nepoužitelný údaj a rozlišit jí tak od jiných hodnot (například hodnoty 0).
4. Dynamický online katalog, který je založen na relačním modelu
Popis databáze (metadat) je vyjádřen na logické úrovni stejným způsobem jako obyčejná data, takže autorizovaný uživatel může aplikovat stejný relační jazyk ke svému dotazu jako uživatel při práci s daty.
5. Obsáhlý datový podjazyk
SŘBD musí mít implementovaný alespoň jeden jazyk, který bude schopný pokrýt celou funkcionalitu databáze.
6. Pravidlo aktualizace pohledů
Všechny pohledy, které lze aktualizovat, lze taktéž aktualizovat systémem.
7. Schopnost vkládání, vytváření a mazání
Práce se základní relací či odvozenou relací jako jediným operandem se musí aplikovat na vyhledávání, vložení či změnu dat.
8. Fyzická datová nezávislost
Činnost aplikací a terminálu nesmí být narušena při změně způsobu uložení dat či změně přístupu k nim.
9. Logická datová nezávislost
Činnost aplikací a terminálu nesmí být narušena při změně způsobu uchování informací v tabulkách.
10. Integritní nezávislost

Integritní omezení musí být definovaný přímo v podjazyku databáze, nikoliv v aplikační rovině.

11. Nezávislost distribuce

Datový manipulační podjazyk musí umožnit aplikacím a terminálu pracovat s daty umístěnými na jiném počítačovém systému.

12. Pravidlo nenarušení databáze

Pokud relační systém podporuje nízko-úrovňový jazyk tak musí zabránit tomu, aby byl tento jazyk použit k obejití či narušení integritních pravidel či omezeních, vyjádřených ve vyšším jazyce.²⁸

1.6.4 Klíče relace

Vzhledem k požadavku na jedinečnost a unikátnost záznamu je potřeba, aby bylo možno ho jednoznačně identifikovat. Tento požadavek je realizován takzvanými relačními klíči.

Známě několik typů relačních klíčů:

- Superklíč
 - Je tvořen atributem či více atributy, které umožňují jedinečnou identifikaci věty tabulky v relaci. V relaci jich může být více.
- Kandidátní klíč
 - Jedná se o superklíč, který je již neredukovatelný (nelze z něj již vypustit žádný z atributů, aniž by neztratil schopnost jednoznačně identifikovat větu tabulky). Taktéž jich může být v relaci více.
- Primární klíč
 - Je jeden, vybraný kandidátní klíč, který slouží k jednoznačnému určení záznamu v relaci.
- Cizí klíč
 - Slouží společně s primárním klíčem jiné relace k vytváření spojení mezi relacemi – umožňuje tedy zachycovat vztahy mezi tabulkami. Je podmíněn existencí stejné hodnoty primárního klíče v jiné relaci. Další

²⁸ GOMORI, S. Dr. E. F. Codd's 12 rules for defining a fully relational database [online]. 2005 [cit. 2013-04-12]. Dostupné z <http://www.cse.ohio-state.edu/~sgomori/570/coddsrules.html>

podmínkou je, aby doména hodnot pro cizí a primární klíč byla stejná. Pokud neexistuje adekvátní primární klíč tak je nastaven na hodnotu NULL²⁹

1.6.5 Integrita relačního modelu

Vzhledem k faktu, že v datovém modelu se snažíme co nejlíže zachytit reálné objekty a jejich vazby a vztahy mezi sebou, tak musíme reflektovat i jista omezení z důvodu konzistence.

1.6.5.1 Integritní omezení pro entity

1. Doménová omezení

- a. Vzhledem k tomu, že každý atribut by měl mít svoji doménu hodnot, tak musí do této množiny spadat veškeré hodnota položky věty v daném sloupci. Toto platí pro všechny n-tice v dané relaci.

2. Entitní integrita

- a. Žádný sloupec primárního klíče nesmí mít hodnotu NULL a musí být jedinečný (jak již plyne z definice primárního klíče).

3. Referenční integrita

- a. Při vytváření záznamu s cizím klíčem kontroluje, zdali existuje adekvátní primární klíč se stejnou hodnotou v patřičné relaci, na kterou cizí klíč odkazuje. Pokud neexistuje patřičný primární klíč, pak musí mít cizí klíč hodnotu NULL.

30 31

1.6.5.2 Integritní omezení pro vztahy entit

²⁹ CONOLLY, T., C. BEGG a R. HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. 2009, s. 66-67

³⁰ CONOLLY, T., C. BEGG a R. HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. 2009, s. 66-67

³¹ KOCH M., B. NEUWIRTH. *Datové a funkční modelování*. 2010, s. 27

S ohledem na **kardinalitu** vztahu (tedy počet k sobě odpovídajících n-tit) můžeme určit čtyři základní vztahy mezi prvky jednotlivých relací. Samozřejmě v situaci kdy mezi daty neexistuje spojitost, tak není potřeba definovat vztah.³²

- **Vztah 1:1** (jedna k jedné)
 - Označuje situaci kdy jedné datové větě přiřazena nanejvýš jedna věta v jiné relaci. Příkladem tohoto vztahu může být například vztah mezi člověkem a zbrojním průkazem. Jeden člověk může mít jeden nebo žádný zbrojní průkaz.
- **Vztah 1:N** (jedna ku více)
 - Označuje vztah, kdy jedné datové n-tici náleží alespoň jedna n-tice v jiné relaci. Tento vztah si můžeme představit například vztah mezi vlastníkem telefonního čísla a telefonním číslem. Jeden člověk může mít více telefonních čísel, ale telefonní číslo může mít jen jednoho vlastníka.
- **Vztah N:1** (více ku jedné)
 - Jedná se o stejný vztah jako v předchozím případě, pouze strany jsou opačné.
- **Vztah N:M** (více ku více)
 - Jedná se o vztah kdy obecně několik n-tic relace odpovídá jedna nebo více n-tic v jiné relaci. Tento vztah již nelze v relačním modelu realizovat přímo na těchto dvou entitách. Je zde tedy potřeba vytvořit novou entitu, jejíž primární klíč bude složen z primárních klíčů původních entit. Nová entita se nazývá vazební a celému procesu se přezdívá dekompozice vztahu. Příkladem může být například akademický titul – jeden člověk může mít více akademických titulů a daný titul může mít více lidí.

1.6.6 Normalizace

³² Tentýž, s. 30-34

Normalizace je jedním z důležitých technik, které se využívají při návrhu relačních databází. Slouží především k omezení redundance dat a omezení složitosti databáze při současném zachování závislostí. Dodržení minimálně prvních 3 normálních forem se velmi doporučuje z důvodu konzistence dat

Známe 5. Normálních forem (NF).

1. Normální forma

- a. „*Relace je v první normální formě, pokud jsou všechny její atributy definovány nad skalárními obory hodnot (doménami)*“³³ Tato normální forma požaduje, aby byly záznamy byly uchovávány v dále nedělitelné (atomické) formě.

2. Normální forma

- a. „*Relace je ve druhé normální formě, pokud je v první normální formě a navíc všechny její atributy jsou závislé na celém kandidátním (primárním klíči)*.“³⁴ Druhá normální forma se týká pouze primárních klíčů, které jsou složeny z více atributů. Pro splnění požadavku 2. normální formy je kromě splnění 1. NF nutná skutečnost, že veškeré atributy jsou nutně závislé na celém kandidátním / primárním klíči. Tedy vazby atributu na jednoznačnou identifikaci skrze celý klíč.

3. Normální forma / Boyce – Coddova normální forma

„*Relace je ve třetí normální formě, pokud je ve druhé normální formě a navíc všechny její neklíčové atributy jsou vzájemně nezávislé*“³⁵ Jedná se zde o vyloučení tzv. tranzitivní závislosti, tedy situace, kdy jsou atributy na primárním klíči závislé přes jiný atribut a nikoliv přímo.

Boyce – Coddova normální forma je specifickou variací třetí normální

³³ KOCH M., B. NEUWIRTH. *Datové a funkční modelování*. 2010, s. 55

³⁴ Tentýž, s. 56

³⁵ Tentýž, s. 58

formy pro situace kdy je více složených kandidátních klíčů, jejichž atributy se překrývají

4. Normální forma

- a. „Relace je ve čtvrté normální formě, pokud je v Boyce – Coddově normální formě, a navíc všechny vícehodnotové závislosti jsou zároveň funkčními závislostmi z kandidátních klíčů (v jedné relaci se nesmí spojovat nezávislé opakované skupiny.“³⁶

5. Normální forma

- a. „týká se případu spojené závislosti, která vyjadřuje cyklické omezení: pokud je relace1 spojena s relací2 , relace2 je spojena s relací 3 a relace3 je spojena zpětně s relací 1, pak všechny tři entity musí být součástí stejného vektoru hodnot.“³⁷

1.7 Entitně-relační modelování

ERM (Entity-relationship model) je jedním ze základních nástrojů pro datové modelování. Pomocí této metody je možné vytvářet Entity-relationship diagramy (ERD) nebo používat slovní popis. Jejich účelem je především znázornit data ontologickým způsobem – tedy zachytit jejich entity a vztahy mezi nimi.

Základními prvky jsou:

Entita – jedná se většinou o objekt reálného světa.

Vztah / Relace – Zachycuje vazby mezi více entitami.

Atribut – Určuje specifické charakteristiku dané entity či atributu.³⁸

³⁶ KOCH M., B. NEUWIRTH. *Datové a funkční modelování*. 2010, s. 61

³⁷ Tentýž, s.62

³⁸ CHEN, P. The Entity-Relationship Model – Toward a Unified View of Data. [online] 1970 [cit. 2013-04-12]. Dostupné z <http://www.csc.lsu.edu/news/erd.pdf>.


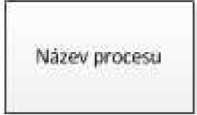

1.8 Funkční modelování

Předmětem funkčního modelování je především analýza a algoritmizace procesů, které v informačním systému probíhají.³⁹ Tedy na rozdíl od dat popisuje procesy, které zpracovávají data.

1.8.1 Vývojový diagram

Je poměrně starým způsobem znázorňování funkcí systému. Jeho aplikovatelnost je velmi široká. Umožňuje konstruovat rozhodovací stromy a znázornit strukturu programu. V níže uvedené tabulce uvádím některé základní značky, jejich název a popis.⁴⁰


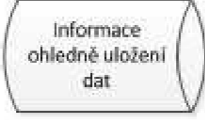
Tabulka 1 Popis vývojového diagramu⁴¹

| Symbol | Popis | Znázornění |
|-----------------------------|--|---|
| <u>Start / Konec</u> | Označuje začátek a konec algoritmu. |  |
| <u>Proces</u> | Provádí samostatnou dílčí část algoritmu. Může se vnitřně dělit do tzv. subprocessů, ve kterých můžeme proces opět dekomponovat na jednotlivé činnosti. |  |
| <u>Rozhodovací podmínka</u> | Znázorňuje podmínku, jejíž vyhodnocení může obecně nastat dvou stavů (podmínka splněna / nesplněna). Na základě tohoto se odvíjí jednotlivé následující sekvence událostí. |  |

³⁹ KOCH M., B. NEUWIRTH. *Datové a funkční modelování*. 2010, s. 76

⁴⁰Tentýž, s. 90

⁴¹ Zdroj: vlastní zpracování


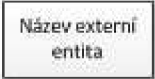
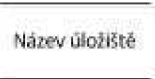
| | | |
|-------------------------------|--|---|
| <u>Vstup / výstup dat</u> | Může být realizováno například tabulkou databáze, datovým souborem atp. Slouží k uchování dat. |  |
| <u>Databáze / uložení dat</u> | Označuje uložení dat (většinou do databáze). |  |

1.8.2. Diagram datových toků

Taktéž znám jako Data Flow Diagram (DFD) Je jednou z nejpoužívanějších metod používaných ve funkčním modelování. Umožňuje zobrazit toky dat v jednotlivých částech systému.


V současnosti existuje několik konvencí (způsobu) zakreslení DFD diagramů. Já v práci použiji konvenci Yourdon and Coad. Její značení a popis jednotlivých symbolu je uveden v tabulce níže. ⁴²

Tabulka 2 Symboly DFD diagramu (konvence Yourdon and Coad) ⁴³

| Symbol | Popis | Znázornění |
|-----------------------|--|---|
| <u>Proces</u> | Jedná se o činitel systému, stará se o transformaci vstupních hodnot na výstupní. |  |
| <u>Externí entita</u> | Je prvkem v okolí systému, se kterým probíhá komunikace. Je zdrojem či naopak příjemce dat. |  |
| <u>Uložení dat</u> | Může být realizováno například tabulkou databáze, datovým souborem atp. Slouží k uchování dat. |  |

⁴² KOCH M., B. NEUWIRTH. *Datové a funkční modelování*. 2010, s. 84

⁴³ Zdroj: vlastní zpracování

| | | |
|-------------------|--|---|
| <u>Datový tok</u> | Znázorňuje tok dat v rámci systému – je reprezentací přechodných (temporálních) dat mezi jednotlivým zpracováním |  |
|-------------------|--|---|

1.9 Technologie předpokládané pro implementaci

Přestože implementace systému není součástí práce, považuji za důležité alespoň částečně uvést základní technologie, které jsou předpokládané pro implementaci.

1.9.1 Databáze MySQL

Patří mezi jednu z nejrozšířenějších databázových systémů. Obzvláště ve spojení se skriptovacím jazykem PHP tvoří typické programové prostředí pro vytváření aplikací menšího až středního rozsahu.

Jako jazyk pro komunikaci s databází slouží jazyk SQL. Jedná se o tzv. software s dvojitým licencováním – je licencována jak pod Open Source (GPL) tak i pod komerční licenci.⁴⁴

Tento databázový systém je považován za poměrně rychlý a s příchodem novějších úložných enginů (InnoDB) podporuje již i pokročilejší funkcionalitu (například transakce)

1.9.2 Jazyk SQL

SQL je zkratkou slov Structured Query Language (Strukturový dotazovací jazyk) a v současné době je považován za nejrozšířenější jazyk pro manipulaci s daty.

Tento neprocedurální jazyk je postaven nad relační algebrou. Při jeho vývoji byla velká pozornost věnována syntaxi z důvodu co největší blízkosti k přirozenému jazyku (angličtině).

⁴⁴ GUTMANS, A., S. Bakken, D. Rethans. Mistrovství v PHP 5. 2007, s. 166-167

V současnosti existuje několik implementací tohoto jazyka (především různými databázovými systémy).⁴⁵

1.9.3. Jazyk PHP

Skriptovací jazyk PHP je velmi rozšířeným zástupcem tzv. skriptovacích jazyků, které slouží k vykonávání požadovaných rutin na straně server. Tento jazyk od roku 1995 prošel výrazným vývojem – od jednoduché sady skriptů určené pro tvorbu jednoduchých dynamických webových stránek po moderní jazyk s podporou moderních programovacích prostředků (např. lambda funkce, closures, kvalitní OOP přístup), na kterém jsou budovány portály pro miliony uživatelů.⁴⁶

Poměrně typickým znakem jazyka jeho syntaktická jednoduchost a orientace na snadné pochopení i pro začátečníky. Jazyk je slabě typový – znamená to tedy, že není třeba explicitně definovat datový typ proměnných – jazyk sám vyhodnocuje jejich vhodný typ a převádí jednotlivé typy mezi sebou. Dále je potřeba zmínit že jazyk je zcela zdarma zveřejněn pod licencí Open Source.⁴⁷

⁴⁵ CONOLLY, T., C. BEGG a R. HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. 2009, s. 72-77

⁴⁶ GILMORE, W. *Velká kniha PHP 5 a MySQL: kompendium znalostí pro začátečníky i profesionály*. 2011, s. 28

⁴⁷ Tamtéž, s. 31 - 33

2 Analýza a popis problému

2.1 Základní informace o společnosti

Název společnosti: Tandem CRC CZ s.r.o.

Sídlo: Tomešova 2b, 602 00 Brno

Jednatel: Ing. Petr Špatný

Webová prezentace: <http://www.tandemcrc.cz>

Počet zaměstnanců: do 5 pracovníků

Datum zápisu do Obchodního rejstříku: 19. 7. 2003



Obrázek 5: Logo společnosti⁴⁸

2.2 Stručný popis společnosti

Společnost Tandem crc cz s.r.o. byla založena v roce 2003 za účelem expanze slovenské společnosti Tandem plus crc, s.r.o na území České republiky. Hlavním předmětem podnikání je prodej vybavení (interiérů) prodejen, především regálových systémů CRC a Guissani – tento prioritní sortiment doplňují regálové systémy dalších výrobců, které společnost v nedávné době přidala do svého portfolia.

Snahou firmy je přistupovat co nejvíce individuálně ke každému klientovi – při návrhu interiéru je tedy přihlíženo k charakteru zboží, značce klienta (např. korporátní identitě).

Dále je společnost v rámci snahy vyjít klientovi maximálně vstříc schopna zařídit celou prodejnu tzv. na „klíč“. Tedy obstarat veškeré procesy související s realizací prodejny

⁴⁸ Zdroj: <http://www.tandemcrc.cz>

(např. zajištění stavebního povolení, zajištění patřičné elektroinstalace, osvětlení, instalace výpočetní / prodejní techniky atp.)

Společnost je řízena jednatelem, který vzhledem k velikosti firmy obstarává většinu provozních, obchodních i technických záležitostí.

Montážní pracovníci pracují pod přímým dohledem jednatele společnosti.

2.2 Popis současné situace

Společnost v současné době nemá žádný zpracovaný systematický postup, kterým by pokrývala oblast styku se zákazníky. Před několika lety se ve společnosti pokusily používat program Microsoft Outlook na kompletní pokrytí agendy zákazníků – tento systém se však ukázal jako silně neefektivní a proto se od něj opustilo.

Informace o obchodních partnerech (ať už stávajících či potenciálních) jsou v současné době různě uloženy v souborech tabulkového procesoru Excel, na různých papírech či vloženy v klasických papírových šanonech.

Vzhledem ke skutečnosti, že v minulosti nedošlo k nastavení kritérií dat, které se mají uchovávat je v dnešní době i ta množina dat, která je uchována

Samotné obchodní komunikace nejsou taktéž nikterak uceleně zachyceny – pouze jako součástí jednotlivých emailů, smluv, faktur a podobných dokladů. Možnost získat souhrnné, uspořádané informace o konkrétním klientovi je tedy velmi obtížné, ne-li nemožné.

2.2.1 Popis vztahu se zákazníkem

Na základě analýzy provedené ve společnosti a obecných principů klientského vztahu byly stanoveny základní fáze životního cyklu kontaktu se zákazníkem ve společnosti včetně zkušeností, které společnost získala.

Fáze kontaktu se zákazníkem.

- Získání klienta

- Je klíčovým akvizičním procesem – dochází zde ke snaze získat kontakt s potenciálním klientem a následně dosáhnout úspěšné konverze na aktivního zákazníka. První kontakt s klientem většinou bývá v této společnosti realizován přes získané doporučení v rámci referencí, poptávkový server či přes webové stránky společnosti, které obsahují přímý e-mailový formulář, který může v případě zájmu potenciální klient okamžitě využít.

Konverze probíhá především po osobním setkání, po vyjasnění vzájemných představ a požadavků. Je zde tedy co největší snaha dosáhnout osobního kontaktu s potenciálním zákazníkem.

- Samotný průběh vztahu

Zde dochází k realizaci většiny interakcí a transakcí v rámci převážně obchodního vztahu. Zkušenosti a informace získané v průběhu spolupráce umožňují lepší pochopení klienta, získání zpětné vazby a možnost se lépe přizpůsobit požadavkům klienta.

Udržení dlouhodobé spolupráce je vysoce důležitým zájmem společnosti – vzhledem k faktu, že pokud by klientská společnost otvírala například další pobočku tak je vysoká pravděpodobnost že opět realizaci interiéru. Dlouhodobé a dobře fungující vztahy velmi usnadňují společnosti stabilní ekonomickou situaci, což vychází z charakteristiky jejího předmětu podnikání

- Ukončení vztahu

Vzhledem k přirozenému vývoji vztahu může dojít k ukončení spolupráce s klientem. Avšak pokud už k takové situaci dojde, je žádoucí, aby byl poznamenán důvod ukončení spolupráce – převážně z důvodů možnosti zpětné analýzy a přijetí případných nápravných opatření.

- Případně obnovení vztahu

Již opakovaně nastaly situace, kdy klient, se kterým byl v minulosti přerušen vztah – pochopení potřeb klienta a informace o klientovi z dřívějšího období (přestože informace nebyly řádně archivované a byly uloženy především v paměti jednatele) pak byly shledány jako velmi užitečné. Z tohoto důvodu, je tedy užitečné uchovávat údaje i o v současnosti neaktivních klientech, a v případě potřeby je pak pouze aktualizovat ty informace, které již pozbyly platnosti.

2.3 Analýza problému

Jak jsem již výše zmínil, v současné době tedy nastává problém jak data o klientech systematicky ukládat, zpracovávat a později vyvolávat.

Výše nastíněná situace již společnosti nedostačuje a omezuje její potenciál růstu. Společnost od systému očekává lepší přehled o svých klientech a umožnění rozvoje v oblasti CRM do budoucna. Jednatel společnosti tedy stanovil následující tři **požadavky na funkčnost** nového systému:

- Evidovat jednotlivé klienty a jejich kontakty
- Zaznamenávat události k jednotlivým klientům
- Možnost nahrát soubory ke klientům

2.3.1 Evidence klientů

Je tím nejdůležitějším požadavkem, který společnost na systém má. Jedná se tedy o schopnost uchovat informace.

Z tohoto pohledu se bude jednat o jistou lepší „kartotéku“, ve které lze snadno vyhledávat.

Předpokládá se, že v ní budou uchovány následující údaje:

- Název zákazníka
- IČ
- Adresní údaje

- Kontaktní osoby a jejich kontaktní údaje.
- Aktuální označení stavu spolupráce se společností Tandem CRC CZ (tedy například zdali se jedná o stálého a zavedeného zákazníka či naopak o zákazníka potencionálního)
- Možnost vložit k firmě poznámku

2.3.2 Zaznamenávání událostí.

Právě touha po zlepšování vztahu se zákazníky dala za vznik požadavku na správu událostí ve vztahu s klientem.

Po konzultaci s jednatelem společnosti jsme dospěli k rozhodnutí navrhnoutí základních kategorií událostí a v nich patřičné jednotlivé druhy událostí. Tedy například zaslaný email s nabídkou vůči klientovi by měl být v systému zaznamenán v kategorii událostí „Elektronická komunikace“ a jako druh události „Zaslání nabídky e-mailem“. Taktéž je požadavek, aby systém umožnil zaznamenávat obsah událostí (tedy zde například shrnutí emailu).

Do systému by tedy měl být uložen každý jev, který má relevanci k vztahu SPOLEČNOST – KLIENT.

Pokud bychom však data pouze do systému nahrávali, ale nijak dál nevyužívali tak by to bylo značně samoučelné chování. Silnou motivací je tedy mít možnost uložená data opět ze systému zpřístupnit, optimálně ve formě sestavy, která umožní hlubší pochopení vztahu s klientem a klientových potřeb obecně.

2.3.3 Nahrávání souborů

Vzhledem k tomu, že pracovníci společnosti často pracují mimo kancelář (typicky na realizaci interiéru prodejny) a jsou tak v současné době obvykle mimo dosah dat a komunikace, která proběhla a taktéž k různým počítačovým souborům, které mohou být potřebné. Z tohoto důvodu je třeba umožnit, aby systém podporoval ukládání souborů k jednotlivým klientům a umožňoval jejich přehlednou správu.

3 Vlastní návrhy řešení

Vzhledem k provedené analýze a vypracované specifikaci jsem se v souladu s principy softwarového inženýrství rozhodl nejdříve zpracovat konceptuální model databáze na základě požadavků společnosti a následně model logický. Z hlediska funkčního diagramu jsem se rozhodl zvolit realizaci diagramem datových toků. Dále slovní popis a vývojový diagram u nejdůležitějších procesů

3.1 Konceptuální model

S ohledem na požadavky společnosti jsem identifikoval základní entity, které se v modelu vyskytují. Vzhledem k předpokládanému využití jazyka SQL jsem si dovilil použít anglické výrazy pro pojmenování jednotlivých entit, atributů a relací v databázovém konceptu.

3.1.1 Identifikace entit

- Customer – jednotlivý klient.
- Event – událost
- Event type – druh určité události
- Event category – jedná se o kategorii události.
- Attachment – příloha / soubor v systému
- Group – uživatelská skupina
- User – uživatel systému
- Contact – kontaktní osoba
- Address – adresa

3.1.2 Identifikace relací

Na základě identifikovaných entit jsem mohl identifikovat relace mezi jednotlivými základními entitami a určit kardinalitu jejich vztahu. Pro větší přehlednost jsem je zaznamenal formou tabulky.

Tabulka 3: Konceptuální schéma - zachycení vztahů⁴⁹

| Entita | Vztah | Entita | Kardinalita |
|-----------|---------------------|----------------|-------------|
| User | Manages | Customer | N:M |
| User | isMemberOf | Group | 1:N |
| User | Pastes | Attachment | N:1 |
| User | Creates | Event | N:1 |
| Customer | AppersIn | Event | N:1 |
| Customer | Has | Address | N:1 |
| Customer | Has | Contact | N:1 |
| Customer | isRelatedTo | Attachment | N:1 |
| Event | isRepresenatationOf | Event_type | 1:N |
| EventType | isPartOf | Event_category | 1:N |

3.1.3 Identifikace atributů

V této fázi jsem na základě analýzy zjistil, jaké faktické informace bude potřeba uchovávat k jednotlivým entitám a vazbám.

Například u entity Customer potřebujeme vzhledem k analýze uchovávat následující atributy:

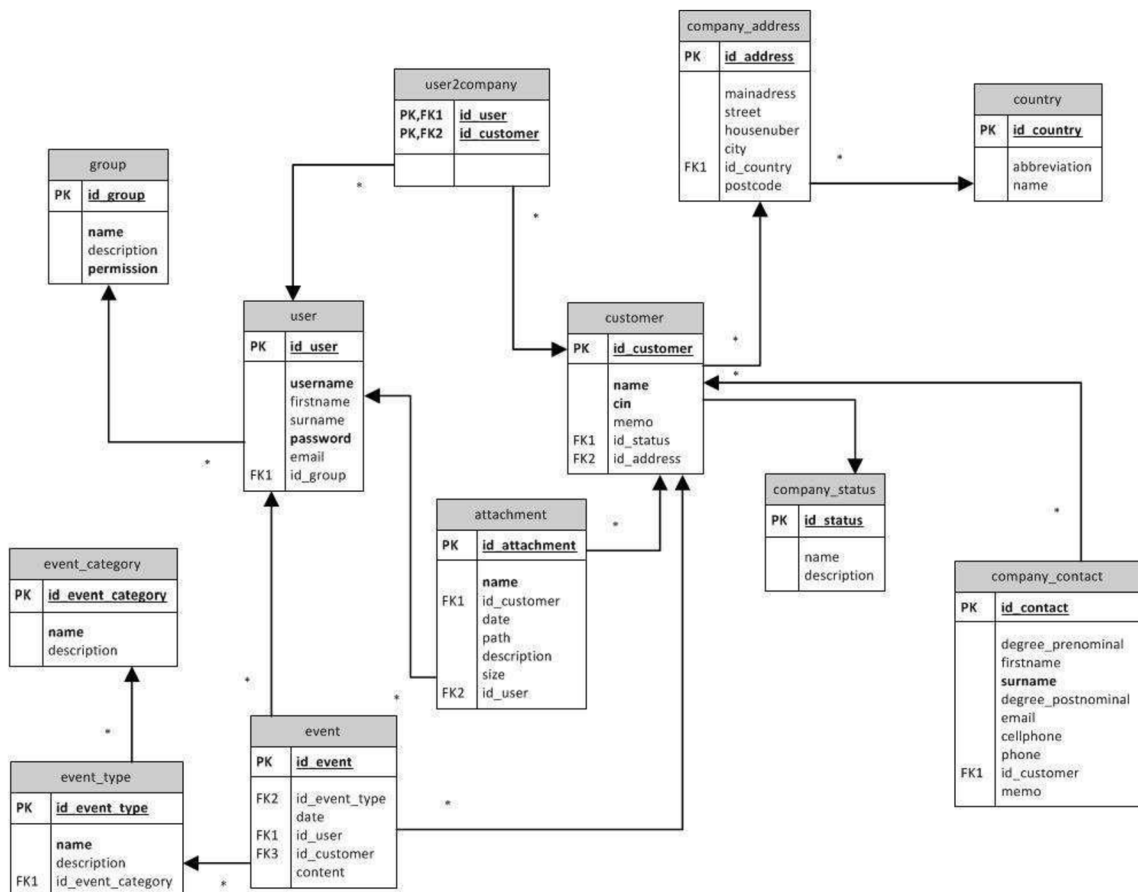
- Název subjektu - name
- Identifikační číslo společnosti – cin
- Adresy - address
- Kontaktní osoby – contact

⁴⁹ Zdroj: vlastní zpracování.

Na základě analýzy jsem takto doplnil atributy u všech entit – všechny jsou uvedené níže v logickém modelu, kde jsou již upraveny s ohledem na relační charakteristiku logického modelu.

3.2 Logický model

Na základě konceptuálního modelu s již doplněnými atributy byl vypracován logický model, kde jsou již data reprezentována ve formě relačního datového modelu. Rovněž již proběhla kontrola na splnění požadavků normálních forem. Pro její splnění bylo vytvořeno několik dodatečných číselníkových tabulek (country, status). Abych graficky alespoň částečně rozlišil mezi entitami z pohledu ER modelu a relacemi z pohledu logického tak jsem zvolil u logického schématu tzv. „lowercase“ – tedy psaní malými písmeny. Vzhledem k charakteristice vztahu musela proběhnout dekompozice přes vazební entitu user2company.



Obrázek 6 Výsledný ER diagram⁵⁰

⁵⁰ Zdroj: vlastní zpracování.

3.3 Popis relací, atributů a vztahů v logickém modelu

customer – je reprezentací entity zákazníka. Každý záznam (datová n-tice) reprezentuje jednoho konkrétního klienta.

Soupis atributů:

- Id_customer - umělý primární klíč. Jedná se o přidělené identifikační číslo klienta. [Numerický atribut]
- Name – název společnosti. [Řetězec znaků]
- Cin – identifikační číslo společnosti. [Numerický atribut]
- id_status – Cizí klíč. Reprezentuje aktuální status společnosti [Numerický atribut]
- id_address – Cizí klíč. Přiřazení adres ke společnosti přes cizí klíč (může jich být tedy i více) [Numerický atribut]
- memo – poznámka k firmě. [Řetězec znaků]

company_contact – obsahuje kontaktní osoby dané společnosti. Jednotlivé položky reprezentují vždy jednu kontaktní osobu.

Soupis atributů

- id_contact - Umělý primární klíč. Jedná se o přidělené identifikační číslo pro rozlišení jednotlivých kontaktů klienta. [Numerický atribut]
- degree_prenominal – titul před jménem [Řetězec znaků]
- Firstname – křestní jméno [Řetězec znaků]
- Surname – Příjmení [Řetězec znaků]
- degree_postnominal – Titul za jménem [Řetězec znaků]
- email – emailová adresa kontaktu [Řetězec znaků]
- cellphone – mobilní telefon [Řetězec znaků]
- phone – telefon (pevná linka) [Řetězec znaků]

- id_company – Cizí klíč. Slouží k vyjádření vztahu ke společnosti. [Numerický atribut]
- memo – poznámka ke kontaktu [Řetězec znaků]

company_address – zachycuje adresy relevantní k jednotlivým klientům. Každý záznam je reprezentací jedné adresy.

Soupis atributů:

- id_address - Úmělý primární klíč. Jedná se o přidělený identifikátor adresy. [Numerický atribut]
- mainaddress – umožňuje nastavit jednu z adres u klienta jako hlavní (pokud jich je více). [Numerický atribut]
- street - označuje adresu. [Řetězec znaků]
- houzenumber – označuje číslo domu.[Řetězec znaků]
- city – označuje město.[Řetězec znaků]
- id_country – cizí klíč, slouží k vazbě na číselník států.[Numerický atribut]
- postcode – poštovní směrovací číslo. [Řetězec znaků]

attachment – slouží jako reprezentace meta-dat uložených souborů (soubory samotné nejsou ukládány do databáze). Jeden řádek odpovídá jednomu uloženému souboru.

- id_attachment - umělý primární klíč. Jedná se o přidělený identifikátor jednotlivého souboru. [Numerický atribut]
- name – jedná se o jméno souboru. [Řetězec znaků]
- Id_company – cizí klíč, slouží k zachycení vztahu s firmou, které se soubor týká [Numerický atribut]
- date – datum a čas vložení souborů [Řetězec znaků]
- size – velikost souborů v Kb. [Numerický atribut]
- path – označuje adresu k fyzickému souboru [Řetězec znaků]
- description – popis souboru [Řetězec znaků]

event – reprezentuje jednotlivé události v kontaktu se zákazníky. Jednotlivé věty jsou zachycením každé jednotlivé události.

- id_event - Umělý primární klíč. Jedná se o přidělený identifikátor jednotlivé události. [Numerický atribut]
- date – datum a čas vložení události. [Řetězec znaků]
- id_event_type – cizí klíč, slouží k určení typu události [Numerický atribut]
- id_company – cizí klíč, slouží k zachycení vztahu s firmou, které se událost týká [Numerický atribut]
- content – obsah a popis samotné události. [Řetězec znaků]

event_type – slouží k uložení jednotlivých typů události v kontaktu se zákazníky. Jednotlivý záznam určuje typ události (například “odeslání emailu klientovi”). Tato relace slouží jako číselník.

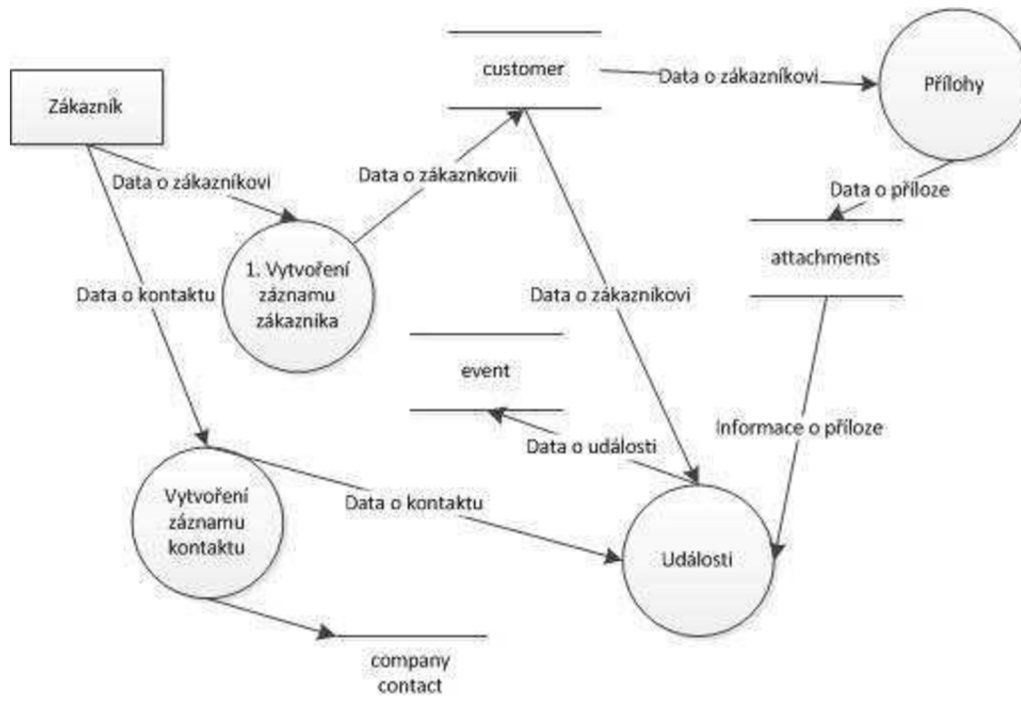
- id_event_type – Umělý primární klíč, slouží k určení druhu události. [Numerický atribut]
- id_event_category – cizí klíč, slouží k vazbě na kategorii události [Numerický atribut]
- name – název typu události. [Řetězec znaků]
- description – popis události. [Řetězec znaků]

event_category – zosobňuje jednotlivé události v kontaktu se zákazníky. Jednotlivý záznam určuje typ události (například kategorie akvizice atp.)

- id_event_category – Umělý primární klíč, slouží k určení druhu události. [Numerický atribut]
- name – název kategorie [Řetězec znaků]
- description – popis kategorie. [Řetězec znaků]

3.4 DFD - Funkční model

Základní funkční pohled na systém poskytuje DFD diagram uvedený níže. Funkční model opět koresponduje se základními požadavky společnosti.



Obrázek 7 Diagram toku dat - konceptuální úroveň⁵¹

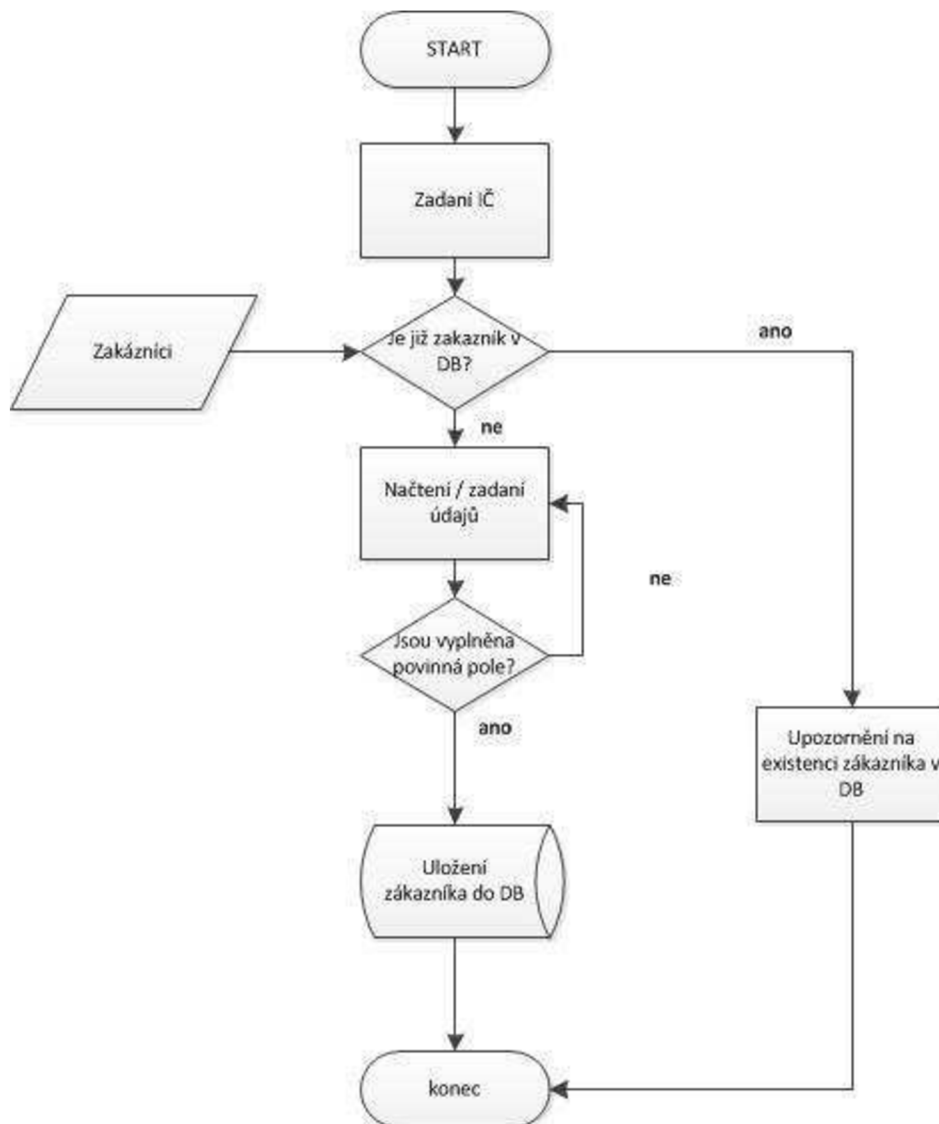
3.5 Popis nejdůležitějších procesů

3.5.1 Vytvoření záznamu zákazníka

Tento proces slouží k vytvoření záznamu o zákazníkovi v databázi. Uživatel si zobrazí modul zákazníci a zvolí přidat nového klienta. Obecně prvně zadaným údajem je IČ klienta (případně jiný unikátní identifikátor v případě), pomocí kterého se ověří, zdali již neexistuje tento klient v databázi. Pokud již klient existuje, uživatel je na to upozorněn a aplikace znemožní přidat duplicitní záznam. Pokud klient v systému zaveden není, tak se program pokusí načíst část údajů z veřejných rejstříků (např. Obchodního rejstříku). Následně uživatel doplní údaje (například způsob navázání kontaktu, aktuální stav, kontaktní osoby atp.). Před finálním uložením údajů systém

⁵¹ Zdroj: vlastní zpracování

zkontroluje, zdali jsou vyplněna veškerá povinná pole (zejména jméno společnosti / klienta a aktuální stav) – pokud některý chybí tak uživatele vyzve k doplnění. Pokud je vše v pořádku, tak zapíše požadovaná data do databáze.

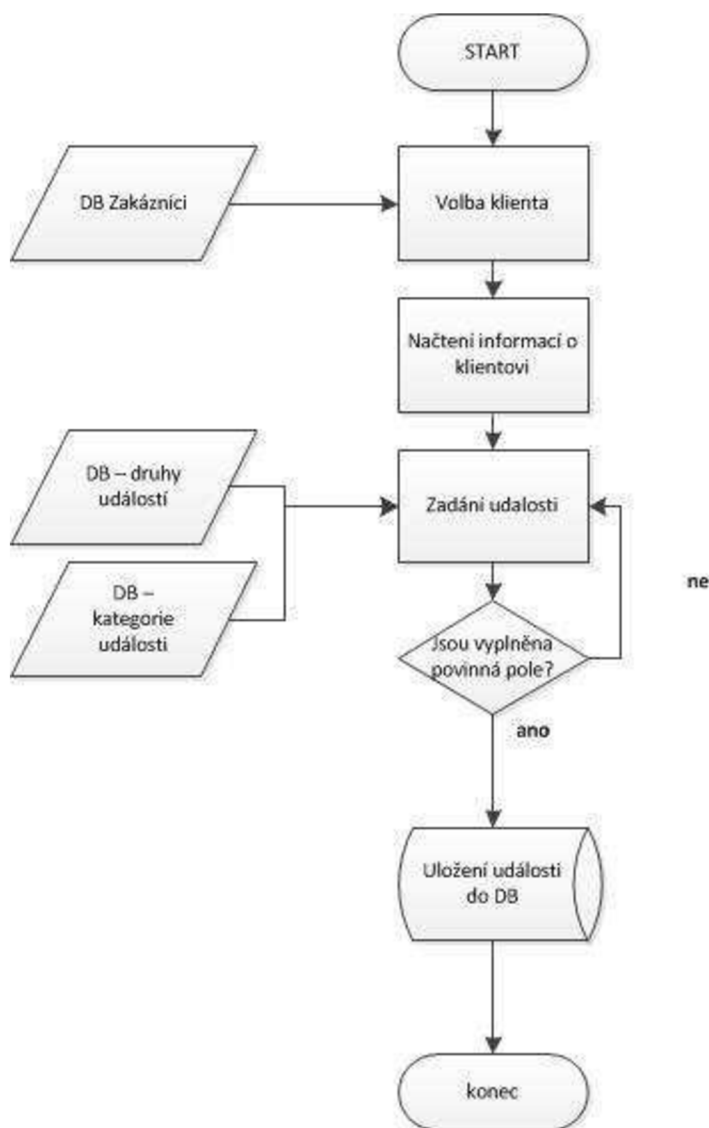


Obrázek 8: Vývojový diagram - přidání zákazníka do DB⁵²

3.5.2 Přidání události

⁵² Zdroj: vlastní zpracování

V rámci modulu zákazníci si uživatel vybere klienta, ke kterému chce přidat novou událost. Systém zobrazí informace o klientovi a výpis posledních událost a připraví formulář pro zadání nové události. Zde si vybere druh události, její kategorii (tyto volby jsou realizované formou číselníku – roletkového menu). Po vyplnění proběhne kontrola povinných údajů – pokud jsou podmínky splněny tak se provede zápis do databáze. Jednotlivé události se následně zapíší do databázové tabulky event.

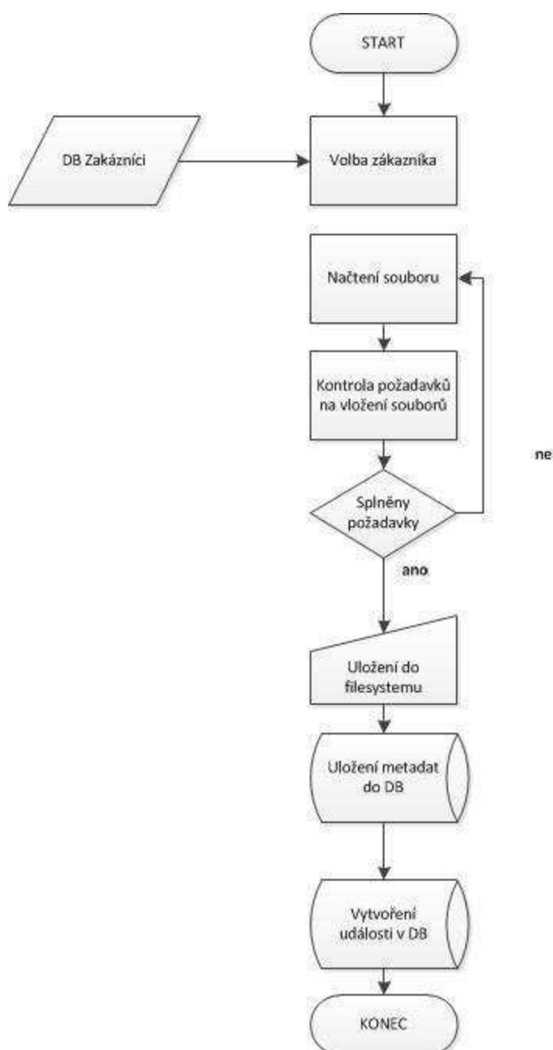


Obrázek 9: vývojový diagram přidání události⁵³

⁵³ Zdroj: vlastní zpracování

3.5.3 Nahrání souboru

Po vybrání konkrétního klienta v modulu zákazníci zvolí uživatel možnost spravovat soubory a následně přidat soubor. V rámci svého prohlížeče zvolí soubor ze svého počítače a zvolí nahrát. Proběhne kontrola podmínek (převážně kontrola na velikost a jméno souboru). Pokud je nalezen problém, tak systém akci přeruší a uživateli znovu zobrazí načítací formulář a informuje jej o dané chybě. V opačném případě soubor uloží na disk do patřičné složky a vytvoří o něm následně dva záznamy. První záznam je v tabulce attachment, kde jsou uloženy metadata o souboru (název, popis, fyzická cesta k souboru). Dále vytvoří událost u daného klienta (v tabulce event), že byl nahrán soubor (daná událost je v číselníku druhů a kategorií událostí).



Obrázek 10: vývojový diagram nahrání souboru⁵⁴

⁵⁴ Zdroj: vlastní zpracování

3.6 Skladba aplikace

Jako doplnění k funkčnímu modelu jsem zpracoval dělbou do základních modulů a podmodulů aplikace

Samotná aplikace má spíše obslužný a stará se především o zobrazování dat a interakci. Většina vykonatelných metod by se měla implementovat v databázové rovině (s určitými výjimkami – například o nahrání souboru je ponecháno na aplikační rovině, v databázi se uchovávají pouze metadata)

Jednotlivý systém je rozdělen do několika modulů, které se starají o jednotlivou funkcionalitu.

3.6.1 Modul zákazník

Je klíčovým modulem, realizuje základní operace s klientem.

Obsahuje tři následující submoduly:

Události – je přehledným zobrazení naprosté většiny informací o zákazníkovi a o interakci s ním. Umožňuje generovat sestavy událostí.

Základní operace:

- Zobrazit výpis událostí.
- Přidat událost
- Smazat událost
- Editovat událost
- Vypsát sestavu událost

Soubory – slouží k manipulaci se soubory, které jsou relevantní k jednotlivým klientům.

Základní operace:

- Prohlížet / stahovat soubory
- Smazat soubory
- Přidat soubor

- Editovat metadata souboru

Kontakty - slouží ke správě kontaktů u jednotlivých klientů.

Základní operace:

- Prohlížet / stahovat soubory
- Smazat soubory
- Přidat soubor
- Editovat metadata souboru

3.6.2 Modul správa

Dlouží především k nastavení uživatelů a jejich skupin. Je přes něj možné vytvořit nového uživatele či editovat již existujícího. Obdobné akce lze provést i se skupinami, které mají ještě možnost nastavit oprávnění.

Daný modul tedy obsahuje dva následující submoduly:

Uživatel

Základní operace:

- Vytvoření nového uživatele
- Editace existujícího uživatele
- Smazání uživatele

Skupina

Základní operace:

- Vytvoření nové skupiny
- Editace existující skupiny
- Smazání skupiny

4 Závěr

Stěžejním cílem mé práce bylo navrhnout systém pro správu klientů společnosti Tandem CRC CZ, s.r.o., který by řešil konkrétní požadavky společnosti.

Návrh měl po přesnějším zanalyzování potřeb společnosti obsahovat především konceptuální a logický model z hlediska databáze. Z logického modelu lze bez větších potíží zformulovat fyzický model a po otestování nastavení databázového systému následně databázi nasadit.

Taktéž funkční model byl zhotoven ve formě diagramu datového toku a důležité části byly zdokumentovány navíc slovním popisem a vývojovým diagramem.

Lze tedy říci, že cíle bakalářské práce tedy bylo dosaženo – v současném stavu je možno systém implementovat podle návrhu.

Klíčová by byla především databázová část, kde při fyzické implementaci by se mělo brát v potaz růst firmy a příčným způsobem systém naddimenzovat. Taktéž implementace frontendu by měla zohlednit různé možnosti přístupu k rozhraní (PC, notebooky, smartphone, tablet...)

Seznam použitých zdrojů

Knížní:

BUREŠ, I. *10 zlatých pravidel péče o zákazníka aneb CRM v digitálním věku*. 2. vyd. Praha: Management Press, 2006. 158 s. ISBN 80-7261-149-6.

DOHNAL, Jan. *Řízení vztahů se zákazníky: procesy, pracovníci, technologie*. 1. vyd. Praha: Grada, 2002, 161 s. ISBN 80-247-0401-3.

GILMORE, W. *Velká kniha PHP 5 a MySQL: kompendium znalostí pro začátečníky i profesionály*. Nové, 3. vyd. Brno: Zoner Press, 2011, 736 s. Encyklopedie Zoner Press. ISBN 978-80-7413-163-9.

GUTMANS, A., S. Bakken, D. Rethans. *Mistrovství v PHP 5*. 2. vyd. Brno: Computer Press, 2007. 655 s. ISBN 978-80-251-1519-0.

CHLEBOVSÝ, V. *CRM: řízení vztahu se zákazníky*. 1. vydání. Brno : Computer Press, 2005. 190 s. ISBN 80-251-0798-1.

KOCH, M., B. NEUWIRTH. *Datové a funkční modelování*. vyd. 4., rozšířené. Brno: Akademické nakladatelství CERM, 2010, 142 s. Učební texty vysokých škol. ISBN 978-80-214-4125-5.

LEHTINEN, J. *Aktivní CRM: Řízení vztahů se zákazníky*. 1. vyd. Praha: Grada, 2007. 158 s. ISBN 978-80-247-1814-9.

SOMMERVILLE, Ian. *Software engineering*. 8th ed. New York: Addison-Wesley, 2007, xxiii, 840 s. ISBN 03-213-1379-8.

On-line:

¹ BATKO, M. *Relační vs. objektově-relační vs. objektové databáze* [online] 2002. [cit. 2013-04-12]. Dostupné z <http://www.fi.muni.cz/~xbatko/oracle/compare.html>

CODD, E. F. *A Relational Model of Data for Large Shared Data Banks* [online]. 1970 [cit. 2013-04-12]. Dostupné z <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>

GOMORI, S. Dr. E. F. Codd's 12 rules for defining a fully relational database [online]. 2005 [cit. 2013-04-12]. Dostupné z <http://www.cse.ohio-state.edu/~sgomori/570/coddsrules.html>

HABAŇ, J., SODOMKA, P. *Efektivní tvorba a provoz datových skladů* [online]. Univerzita Tomáše Bati ve Zlíně, Ústav managementu výroby: 2003 [cit: 2013-04-12]. Dostupné z: <http://si.vse.cz/archive/proceedings/2003/efektivni-tvorba-a-provoz-datovych-skladu.pdf>

CHEN, P. The Entity-Relationship Model – Toward a Unified View of Data. [online] 1970 [cit. 2013-04-12]. Dostupné z <http://www.csc.lsu.edu/news/erd.pdf>.

KIMBLE, Ch., The Software Crisis [online]. 2005 [cit. 2013-04-12]. Dostupné z http://www.chris-kimble.com/Courses/World_Med_MBA/Software_Crisis.html

KOCAN M., *Databáze nejsou jen relační, díl druhý – HDM*. [online] 2001. [cit. 2013-04-12]. Dostupné z : <http://www.dbsvet.cz/view.php?cisloclanku=2001112601>

KOCAN M., *Databáze nejsou jen relační, díl druhý – HDM*. [online] 2001. [cit. 2013-04-12]. Dostupné z : <http://www.dbsvet.cz/view.php?cisloclanku=2001112601>

RANDELL, Brian. The 1968/69 NATO Software Engineering Reports. The NATO Software Engineering Conferences [online]. 2001 [cit. 2013-04-12]. Dostupné z: <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/NATOREports/index.html>

SOMMERVILLE, Ian. Software engineering history. Software engineering history [online]. 2008 [cit. 2013-04-12]. Dostupné z: <http://www.cs.st-andrews.ac.uk/~ifs/Books/SE9/Web/History/>

WILLIAMS, Kenneth. Notes on Software Engineering Goals [online]. 2005. [cit. 2013-05-31]. Dostupné z: <http://williams.comp.ncat.edu/segoals.htm>

Seznam obrázků.

| | |
|--|-----------|
| Obrázek 1: Lineární datový model | 19 |
| Obrázek 2 : Hierarchický datový model | 20 |
| Obrázek 3: Síťový datový model | 21 |
| Obrázek 4: relační datová model | 22 |
| Obrázek 5: Logo společnosti | 34 |
| Obrázek 6 Výsledný ER diagram | 41 |
| Obrázek 7 Diagram toku dat - konceptuální úroveň | 45 |
| Obrázek 8: Vývojový diagram - přidání zákazníka do DB | 46 |
| Obrázek 9: vývojový diagram přidání události | 47 |
| Obrázek 10: vývojový diagram nahraní souboru | 48 |

Seznam tabulek

| | |
|--|-----------|
| Tabulka 1 Popis vývojového diagramu | 30 |
| Tabulka 2 Symboly DFD diagramu (konvence Yourdon and Codad) | 31 |
| Tabulka 3: Konceptuální schéma - zachycení vztahů | 40 |

Seznam příloh

Práce nemá žádné přílohy