

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
DEPARTMENT OF INFORMATICS

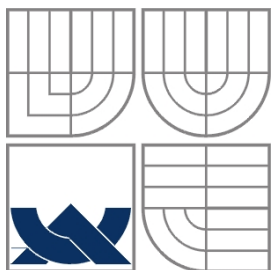
ANALÝZA, NÁVRH A IMPLEMENTACE
INFORMAČNÍHO SYSTÉMU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

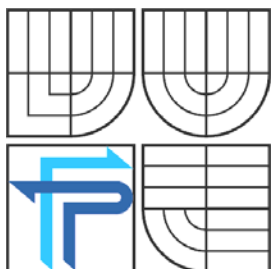
AUTOR PRÁCE
AUTHOR

Bc. Jan Wohlgemuth

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
DEPARTMENT OF INFORMATICS

ANALÝZA, NÁVRH A IMPLEMENTACE INFORMAČNÍHO SYSTÉMU

ANALYSE, CONCEPT AND IMPLEMENTATION OF INFORMATION SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

BC. JAN WOHLGEMUTH

VEDOUCÍ PRÁCE

SUPERVISOR

ING. PAVEL WEIRICH

BRNO 2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Wohlgemuth Jan, Bc.

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:

Analýza, návrh a implementace informačního systému

v anglickém jazyce:

Analyse, Concept and Implementation of Information system

Pokyny pro vypracování:

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



Seznam odborné literatury:

RYCHLÝ M., WEISS P. Architektura orientovaná na služby, návrh orientovaný na služby, webové služby. FIT VUT v Brně, Brno 2007.

ŘEPA, VÁCLAV. Podnikové procesy: procesní řízení a modelování. Praha: GRADA Publishing, 2007. 281 s. ISBN 978-80-247-2252-8.


AMSDEN, J. Business services modeling [online]. Poslední revize 27.12.2005. Dostupné z: <http://www.ibm.com/developerworks/rational/library/05/1227_amsden/>.


WILLIAMS, H. a LANE, D. Programujeme webové aplikace pomocí PHP a MySQL. ISBN 80-7226-760-4.

Vedoucí bakalářské práce: Ing. Pavel Weirich

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2008/09.




Ing. Jiří Kříž, Ph.D.
Ředitel ústavu


doc. RNDr. Anna Putnová, Ph.D., MBA
Děkanka fakulty

V Brně, dne 28.2.2009

Abstrakt

Tato práce se zabývá teorií návrhů firemního informačního systému od počáteční analýzy firemních procesů až po závěrečnou implementaci navrženého systému na běžně zdarma dostupných technologiích. Všechny postupné kroky jsou demonstrovány na případu malého živnostenského podnikání v oboru IT.

Klíčová slova

Informační systém, UML, ER diagram, Use Case, Business procesy, Databáze

Abstract

This thesis deals with the theory of the designs of the company information system, starting from the initial analysis of the business processes to the final implementation of the system based on the affordable technologies (commonly available for free). All the successive steps of the process are demonstrated on the example of a self-employed entrepreneur in the IT sector.

Keywords

Information system, UML, ER Diagram, Use Case, Business processes, Database

Citace

WOHLGEMUTH, J. *Analýza, návrh a implementace informačního systému*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2009. 59 s. Vedoucí bakalářské práce Ing. Pavel Weirich.

Analýza, návrh a implementace informačního systému

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Pavla Weiricha. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Bc. Jan Wohlgemuth
25. 5. 2009

Poděkování

Rád bych na tomto místě poděkoval vedoucímu své bakalářské práce Ing. Pavlu Weirichovi, který mi ochotně radil a pomáhal v průběhu celého zpracování. Dále bych rád poděkoval vedení společnosti HOPE – E.S., v.o.s., které mi dodalo další cenné rady a prostor pro studium.

© Jan Wohlgemuth, 2009.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě podnikatelské. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

OBSAH	8
ÚVOD	8
1 ANALÝZA PROBLÉMU	9
1.1 OBOR ČINNOSTI.....	9
1.1.1 Živnostenské oprávnění	9
1.1.2 Poskytované služby.....	10
1.2 AKTUÁLNÍ SITUACE.....	10
1.3 NÁVRH NA ZLEPŠENÍ	11
2 TEORETICKÁ VÝCHODISKA PRÁCE.....	12
2.1 PROJEKTOVÉ ŘÍZENÍ	12
2.2 ANALYTICKÁ A NÁVRHOVÁ ČÁST	14
2.2.1 Návrh relačních databází.....	14
2.2.2 UML	19
2.2.3 Byznys procesy – BPMN.....	23
2.3 IMPLEMENTAČNÍ ČÁST	24
2.3.1 PHP	24
2.3.2 HTML a CSS	25
2.3.3 MySQL	26
2.3.4 Zabezpečení	27
3 ANALÝZA SOUČASNÉ SITUACE.....	29
3.1 CÍTLIVÉ ÚDAJE	29
3.2 EFEKTIVNOST	30
3.3 SPOLUPRÁCE V TÝMU.....	30
3.4 SHRNUÍ	30
4 VLASTNÍ NÁVRH	32
4.1 SESTAVENÍ ČASOVÉHO PLÁNU PROJEKTU	32
4.1.1 Časový plán.....	32
4.2 ANALÝZA POŽADAVKŮ	33
4.3 SESTAVENÍ MODULŮ BUDOUCÍHO IS	33
4.3.1 Napojení budoucích modulů	35
4.4 MODEL UŽITÍ SYSTÉMU	36
4.5 MODEL DATABÁZE – ER DIAGRAM	38
4.6 NÁVRH ZABEZPEČENÍ SYSTÉMU	41

4.6.1	Důvěrnost dat	41
4.6.2	Integrita dat	41
4.6.3	Dostupnost služeb	41
4.6.4	Ochrana před zneužitím dat	42
5	VLASTNÍ IMPLEMENTACE	44
5.1	SYSTÉMOVÉ POŽADAVKY	44
5.2	ZAVEDENÍ SYSTÉMU	45
5.3	DESIGN SYSTÉMU	45
5.4	NÁVRH MOŽNÝCH ROZŠÍŘENÍ	47
	ZÁVĚR	49
	LITERATURA	50
	SEZNAM POUŽITÝCH OBRÁZKŮ	51
	SEZNAM POUŽITÝCH TABULEK	52
	SEZNAM POUŽITÝCH ZKRATEK	53
	SEZNAM PŘÍLOH	54
	PŘÍLOHY:	55

Úvod

Tento úvod bych rád začal zamyšlením, co hýbe dnešním světem, co je stěžejní pro rozhodování každého z nás. Ať už se jedná o rozhodnutí, zda jít večer do společnosti nebo raději zůstat doma. V lepším případě rozhodování obchodního charakteru, kdy, kolik a za kolik nakoupit surovin, které zaměstnance propustit v době ekonomické krize a podobně. Co stojí za tím či oním rozhodnutím?

Každý se rozhoduje na základě INFORMACÍ, které má aktuálně dostupné. V osobním životě je to poměrně snadné, stačí zavolat manželce, přítelkyni, kamarádovi, zeptat se, jestli se bude něco dít nebo zda bude lepší zůstat doma. Rozhodování v tomto případě nepotřebuje žádnou složitou podporu, abychom se rozhodli. Ale zamysleme se. Nebylo by lepší mít úplný přehled o všech akcích, které se aktuálně budou daný večer odehrávat? Přehled o tom, zda vaši kamarádi a kolegové zvolí právě jednu nebo druhou variantu, abyste se buď potkali nebo naopak nebyli naštvaní, že jste se minuli?

Pojďme se ale podívat na případ obchodníků, kteří se musí aktuálně rozhodovat o obchodech za významné peníze a na jejichž rozhodnutí často stojí osud celé společnosti a tím i dalších lidí. Oni se také mohou rozhodovat podle své vůle na základě informací, které nasbírali zkušenostmi a pozorováním? Asi mohou, ale určitě budou jejich rozhodnutí účinnější, pokud budou podložena reálnými fakty – tedy informacemi nasbíranými za léta svých zkušeností a obchodování. Vývoje cen surovin, trhy konkurentů, správa klientů, analýza činnosti zaměstnanců, to je jen zlomek z činností, které jdou dlouhodobě sledovat za pomoci informačních systémů a jejich využívání.

1 Analýza problému

Rozhodl jsem se analyzovat situaci mně nejlépe známou a zároveň tím přispět k lepší efektivitě práce ve svém vlastním podnikání. Zároveň bych rád ukázal, že i pro malý byznys podnikání fyzické osoby, má smysl budovat informační systém.

Svou podnikatelskou činnost jsem začal v srpnu loňského roku (20. 08. 2008) s tím, že jsem potřeboval prokázat odbornou způsobilost k činnostem, které jsem již delší dobu vykonával pro různé společnosti na dohodu o pracovní činnosti.

1.1 Obor činnosti

Oborem mé činnosti je všechno, co se týká informačních a komunikačních technologií (ICT) v malých a středních společnostech, zpravidla do 20ti zaměstnanců. Tyto společnosti nemají dostatek prostředků, aby vedli samostatné oddělení vyhrazené pro tuto činnost, ale zároveň tyto technologie musí využívat také a jsou pro ně stejně podstatné. Proto se zaměřuji právě na oblast menších a středních podniků, nejlépe v oblasti služeb, kde se na informační systémy doposud zapomínalo, a poskytnout jim kompletní servis za velmi nízké ceny.

1.1.1 Živnostenské oprávnění

Živnostenské oprávnění č. 1:

Předmět podnikání:

Výroba, obchod a služby neuvedené v přílohách 1 až 3 živnostenského zákona

Obory činnosti:

Zprostředkování obchodu a služeb

Poskytování software, poradenství v oblasti informačních technologií, zpracování dat, hostingové a související činnosti a webové portály

Poradenská a konzultační činnost, zpracování odborných studií a posudků

Příprava a vypracování technických návrhů, grafické a kresličské práce

Reklamní činnost, marketing, mediální zastoupení

Druh živnosti:

Ohlašovací volná

Vznik oprávnění:

20. 08. 2008

Zahájení provozování živnosti:

20. 08. 2008

Doba platnosti oprávnění:

na dobu neurčitou

1.1.2 Poskytované služby

Mezi hlavní činnosti nabízené klientům patří správa počítačové sítě, často podpořená firemním serverem. Dále to bývá správa webových portálů, navrhování a zavádění informačních systémů, jakožto i jejich následná správa. K tomu někteří klienti vyžadují obstarávání služeb třetích stran – poskytovatelé připojení k internetu, hostingových služeb, ale i telefonních pevných i mobilních služeb.

Tuto činnost nejsem schopen vykonávat sám, takže pro jednotlivé klienty najímám po dohodě buď přímo já nebo klient sám, mladé a schopné pomocné síly. Většinou se jedná o studenty vysokých škol, kterým taková práce dává vyšší šance na lepší budoucí uplatnění a rozhled v oboru.

1.2 Aktuální situace

V současné době poskytuji služby dvěma významným společnostem. Jedná se především o společnost HOPE – E.S., v.o.s. (HOPE), která působí v oboru jazykových služeb a odborného poradenství spojeného s čerpáním prostředků z Evropských fondů. Druhým stěžejním klientem je Kadeřnictví Pitty slečny Zuzany Pitauerové, pro které vyvíjím webový objednávkový systém.

Praxe je taková, že klient navrhne službu, kterou potřebuje a po vzájemné dohodě je služba zřízena a dále obstarávána. Jako službu si můžeme představit například zřízení emailových účtů nebo i instalace určité pracovní stanice či jiného prvku vnitropodnikové sítě. Společnost HOPE je pro mě jako klient specifická také v tom, že

má vlastní kancelář také v Praze, což znamená zajišťování správy i pražské pobočky. Drobné úpravy řeším buď pomocí nástrojů pro vzdálenou správu nebo po telefonu přímo s pracovníky společnosti. Ovšem pro větší zásahy do sítě nebo její rozšiřování je nutný výjezd. Mimo tyto dva klíčové klienty poskytují své služby drobnějším živnostníkům nebo za určitých podmínek i soukromým osobám.

Hlavní problém, který je potřeba odstranit, je v rychlém zorientování se v dané problematice té konkrétní služby či přístroje. V sítích, o které se starám, jsou použity různé síťové prvky od různých výrobců. Bývá zvykem, že při složitějším problému je nutné podívat se do dokumentace k přístroji nebo kontaktovat podporu k určité službě (ISP, hosting, apod.). Proto je bezpodmínečně nutné ukládat si tyto informace v co možná nejpřehlednější struktuře a aktuálních verzích. Toto aktuálně řeším naprosto nystematicky pouze na lokálním disku svého počítače, tak abych měl informace k dispozici, když je budu potřebovat. Toto řešení je však velmi nepraktické a způsobuje nemalé problémy v době například akutních zásahů, kdy není pravidlem, že bych měl počítač sebou vždy a všude. Pro tento případ mám doma počítač připojený na síť a mám možnost se na něj přihlásit vzdáleně, ale ani tento způsob není ideální. Nehledě na to, že takovéto řešení absolutně neumožňuje zapojení dalších pomocníků a sdílení těchto důležitých informací s nimi.

1.3 Návrh na zlepšení

Pro vyřešení problému s ukládáním, archivací a sdílením informací nutných pro správu služeb pro své klienty popsaného v kapitole 1.2 jsem se rozhodl navrhnout webový informační systém, který usnadní práci mě i mým případným zaměstnancům a zároveň dá zajímavý nástroj pro sledování nákladů na ICT prostředky do rukou mých klientů.

Tento systém by měl být navržen s ohledem na současné technologie, potřeby jak mé vlastní, tak potřeby mých klientů. Jednoznačným kritériem číslo jedna by měla být bezpečnost přenosu a uchování dat. Důležité také musí být intuitivní ovládání a množství doprovodných zlepšení.

2 Teoretická východiska práce

Po vzájemné komunikaci s vedoucím mé práce, jsem se rozhodl psát informační systém na technologiích, které nejlépe ovládám a přicházím s nimi pravidelně do styku. Tyto technologie patří k nejpoužívanějším v jedno-programátorských projektech, ale jsou často vidět i na velikých a sofistikovaných aplikacích.

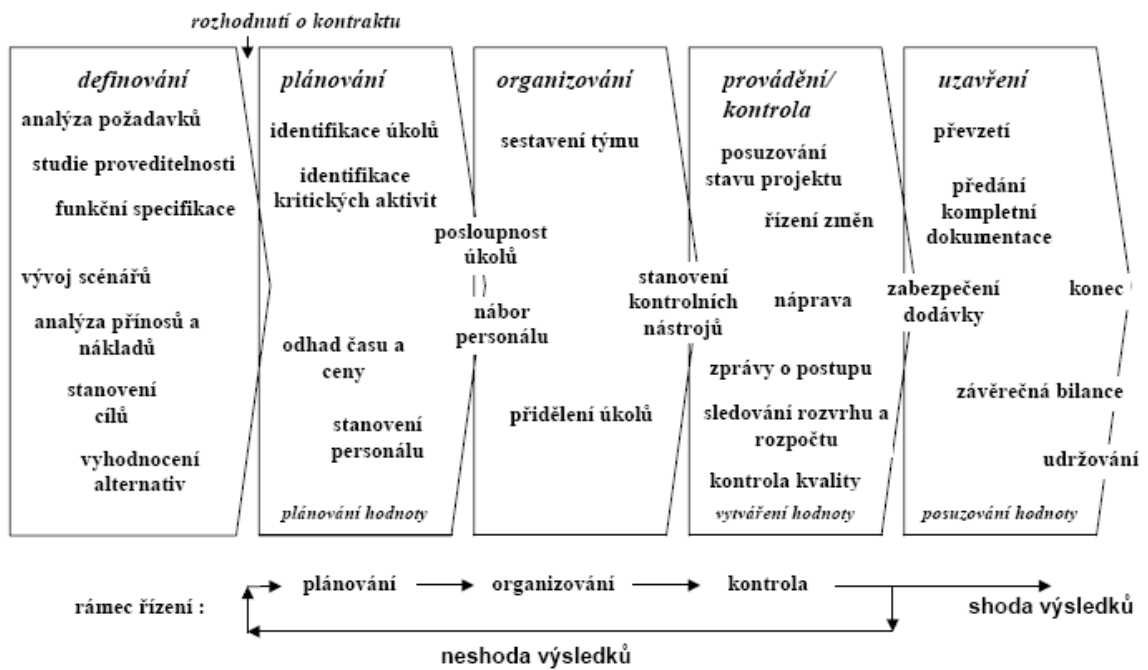
2.1 Projektové řízení

První nutnou vstupní podmínkou, před vytvořením informačního systému, je třeba mít správně nastaveny všechny důležité projektové součásti. Tímto se zabývá obor projektového řízení, který se pokusím objasnit v této kapitole.

Motivace projektového řízení:

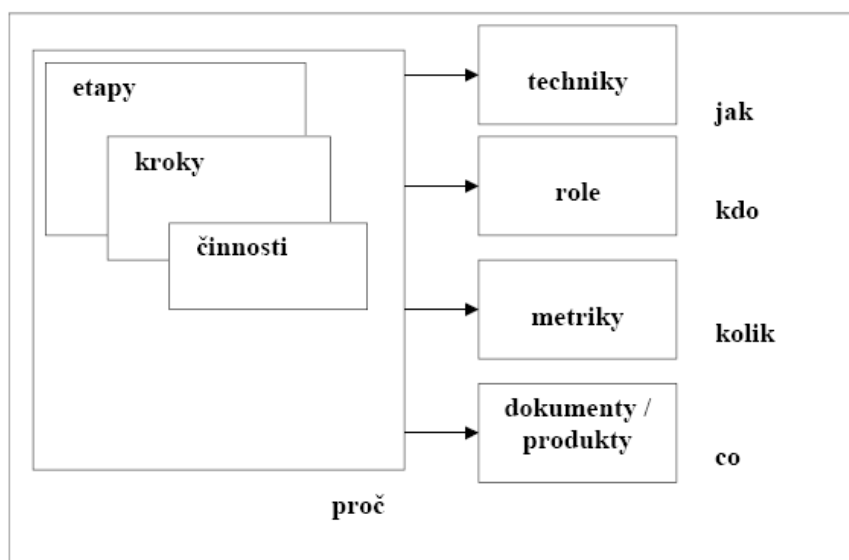
Projektové řízení patří v západních zemích ke standardnímu způsobu práce úspěšných firem a západní management považuje znalosti projektového řízení za nedílnou součást dovednosti řídicích i řadových pracovníků. U nás je bohužel projektové řízení zatím poměrně málo známé. Proto řada důležitých akcí v našich podnicích probíhá neúspěšně. Navíc mají naše firmy potíže při komunikaci a spolupráci se západními firmami, kde je projektové řízení velmi rozšířené a běžně používané. [1]

Vzhledem k tomu, že projekty obsahují jedinečné činnosti, zahrnují určitý stupeň nejistoty. Proto se každý projekt dělí na několik fází, s cílem usnadnit operativní řízení a vytvořit odpovídající vazby se stálými činnostmi v rámci prováděcí organizace. Ve svém souhrnu jsou fáze projektu známy jako životní období projektu, tak jak je vidíme na obrázku č.1.



Obrázek č. 1 Životní cyklus projektu z hlediska projektového řízení [3]

Každá etapa projektu se vyznačuje dokončením jednoho nebo více předmětů dodávky. Předmět dodávky je hmotný, ověřitelný produkt práce. Uzavření etapy projektu se obvykle vyznačuje přezkoumáním klíčových předmětů dodávek a plnění projektu s cílem určit, zda projekt může pokračovat další fází případně s minimálními náklady zjistit a opravit chyby.



Obrázek č. 2 Životní cyklus projektu z hlediska vývoje produktu [3]

2.2 Analytická a návrhová část

V analytické části se budu snažit využít maximum z toho, co jsem se naučil při svém studiu různých inženýrských a manažerských oborů na vysokých školách v Brně. Prošel jsem několika předměty, které se zabývají navrhováním a projektováním různých informačních systémů. Nikoli na svou obranu, ale jako fakt, bych zmínil nejednoznačnost těchto předmětů mezi sebou a určitou opožděnost za používanou praxí.

2.2.1 Návrh relačních databází

Relační databáze se používají v různých firemních aplikacích, např. na správu objednávek, databázi zaměstnanců atd. Ještě hojnější využití relačních db. najdeme u Internetových aplikací, ve spolupráci se skriptovacími jazyky PHP, ASP apod. Relační databáze je složená z řady tabulek, jejichž sloupce mohou být vázány na sloupce v ostatních souvisejících tabulkách, takto propojená datová pole jsou na sebe určitým způsobem závislá, mají mezi sebou nějaký logický vztah.

Základní pojmy:

entita – cokoliv, o čem potřebujeme v systému uchovávat informace

atributy – sledované skutečnosti týkající se dané entity

domény – popisují typ dat, obor hodnot = spojení datového typu a validačního pravidla

vztahy – jisté asociace mezi entitami, nemusí mezi určitými entitami existovat

Účastník – entita zapojená v určitém vztahu

Stupeň vztahu – počet účastníků (binární, ternární, ...) (výrobce – prodává – zákazníkovi – výrobky)

Vazby mezi tabulkami (kardinalita):

- Tabulky nejsou v relacích – jsou v nich nesouvisející údaje.
- Mezi tabulkami je vztah 1:1 – jednomu záznamu v jedné tabulce, odpovídá přesně 1 záznam v tabulce druhé.
- Mezi tabulkami je relace 1:N – jednomu záznamu v jedné tabulce, odpovídá více záznamů v druhé tabulce. Je to nejpoužívanější typ relace. 1 zákazník, může mít N objednávek. 1 typ zboží, může být obsažen v N objednávkách.
- Mezi tabulkami je relace M:N – více záznamům v jedné tabulce odpovídá více záznamů v tabulce druhé. V běžných databázových systémech nelze tuto relaci vytvořit přímo, proto se mezi takové tabulky vkládá ještě navíc jedna pomocná tabulka, která umožní vytvořit dvě relace 1:N a tím zajistit, že první dvě tabulky budou v požadované relaci M:N.

Parcialita/totalita vztahu – povinnost/nepovinnost existence role příslušné entity vztahu

vztah jednostranně parciální – zaměstnanec musí náležet k 1 pojišťovně, ale pojišťovna nemusí mít v evidenci žádného zaměstnance

vztah oboustranně parciální – zaměstnanec nenáleží k žádné pojišťovně, pojišťovna nemusí mít žádného zaměstnance v evidenci

Diagramy entit a vztahů

Pro návrh a zápis vztahů mezi jednotlivými entitami naší databáze byl vytvořen model *E-R diagramů*. Tento model byl zaveden a poprvé použit panem Peterem Pin Shan Chenem v roce 1976. Mluvil o diagramech entit a vztahů (Entity Relationship Diagrams), který se brzy rozšířil a stal se obecně uznávaným standardem. Více o ER diagramu v následující kapitole o UML, do jejíhož standardu v současnosti patří.

Normalizace

Normalizace ER modelu je sada pravidel, jak byste měli postupovat při transformaci struktury entit a relací ER modelu na strukturu fyzického uspořádání tabulek a relací v databázi. Proč normalizovat? Normalizace je odstranění redundantních (opakujících) se dat, omezení složitosti (rozložením složité relace na dvojrozměrné tabulky) a zabránění tzv. aktualizacím anomáliím (např. abychom smazáním všech knih autora nepřišli o data o autorovi). Což by mělo vést k databázi přehlednější, rozšiřitelnější a výkonnější. Normalizace by měla vést k vzniku tabulek, které lze snadno udržovat a efektivně se na ně dotazovat. Normalizované schéma musí zachovat všechny závislosti původního schéma a relace musí zachovat původní data, což znamená, že se musíme pomocí přirozeného spojení dostat k původním datům.

Normální formy:

1.NF – První normální forma

2.NF – Druhá normální forma

3.NF – Třetí normální forma

4.NF – Čtvrtá normální forma

5.NF – Pátá normální forma

1. normální forma (1.NF)

Relace je v první normální formě, pokud každý její atribut obsahuje jen atomické hodnoty. Tedy hodnoty z pohledu databáze již dále nedělitelné. Například v relaci obsahující data o nějaké osobě budeme chtít mít více telefonních čísel:

Jméno	Příjmení	Adresa	Telefony
Jan	Novák	Hlaváčkova 5	125789654;601258987;789456123
Petr	Kolář	Svatouškova 13	369852147;357951456;963852741
Pavel	Pavel	Papalášova 19	546789123;123456789;987456123

Tabulka č. 1 Tabulka nespĺňující 1.NF

S takovouto tabulkou by byla spousta problémů, například by se dost špatně prováděly změny čísel, případně vyhledávání podle telefonního čísla. Aby tabulka byla v 1NF musíme buďto rozdělit atribut telefon do více atributů (pouze za předpokladu, že jsme si jisti, že se množství telefonních čísel nezvýší), nebo oddělit telefonní čísla do samostatné tabulky, což já osobně preferuji, protože je to podstatně flexibilnější řešení:

ID	Jméno	Příjmení	Adresa
1	Jan	Novák	Hlaváčkova 5
2	Petr	Kolář	Svatouškova 13
3	Pavel	Pavel	Papalášova 19

Tabulka č. 2 Tabulka bez problémových telefonních čísel opatřená klíčem v podobě ID

ID osoby	Telefon
1	125789654
1	601258987
1	789456123
2	369852147
2	357951456
2	963852741
3	546789123
3	123456789
3	987456123

Tabulka č. 3 Tabulka telefonních čísel spojené pomocí klíče ID osoby

2. normální forma (2.NF)

Relace se nachází v druhé normální formě, pokud je v první normální formě a každý neklíčový atribut je plně závislý na primárním klíči, a to na celém klíči a nejen na nějaké jeho podmnožině. Ukážeme na příkladu:

V tabulce zboží v obchodě bude název zboží, výrobce, telefon na výrobce, cena zboží a množství na skladě.

Název	Výrobce	Telefon	Cena	Množství
Mléčná čokoláda	Milka	+420123456789	30 Kč	2500
Oříšková čokoláda	Milka	+420123456789	30 Kč	2800
Tyčinka Milkyway	Milka	+420123456789	15 Kč	7000
Mléčná čokoláda	Orion	+420987654321	25 Kč	5888
Oříšková horalka	Horalka	+420897651234	7 Kč	3500

Tabulka č. 4 Tabulka výrobků nesplňující podmínky 2.NF

Klíčem této relace je kombinace atributů Název a Výrobce. Telefon výrobce ovšem není závislý na celém klíči, ale pouze na atributu výrobce. To by vedlo k aktualizací anomálii, a to k té, že pokud by se vymazaly veškeré výrobky od výrobce Milka, ztratilo by se telefonní číslo na výrobce Milka, což není zrovna žádané. Řešením je opět rozpad na dvě tabulky:

Název	ID Výrobce	Cena	Množství
Mléčná čokoláda	1	30 Kč	2500
Oříšková čokoláda	1	30 Kč	2800
Tyčinka Milkyway	1	15 Kč	7000
Mléčná čokoláda	2	25 Kč	5888
Oříšková horalka	3	7 Kč	3500

Tabulka č. 5 Tabulka výrobků odkazující na výrobce pouze pomocí klíče ID výrobce

ID Výrobce	Výrobce	Telefon
1	Milka	+420123456789
2	Orion	+420987654321
3	Horalka	+420897651234

Tabulka č. 6 Tabulka výrobců s klíčem v podobě ID

3. normální forma (3.NF)

V této formě se nachází tabulka, splňuje-li předcházející dvě formy a žádný z jejích atributů není tranzitivně závislý na klíči. Jiné vyjádření téhož říká, že relace je v 3.NF, pokud je ve 2.NF a všechny neklíčové atributy jsou navzájem nezávislé. Opět definice, která zní nesrozumitelně, ale její použití je vlastně jednoduché. Tranzitivní závislost je

taková závislost mezi minimálně dvěma atributy a klíčem, kde jeden atribut je funkčně závislý na klíči a druhý atribut je funkčně závislý na prvním. Typickým učebnicovým příkladem porušení 3. normální formy je příklad s uložení města a poštovního směrovacího čísla do jedné tabulky. Aby taková tabulka dodržela 3.NF, musela by být rozdělena na dvě samostatné tabulky, v hlavní z nich by byly všechny potřebné údaje, ovšem z údajů o bydlišti pouze PSČ a v druhé tabulce by byly všechny příslušné PSČ a měst.

Čtvrtá normální forma (4.NF)

Tabulka je ve čtvrté normální formě, je-li v BCNF a popisuje pouze příčinnou souvislost (jeden fakt). Sice jednoduché vyjádření bez složitých definic, ale poněkud nic neříkající, takže zkusíme jinou definici: „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 (zjednodušeně: v jedné relaci se nesmí spojovat nezávislé opakované skupiny).“

Pátá normální forma (5.NF)

Relace je v páté normální formě, pokud je ve čtvrté a není možné do ní přidat další atribut (skupinu atributů) tak, aby se vlivem skrytých závislostí rozpadla na několik dílčích relací.

Shrnutí

Normalizovat je určitě potřeba a čím složitější databáze a čím více dat, tím více je potřeba normalizovat. Ale i tady platí všeho s mírou. Například u příkladu u 3.NF by firma s několika desítkami zaměstnanců asi neměla potřebu dávat PSČ do další tabulky a bylo by to zbytečné. Ostatně přesně tak je učiněno v praktických kapitolách této práce, kdy jsem v návrhu databáze často 3.NF nedodržel.

2.2.2 UML

Unified Modeling Language je grafický jazyk vytvořený pro vizualizaci, specifikaci a navrhování programových systémů. UML nám umožňuje jak vytvořit přímo konkrétní

návrh systému, tak i dílčí části projektové analýzy v podobě zápisu business procesů a dalších systémových funkcí.

UML podporuje objektivě orientovaný přístup k tvorbě softwaru. Zároveň ale neobsahuje popis jak se má používat, ani metodiku, jak analyzovat či navrhovat informační systémy. Toto dává obrovské tvůrčí možnosti do rukou analytiků, kteří nejsou svázáni přesným popisem, co musí či nemusí pro každý projekt vytvořit.

Velkou výhodou UML jazyka je, že se může chlubit bohatou podporou ze strany velkých softwarových firem, mezi které patří například společnosti IBM, Oracle, Microsoft a další. V současné době se o standard UML 2.0 stará standardizační skupina Object Management Group (OMG).

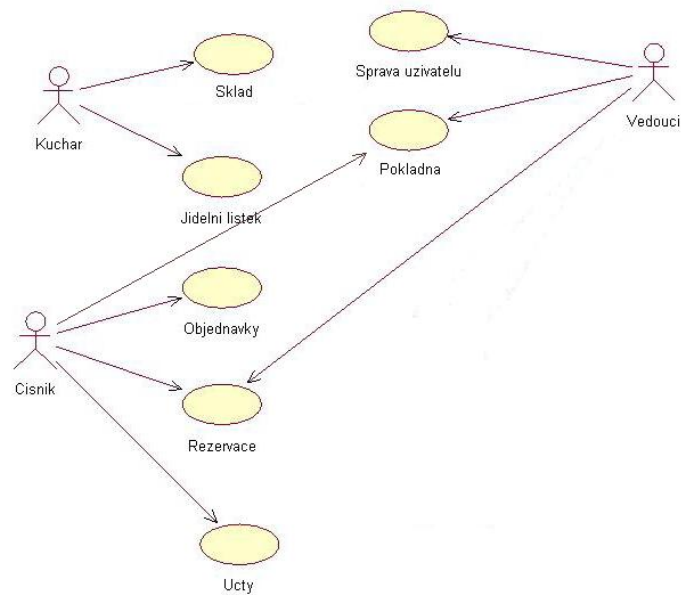
2.2.2.1 Historie UML

Jazyk UML vznikl v roce 1994 ve společnosti Rational software, která nyní patří do skupiny IBM. Sloužil pro objektivé návrhy softwaru vytvářeného touto společností. Prvním výstupem vývoje UML byl UML verze 0.9 a metodika RUP (Rational Unified Process). V roce 1997 přijala UML jako standard skupina OMG. V současné době je aktuální standard UML 2.0, jež obsahuje výrazně více rozšíření než její předchůdci.

2.2.2.2 Součásti jazyka UML

Use Case diagram

Use Case diagram je nyní součástí specifikace jazyk UML 2.0. Avšak nebylo tomu vždy tak a tento model byl do standardu začleněn později. Diagram popisuje vztahy mezi externími uživateli (Actors) a prvky budoucího systému.



Obrázek č. 3 Příklad Use Case diagramu - návrh systému restaurace

Activity diagram

Diagram aktivit se používá pro popis dynamických struktur uvnitř systému. Svou specifikací připomíná klasický vývojový diagram a popisuje průběh v čase. Používá se také k modelování business procesů a workflow.

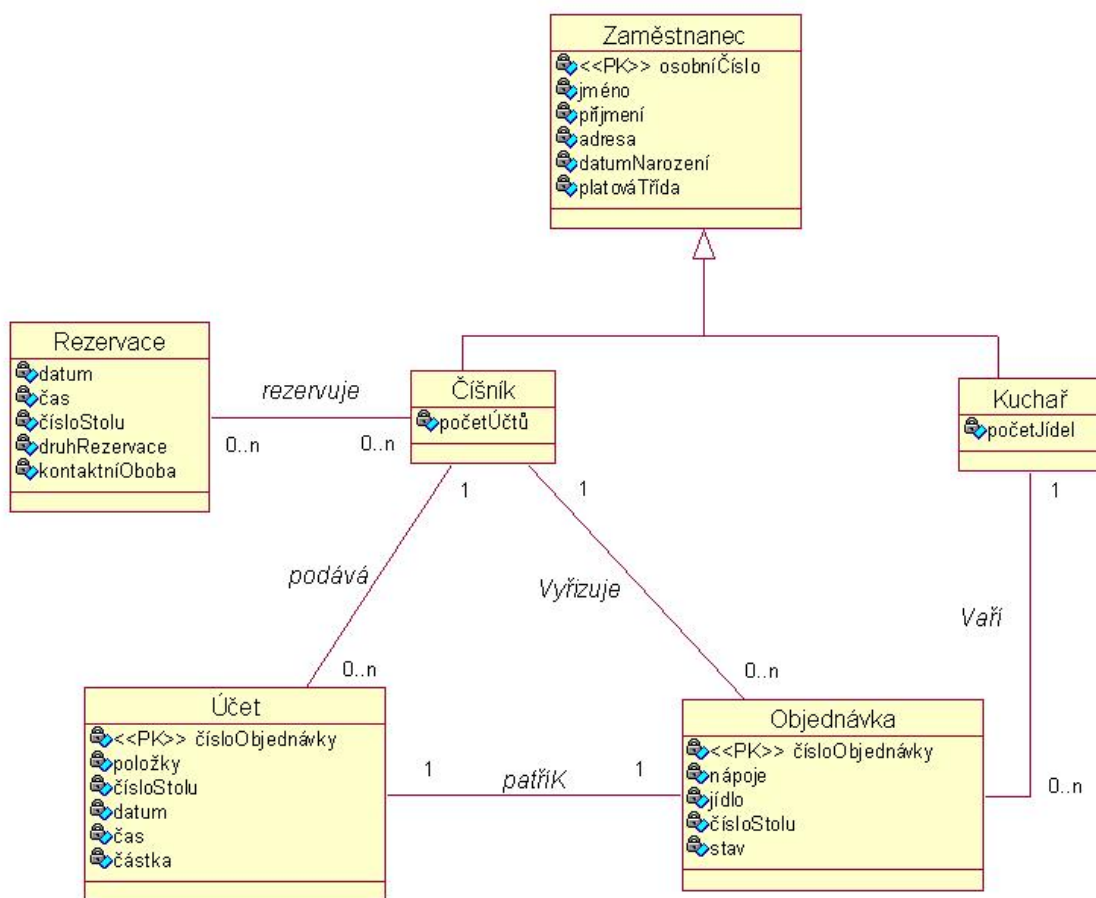
Prvky diagramu aktivit:

- **aktivita (activity)** vyjadřuje aktivitu systému nebo interakci uživatele,
- **oddíly (partitions, swimlanes)** umožňují rozdělení diagramů na více částí pro znázornění odpovědností za různé části aktivity,
- **hrana, tok (edge, flow)** zobrazuje přechod z jedné aktivity do další,
- **objekt (object)** může vstupovat do aktivit, může být jejich výsledkem,
- **počáteční, finální uzel aktivity (initial, activity final)** speciální uzly pro označení počátku a konce aktivity.

Entity-Relationship Diagram (ERD)

ERD se v softwarovém inženýrství používá jako abstraktní a konceptuální model pro vyjádření dat. ERD modelování vytváří relační schéma, obvykle návrh relačního schématu databáze systému.

V diagramu se vyskytují entity – prvky systému. Pro jejich specifikaci slouží atributy. Vzájemným propojením jednotlivých entit získáme představu o návaznosti jednotlivých prvků v databázi systému. Zároveň se propojení přiřazuje kardinalita jednotlivých entit, která nese informaci o mocnosti jednotlivých propojených entit.



Obrázek č. 4 Příklad ER Diagramu - návrh databáze systému restaurace

2.2.3 Byznys procesy – BPMN

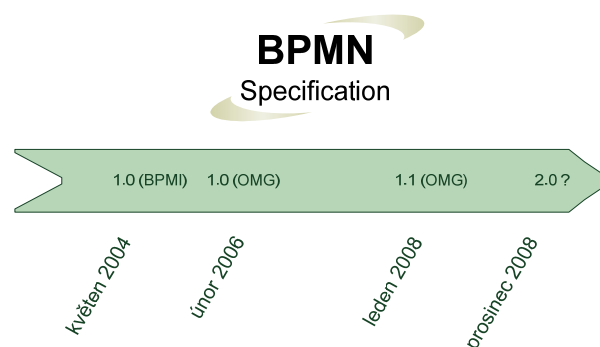
Business Process Modeling Notation (BPMN) je standardizovaná grafická reprezentace pro kreslení obchodních procesů ve firemní sféře. BPMN se snaží zjednodušit čitelnost pro všechny účastníky životního cyklu procesu, ať už se jedná o vrcholné manažery, systémové analytiki, business analytiki nebo vývojáře.

2.2.3.1 Vznik

Za vznikem této notace stála iniciativa BPMI (Business Process Management Initiative), která se snažila zacelit komunikační díru mezi návrhem a následnou implementací procesu. Původně vyvinutý jazyk BPML (jazyk založený na XML) potřeboval pro svou realizaci grafickou podobu, jímž se právě BPMN stala. Jazyk BPML byl později nahrazen jazykem BPEL určeným výhradně pro realizaci byznys procesu. BPMN notaci následně v roce 2006 přebírá OMG (Object Management Group), respektive BPMI se stává součástí OMG.

2.2.3.2 Verze

Aktuální verze BPMN je 1.1, která byla oficiálně přijata v lednu loňského roku (2008). Od původní 1.0 se však příliš neliší a právě proto je zatím daleko rozšířenější její předchůdkyně z února 2006. V chystané specifikaci 2.0, která má být představena koncem roku, by se měly projevit zásadnější změny.



Obrázek č. 5 Vývoj verzí specifikace BPMN

2.3 Implementační část

V této kapitole přiblížím implementační nástroje, na kterých bude celý systém postaven. Jedná se především o skriptovací jazyk PHP. Pro zobrazení výstupu jsem použil klasické HTML kódování obsahu nastýlované pomocí kaskádových stylů (CSS), a to vše podpořené databázovým systémem MySQL. V některých případech bylo nutné použít Java skriptování, například pro kontrolu správnosti zadaných hodnot, ale tou se v této práci zabývat nebudu.

2.3.1 PHP

PHP je další z řady skriptovacích jazyků určený především pro tvorbu internetových aplikací. Jeho zkratka na počátku znamenala Personal Home Page, postupně se však z tohoto jazyka stal mocný nástroj, který může bez nadsázky konkurovat rozsáhlým projektům jako je například ASP od Microsoftu. Proto byl upraven význam zkratky na Hypertext Preprocessor.

Obrovskou výhodou PHP je, že se jedná o open source produkt, který je na internetu volně šířen a má rozsáhlou komunitní podporu. K oblíbenosti jazyka taky přidává fakt, že PHP podporuje syntaxi všech nejznámějších jazyků, z nichž zmíním hlavně C/C++ a Javu. Od nových verzí 5.0 a výše se v PHP objevuje také objektový přístup, což jen umocňuje použitelnost tohoto mocného nástroje.

Součástí jazyka je také velké množství předchystaných funkcí na podporu klasických činností, které využijete téměř v každém rozsáhlejší projektu. Musím zmínit alespoň funkce pro práci s různými databázovými systémy (v projektu vystačím s MySQL), funkce pro spolupráci s mailserverem, uploadování souborů na server a další.

Shrnutí výhod PHP

- PHP je relativně jednoduché na pochopení
- PHP podporuje širokou řadu souvisejících technologií, formátů a standardů
- Je to otevřený projekt s rozsáhlou podporou komunity

- Dají se najít kvanta již hotového kódu k okamžitému použití nebo funkční PHP aplikace. Podstatná část z hotového kódu je šířena pod nějakou svobodnou licenci a dá se použít ve vlastních projektech
- PHP si dobře rozumí s webovým serverem Apache
- PHP snadno komunikuje s databázemi, jako je MySQL, PostgreSQL a řada dalších
- PHP je multiplatformní a lze jej provozovat s většinou webových serverů a na většině dnes existujících operačních systémů
- PHP podporuje mnoho existujících poskytovatelů webhostingových služeb
- Dobrá podpora objektového přístupu

2.3.2 HTML a CSS

Hypertext Markup Language, který se označuje zkratkou HTML, je značkovací jazyk určený pro kódování dokumentu šířeného pomocí sítě internet.

2.3.2.1 Vývoj značkovacích jazyků

Potřeba mít jazyk umožňující popsat strukturu a obsah textu bez jeho grafické podoby vznikla hlavně rozvojem internetu. Kde více je potřeba, aby stejný obsah bylo možné zobrazit v různých grafických úpravách, případně jako čistý text pro účely tisku. Svým způsobem to již umožnil jazyk HTML (HyperText Markup Language), ale tento jazyk obsahuje spoustu nevýhod, které právě Extensible Markup Language (XML) nemá. Největší nevýhodou je absence možnosti vytvářet vlastní značky.

Asi prvním známým značkovacím jazykem byl GML (Generalized Markup Language), který si vytvořil tým lidí z IBM pro lepší ukládání a následné využití právních textů uvnitř společnosti. Za přispění organizací ANSI a GCA (Graphics Communications Association), jež měly podobné cíle, vznikl jazyk SGML (Standard Generalized Markup Language), který byl definován již v normě ISO z roku 1986.

Standard SGML byl opravdu hodně obecný, podporoval volbu vlastních značek i mnoho dalších předvoleb. Asi nejznámějším využitím je právě jazyk HTML, který pomocí DTD (definice typu dokumentu) určoval značky, které může dokument obsahovat. U HTML se typ dokumentu měnil s každou novou verzí.

2.3.2.2 Terminologie HTML jazyka

- **Tag** základní prvek HTML jazyka, zapisovaný do ostrých závorek: $\langle \rangle$.
- **Značka** interpretace tagu v HTML zápisu.
- **Element** prvek stránky představený tagem. Jde o tag samotný i jeho obsah (to, co je mezi úvodní a koncovou značkou).
- **Atribut** parametr tagu, zapisovaný ještě uvnitř ostrých závorek nastavuje elementu nějaké vlastnosti.
- **Hodnota atributu** to, co je za rovnítkem v uvozovkách. Pokud hodnota neobsahuje mezeru, jsou uvozovky nepovinné.
- **Obsah tagu** to, co je mezi úvodní a koncovou značkou.

Symbolický zápis tagu:

`<tag atribut="hodnota atributu" jinýatribut=hodnota>Obsah tagu</tag>`

2.3.3 MySQL

MySQL je databázový systém vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael Widenius a David Axmark. Zároveň se řadí mezi průkopníka ve dvojím licencování. MySQL je totiž k dispozici jak zdarma pod GPL licencí, tak i pod komerční placenou licencí.

Databázový systém MySQL je multiplatformní, což značí využití na všech běžně používaných operačních systémech pro podporu různých webových serverů (Apache, IIS a dalších). Komunikace probíhá pomocí příkazů jazyka SQL s lehkým dialektem a drobnými rozšířeními.

Pro svou snadnou implementovatelnost, výkon a především díky tomu, že se jedná o volně šířitelný software, má vysoký podíl na aktuálně používaných databázích. Zároveň patří do oblíbeného trojlístku technologií pro webové servery: MySQL, PHP a Apache.

MySQL bylo od počátku optimalizováno pro výkon. Stěžejním kritériem byla rychlost, a to i za cenu některých zjednodušení. MySQL má jen jednoduché způsoby zálohování a zároveň až donedávna nepodporovala pohledy, trigger a uložené

procedury. Toto se ale již na popud programátorů změnilo a současné verze MySQL podporují vše podstatné, co najdete například v komerčním MSSQL od Microsoftu.

2.3.4 Zabezpečení

„Na stránkách organizace OWASP můžete najít seznam deseti nejzávažnějších bezpečnostních problémů za rok 2004, které se vyskytují v internetových aplikacích.

1. Nekontrolovaný vstup dat

Informace, které aplikace získá z HTTP požadavku, nejsou kontrolovány před použitím v aplikaci. Útočník může využít této chyby k útoku na webovou aplikaci.

2. Narušená kontrola přístupu

Omezení, co vše je povoleno autentizovanému uživateli, není správně zajištěno. Útočník může této chyby využít k přístupu na účty ostatních uživatelů, zobrazovat citlivé soubory nebo použít nepovolené funkce.

3. Narušená autentikace a správa session

Pověření k účtu a informace o přihlášení nejsou řádně chráněna. Útočník může kompromitovat hesla, klíče, session cookies nebo další tokeny. Může narušit bezpečnostní omezení a zjistit identity ostatních uživatelů.

4. Cross Site Scripting (XSS) chyba

Webová aplikace může být použita jako mechanismus pro přenesení útoku přímo do internetového prohlížeče připojeného uživatele. Úspěšný útok může odhalit přihlašovací údaje uživatele, umožnit útok na uživatelův počítač nebo podvrhnout obsah stránky k oklamání uživatele.

5. Přetečení vyrovnávací paměti (Buffer Overflow)

Komponenty webových aplikací v některých jazycích nekorektně kontrolují vstupní data. Může dojít k jejímu zhroucení a někdy i následné kontrole běžícího procesu. Tyto komponenty mohou zahrnovat CGI, knihovny, ovladače a komponenty webového serveru, na kterém běží aplikace.

6. Chyba umožňující vkládání kódu

Webová aplikace používá zasílané parametry k přístupu na externí systémy nebo k operačnímu systému. Pokud útočník dokáže tyto parametry pozměnit a připojit vlastní kód, externí systém tyto příkazy spustí s oprávněními serveru.

7. Nesprávné ošetřování chyb

Chybové podmínky, které nastanou za běhu aplikace, nejsou korektně ošetřeny. Pokud může útočník vyvolat nějaké chyby, které aplikace neošetřuje korektně, může se dostat k detailním informacím o celém systému, zakázat celou službu, obejít bezpečnostní mechanismus nebo způsobit pád serveru.

8. Nezabezpečené úložiště dat

Webové aplikace často používají kryptografické funkce k ochraně informací nebo přihlašovacích údajů. Správná implementace je poměrně složitá, proto většinou dochází k oslabení této ochrany díky nesprávnému použití.

9. DOS útok (Denial of Service)

Útočník může spotřebovávat aplikační zdroje, aby další oprávnění uživatelé nemohli službu nadále používat nebo k ní přistupovat. Útočník může uživatelům dokonce zamezit přístup k jejich účtům a nebo zavinit selhání celé aplikace.

10. Nezabezpečená konfigurační správa

Velké konfigurační nároky na server mohou mít špatný vliv na zabezpečení webové aplikace. Mnoho konfiguračních možností ovlivňuje i bezpečnost aplikace v případě špatného nastavení.“ [2]

3 Analýza současné situace

Tato kapitola volně navazuje na kapitolu 1.2 této práce, ve které popisují aktuální situaci. Pro vývoj informačního systému řešící konkrétní případ podnikatelské činnosti je nezbytně nutné dobře zmapovat aktuální situaci.

3.1 Citlivé údaje

Proto se v této kapitole pokusím popsat současnou situaci při řešení konkrétních zakázek v mém oboru podnikání – tedy poskytování IT služeb. Stěženi formou zadávání zakázek je osobní či emailová komunikace. Za základě dohody se stanoví rozsah služeb a termíny dodání. Mezi drobné zakázky patří například objednání a nastavení nových technologií, například nových notebooků. Pokud se nakupuje pro klienta nový notebook, je zapotřebí ho objednat, následně vyzvednout na prodejně nebo ho nechat dopravit přepravní službou. Tím ale práce nekončí. Klient vyžaduje mít nástroj plně připraven k používání, je tedy potřeba nainstalovat operační systém, další standardní programové vybavení a dále ještě konkrétní software odlišný podle konkrétního budoucího uživatele. Při instalacích je nutné nastavit také příjem firemní pošty a připojení na bezdrátové síť klienta. Tyto informace musí mít k dispozici člověk, který takovou přípravu notebooku provádí. Pracovník si musí tyto informace vyžádat ode mě a já mu je zašlu nebo jinak předám. Citlivé údaje jako přístupové údaje k emailovým schránkám klientů nebo síťového klíče by neměli být ukládány na lokálním disku mého počítače a předávány emailem nebo jiným způsobem, kdy se přenáší v čisté textové formě. Hrozí tím odposlech sítě a možná ztráta citlivých údajů.

Zároveň jediné možné úložiště přístupových údajů na počítači každodenní potřeby také není rozumné řešení. Hrozí odcizení celého počítače, havárie disku a podobně. Pro případ zničení hardwaru jsou sice vytvářeny zálohy, ale nejsou pravidelné a tyto přístupové údaje se mohou často měnit.

3.2 Efektivnost

V předchozí kapitole jsem popsal současný problém z pohledu riskantních operací s citlivými údaji. Zde bych rád popsal neefektivnost s postupem instalací obdobných systémů a aktualizací různých IT zařízení, jakou jsou tiskárny, routery a podobně. Například zálohy nastavení síťových zařízení je potřeba pravidelně provádět a někde archivovat. Toto se v současné době děje pouze ukládáním na lokální disk mého počítače a každý kdo potřebuje zálohu upravit nebo nahrát zpět do zařízení, si ji musí ode mě vyžádat. Toto řešení je značně neefektivní, neexistuje žádný záznam, kdo se zálohou pracoval a podobně.

Zajímavé je také sdílení manuálů a technických textů k jednotlivým zařízením. Toto je také ukládáno na lokálním disku a bývá často jednodušší si tyto informace znovu najít na stránkách výrobce než hledat tyto texty na lokálním disku, který ani nemusí být vždy po ruce.

3.3 Spolupráce v týmu

V současné době většinu zakázek vyřizují osobně a externích pracovníků využívám jen na větší akce, kdy by byla činnost v jednom člověku značně zdlouhavá. Často ale i klient požaduje dodání služby ve zkráceném čase a zde přichází nutnost spolehnout se na pomoc další pracovní síly.

V souvislosti s tímto je nutné sdílet informace o jednotlivých službách s těmito pomocnými pracovníky. V současné době se údaje sdílejí pomocí flash disků, emailem a podobnými neefektivními kanály.

3.4 Shrnutí

Popisované skutečnosti jasně napovídají, že jediným možným řešením je zavedení centrálního řešení sdílení informací mezi více pracovníky. Zároveň je nutné zabezpečit všechny cesty přístupu k těmto informacím tak, aby nemohly být odposlechnuty

a zneužity nebo ztraceny. Je nutné se zabývat pravidelným zálohováním a archivováním činností, které jsou s těmito daty prováděny.

4 Vlastní návrh

Kapitola se bude věnovat samotnému průběhu praktické části této práce. Jako první činnost v kterémkoliv projektu zabírá sestavení časového rozvrhu jednotlivých fází. Bezprostředně po něm budou následovat další kroky analýzy a návrhových řešení.

4.1 Sestavení časového plánu projektu

V samotném počátku projektu bylo zapotřebí sestavit časový plán jednotlivých etap. Jednotlivé etapy vycházejí z toho, že projektem bylo vytvořit softwarové dílo v jednom člověku (chybí týmová spolupráce). Projektový tým se tedy skládal pouze z řešitele a vedoucího práce.

4.1.1 Časový plán

- Zahájení projektu
 - 1. 12. 2008
- Ustanovení tématu, podmínky, požadavky
 - Do 12. 12. 2008
- Zajištění studijních materiálů
 - Do 31. 12. 2008
- Studium materiálů
 - Do 31. 1. 2009
- Analýza požadavků
 - Do 14. 2. 2009
- Navržení modulů systému
 - Do 27. 2. 2009
- Vypracování specifikací jednotlivých modulů
 - Do 15. 3. 2009
- Návrh databáze systému
 - Do 31. 3. 2009
- Implementace
 - Do 5. 5. 2009

- Testování
 - Do 15. 5. 2009

4.2 Analýza požadavků

Zadáním bylo vytvořit více-uživatelský informační systém pro správu a vedení informacích o klientech a službách, které jsou jim poskytovány. Službami se v tomto modelu rozumí přístupy na různé webové portály, spravované hardwarové zařízení nebo i konzultační služby k různým projektům. U těchto služeb je nutné ukládat jejich přesné nastavení, údaje pro přístup, zálohy nastavení. Zároveň by ke každé položce služby měl být přiložen aktuální manuál ke stažení. Tyto služby by měly být pravidelně aktualizované jednotlivými zaměstnanci, kteří je budou mít na starosti.

Systém bude sloužit pro potřeby mé osoby a dále všem mým pracovníkům (stavajícím i budoucím), kteří mají přidělené jednotlivé klienty, o jejichž zařízení a služby se starají. Objednávky služeb bývají zpravidla na základě osobní či emailové komunikace a nemívají podobu oficiálního dokumentu. Z tohoto důvodu vyžadují, aby součástí systému byl modul pro archivaci vzájemné komunikace mezi mými pracovníky a zástupci klienta. Dále je potřeba uchovávat důležité informace o klientech, kontaktních osobách, službách a nesmí chybět modul pro uchování informací o pracovnících, kteří mají přímý přístup do systému. Dále bych rád do systému zavedl modul pro fakturaci z důvodu nevyhovujícího freewarového nástroje, který doposud používám. Položky faktury se budou vytvářet samostatně a jednotlivé faktury budou navázány na klienty. Vytvořené faktury bude možné buď vytisknout lokálně a předat v papírové podobě (s možností archivace), ale také zaslat elektronicky v podobě el. odkazu přímo klientovi.

4.3 Sestavení modulů budoucího IS

Na základě požadavků jsem vytvořil seznam potenciálních modulů systému. Přikládám popis jednotlivých modulů:

Modul Klienti

Bude sloužit pro přehled a uchování veškerých obecných informací o klientech. Jedná se zejména o kontaktní údaje, bankovní spojení, poštovní a elektronické adresy, fakturační údaje a podobně. Na základě vložení do systému bude každému klientovi přiděleno identifikační číslo (ID) a k němu budou dále přiřazovány prvky dalších modulů.

Modul Kontaktní osoby

Tento modul je úzce spojen s předchozím modulem Klienti. Přiřazuje každému klientovi neomezené množství kontaktních osob a umožňuje o takové osobě uchovávat další cenné informace, zejména kontaktní údaje nebo například funkci, kterou v organizační struktuře klienta zastupuje.

Komunikace

Modul komunikace bude sloužit pro uchování vzájemné komunikace mezi kontaktní osobou klienta a zaměstnancem, který bude přihlášen do systému. Komunikace bude mít podobu položky v databázi popisující kdy, kdo, jakým způsobem – telefonická komunikace, email, písemnost, osobní jednání – komunikace probíhala.

Faktury

Fakturace je navrhována do systému jako určité rozšíření. Celý systém není navržen jako ekonomický informační systém, který by řešil podstatné problémy funkčního charakteru, jako jsou náklady, příjmy, přehledy výsledků a podobně. Systém má sloužit především jako podpora spravování jednotlivých služeb klientů. Ale vzhledem k tomu, že vystavování faktur za provedenou činnost klientům je v úzké vazbě na informace uložené v tomto systému, jeví se přidání modulu pro vystavení, distribuci a zálohování faktur, jako naprosto logické a efektivní řešení.

Zaměstnanci

Modul zaměstnanci bude sloužit pro uchování kontaktních údajů a přístupových práv pro své zaměstnance a ostatní pracovníky, kteří budou mít přístup do systému. Přístup k tomuto modulu bude mít pouze uživatel s administrátorskými právy.

Služby

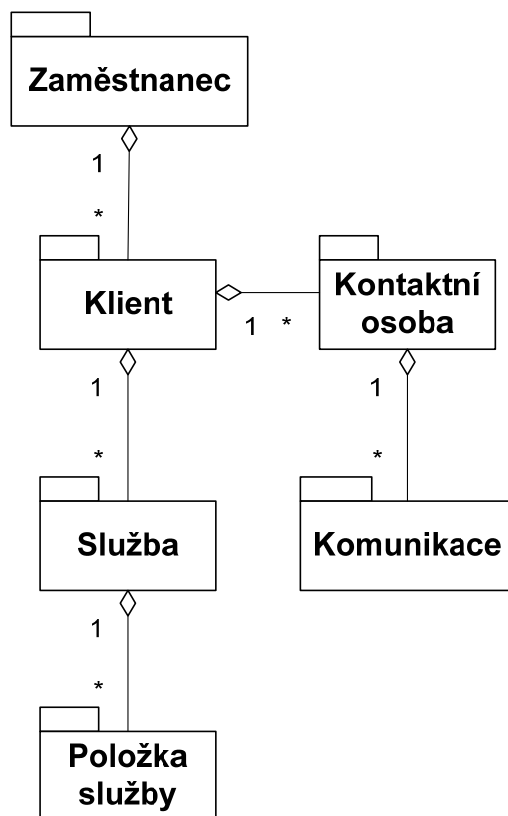
Modul služby slouží pro určení struktury služeb pro jednotlivé klienty. Každý klient může mít přiřazený libovolný počet služeb, jimiž mohou být webové portály klienta, počítačová síť klienta, nebo například celá pobočka klienta, kterou máme pod správou z hlediska ICT. V posledním případě by potom položky služby obsahovaly jednotlivé počítače na pobočce, tiskárnu nebo internetové připojení, telefon a podobně.

Položky služby

Popis položky služby jsem již naznačil v popisu modulu služeb. Jedná se o jednotlivé položky každé služby. Položky služeb různých klientů se mohou významově lišit a za jejich strukturu ponese odpovědnost vždy zaměstnanec, který má danou službu konkrétního klienta na starosti.

4.3.1 Napojení budoucích modulů

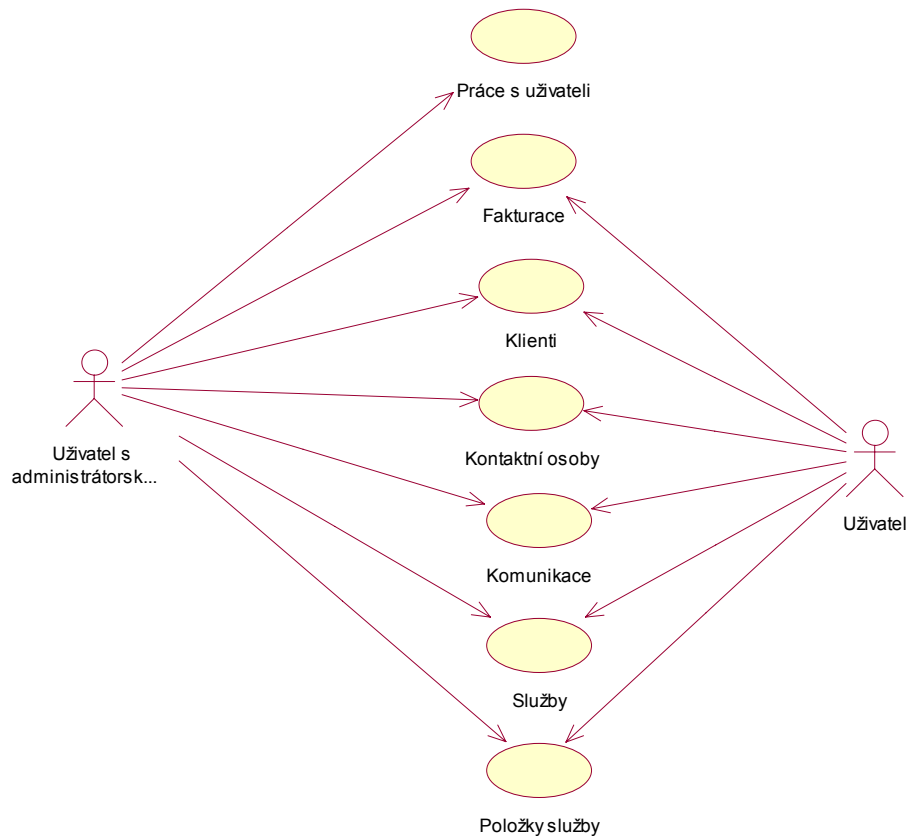
Pro lepší ilustraci problému jsem návrh diagram napojení modulů systému. Tento diagram by se dal připodobnit k diagramu komponent UML, ale protože nesplňuje všechny jeho specifika, nebudu ho v této práci takto nazývat. U jednotlivých vazeb je určena kardinalita, tak jak bude odpovídat v reálu.



Obrázek č. 6 Model napojení modulů systému

4.4 Model užití systému

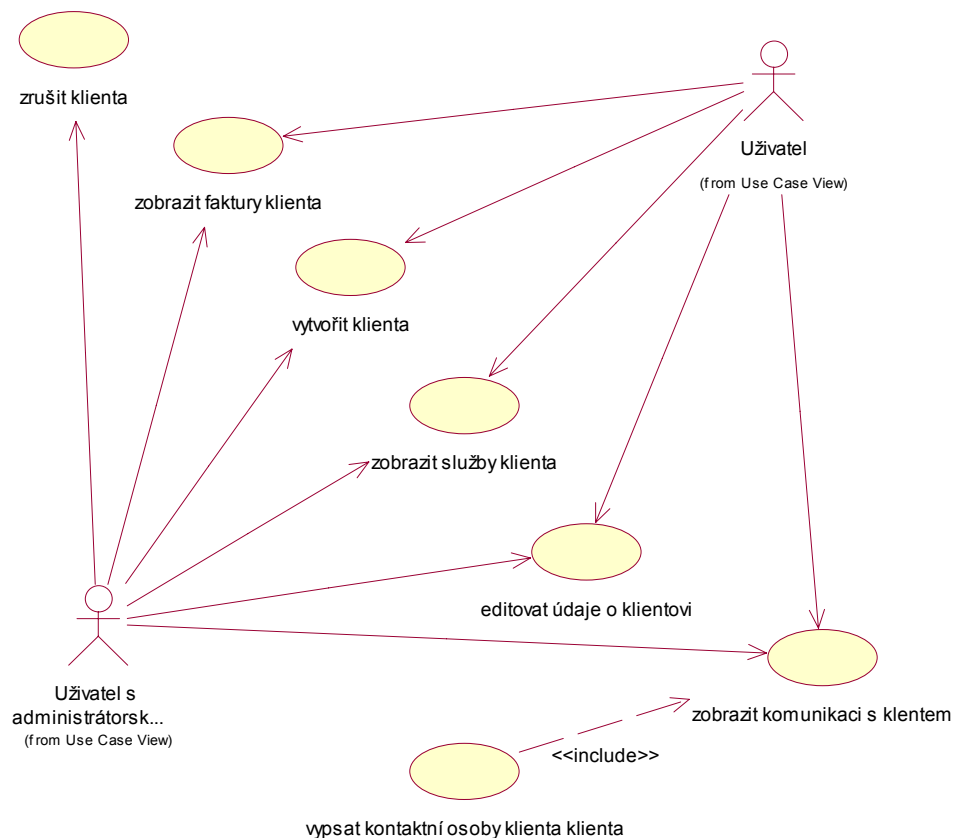
Na základě analýzy požadavků na budoucí systém jsem vytvořil několika úrovněvý model užití – Use Case model. První úroveň, která popisuje hrubý pohled na systém ze strany budoucích uživatelů, můžete vidět na obrázku 6. Tento diagram první úrovně určuje hranice systému. Diagramy nižší úrovně popisuje hranice každého modulu.



Obrázek č. 7 Use Case model systému - nejvyšší úroveň

Na diagramu jsou znázorněny dvě role uživatelů, které ale nemají v systému vysoký význam. Tento diagram popisuje, k jakým modulům budou přistupovat jaké typy uživatelé (v Use Case modelu hovoříme o aktorech). My tedy máme dva aktory, kteří se liší pouze oprávněním k přístupu k modulu pro správu zaměstnanců a jejich přístupu do systému. V praxi by měl být pouze jeden jediný uživatel s administrátorskými právy a libovolný počet ostatních uživatelů.

Jednotlivé případy užití korespondují s popisem jednotlivých modů v kapitole o návrhu modulů systému. Pro náhled, jak vypadají Use Case diagramy nižší úrovně, které specifikují jednotlivé případy užití, uvedu pouze na příkladu, konkrétně na případu užití klienti.



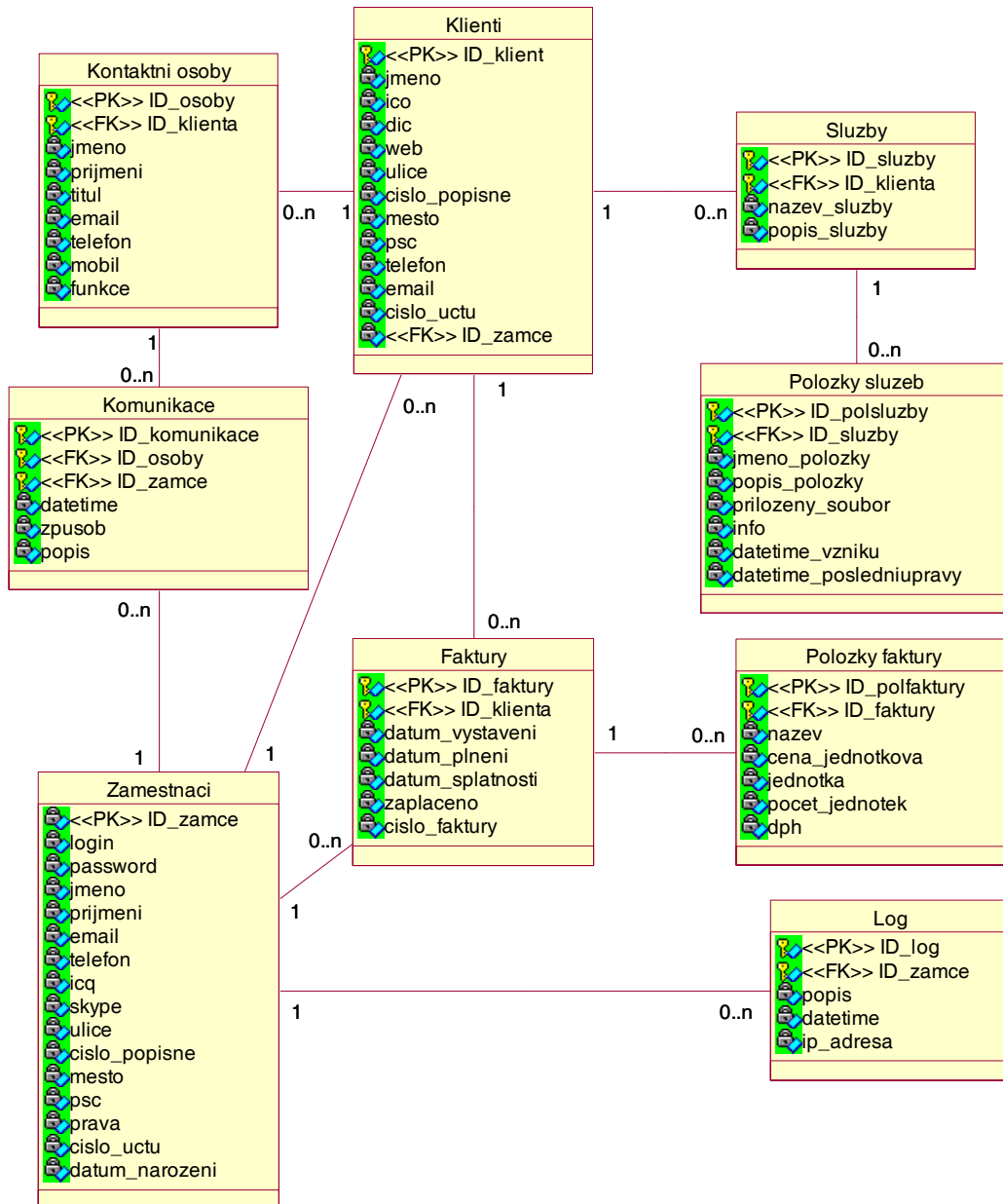
Obrázek č. 8 Use Case model - Use Case modulu klient

Na diagramu znázorněném na obrázku č.7 můžeme vidět „ohraničení“ modulu klient. Jednotlivé případy užití popisují, co může daný uživatel v systému provést. Speciální spojení include značí činnost, která je bezpodmínečně nutná k provedení předchozího případu užití, v tomto případě k výpisu komunikace s klientem. Toto je dáno přímou závislostí komunikace na kontaktní osoby, nikoli však na klienta.

4.5 Model databáze – ER diagram

Na základě analýzy požadavků, definice jednotlivých modulů a případů užití, jsem sestavil návrh databáze pro informační systém. Databáze je navržena tak, aby splňovala všechny zákonitosti nezbytné pro tvorbu obdobných systémů, zejména je kladen důraz na minimalizaci redundance dat. Tabulky jsou navrženy v minimální možné formě.

Zároveň je ale kladen důraz na rychlost přístupu k datům uloženým v databázi, proto některé části nedodržují zákonitosti Boyce/Coddově normální formy. Bylo by velmi neefektivní například mít samostatnou tabulku měst s klíčem v podobě poštovního směrovacího čísla.



Obrázek č. 9 ER Diagram informačního systému

Na obrázku č.6 vidíme ER Diagram navrhovaného informačního systému. Hlavní výchozí bod systému tvoří tabulka Zaměstnanci. Jedná se o tabulku s atributy popisujícími jednotlivé uživatele, kteří budou do systému vstupovat. Proto jejich součástí je uživatelský login a heslo. Heslo se bude v databázi vyskytovat ve formě hashe¹ vytvořeného pomocí hashovací funkce MD5². Dále je potřeba ukládat kontaktní údaje, číslo účtu pro odeslání odměny a podobně. Při tvorbě návrhu jsem vycházel ze zkušeností a obecně platných postupů.

Na tabulku Zaměstnanců je pomocí cizího klíče napojena tabulka klientů s kardinalitou 0 až N což znamená že každému zaměstnanci může být přiřazeno 0 až N klientů. Atributy klientů zase vycházejí z informací, které si o nich potřebujeme uchovávat.

Další tabulky nebudu podrobněji popisovat. Jedná se o tabulku faktury, která je propojena jak s klientem, pro kterého je vystavena, tak se zaměstnancem, který ji vystavil. Faktura má dále své položky faktury. Kontaktní osoby jsou propojeny pomocí cizího klíče ID_klienta s klientem. Tím se dává jasně najevo, že kontaktní osoby jsou navázány na klienta. Komunikace je provázána s kontaktní osobou (za stranu klienta) a se zaměstnancem z naší strany. Na klienta je dále napojena tabulka, která řeší účel tohoto systému, a to konkrétně tabulka služeb a k ní tabulka položek služeb. Kardinalita zde vychází ze zadání požadavků na systém, a to 1 klient – N služeb, 1 služba – N položek.

Jako speciální tabulku ER diagram obsahuje tabulku Log, která bude sloužit pro logování veškerých událostí v systému. To umožní zpětnou kontrolu zaměstnanců a může pomoci při odhalení vzniklých problémů. Do logu se budou ukládat veškeré uživatelské akce, které byly nad systémem vykonány. Například tedy přihlášení uživatele, vytvoření faktury a podobně. Každá změna položky databáze nebo nalogování uživatele do systému bude uloženo a díky tomu bude existovat možnost

¹ HASH – speciální otisk daného vstupního parametru do neidentifikovatelné podoby.

² MD5 – **Message-Digest algorithm** je rozšířená rodina hashovacích funkcí, která vytváří ze vstupních dat výstup (otisk) fixní délky. Otisk je též označován jako miniatura, kontrolní součet (v zásadě nesprávné označení), fingerprint, hash (česky někdy psán i jako haš). Jeho hlavní vlastností je, že malá změna na vstupu vede k velké změně na výstupu, tj. k vytvoření zásadně odlišného otisku.[4]

zjistit, kdo provedl změnu čeho a hlavně kdy k tomu došlo. Log je díky cizímu klíči zaměstnance spojen s konkrétním zaměstnancem, který změnu provedl.

4.6 Návrh zabezpečení systému

Na bezpečnost informačního systému se musíme dívat z několika pohledů. Zmíním zde několik pojmů, které s bezpečností dat souvisejí.

4.6.1 Důvěrnost dat

Jedná se o základní atributy zabezpečení následujících aplikací:

- identifikace - každý uživatel je jednoznačně identifikován,
- autentizace - uživatel prokáže svoji totožnost (uživatelským jménem a heslem),
- autorizace - každý uživatel je oprávněn k úkonům odpovídajícím roli, kterou zastává.

4.6.2 Integrita dat

Jedná se zejména o zpracování dat vhodně zvolenými technickými prostředky tak, aby uložená data byla neustále dostupná v plné míře.

4.6.3 Dostupnost služeb

Je nutné vyvarovat se problémům s výpadky elektrické sítě nebo s výpadkem konektivity serveru. Často se používá metoda duplikace hardwaru pro případ poruchy fyzického stroje. V současné době vám nikdo nenabídne 100% dostupnost služeb, ale relativní hodnoty se pohybují okolo hodnoty 99% dostupnosti služeb.

V souvislosti se službami ještě zmíním termín důvěrnost služeb, jež musí předcházet případným phishingovým³ útokům. Je třeba zajistit důvěru uživatelů, že vstupují do systému, který je opravdu ten jejich.

4.6.4 Ochrana před zneužitím dat

Riziko vnitřní

Je velmi dobré, když máte vlastní aplikaci dokonale zabezpečenou proti všem vnějším zásahům a potencionálním útočnickům, ale co když si váš zaměstnanec nechá přístupové údaje napsané na nástěnce v kanceláři? Nebo nedej bože sdělí svoje přístupové údaje telefonem kolegovi, kterému se nedaří přihlásit. K čemu potom to všechno snažení. Vnitřní bezpečnost musí vycházet z dobré morálky samotných uživatelů systému. Bývá také zvykem, že s předáním přístupových údajů do větších internetových aplikací musí uživatel podepsat prohlášení o bezpečnosti a ztotožnit se s základními bezpečnostními předpisy.

Prolomení hesla hrubou silou

Prolomení hesla může dokázat určitý software, který postupně zkouší velký počet různých kombinací slov a znaků, často vycházející z dostupných údajů o konkrétním uživateli. Útokem hrubou silou se dá zabránit pomocí omezení počtu zadání hesla pro vstup. Máte například možnost pokusit se přihlásit pětkrát po sobě a pokud ani na popáté nezadáte správné heslo, systém vám na určitý čas zabrání se přihlásit. Toto opatření odradí potencionální útočníky.

Prolomení hesla sociálním útokem

Prolomením hesla sociálním útokem nazýváme chybu lidského faktoru. To znamená, že heslo je buď zjištěno neoprávněnou osobou z důvodu nedbalosti odpovědného

³ Phising útok – nelegální způsob získávání citlivých údajů uživatelů pomocí takzvaného nahození.

Útočníci pošlou oběti email, který vypadá jako originální žádost systému o přihlášení a použijí skrytý odkaz na jejich stránky, kam oběť nevědomky vyplní přístupové údaje do originálního systému, které mohou útočníci zneužít.

pracovníka (zvolil si jako heslo jméno manželky, domácího mazlíčka, oblíbenou barvu apod.).

Hardwarové útoky

Hardwarové útoky jsou takové, kdy útočník má přímý přístup k vaší síti a podaří se mu odposlouchávat odesílané pakety na vaší síti. Toho je možné dosáhnout napojením do stejné podsítě pomocí ethernetového kabelu. Ještě snazší je to v případě připojení pomocí nezabezpečené bezdrátové sítě.

5 Vlastní implementace

Výše popsaný systém jsem naprogramoval pomocí technologií popsaných v kapitole 2. Pro chod systému je zapotřebí počítačový server s dobrou konektivitou internetu. Systém se zatím nachází v testovací fázi a je zprovozněn pouze na lokální stanici na mé vlastní počítačové síti.

5.1 Systémové požadavky

Pro konečné umístění systému bude zapotřebí zajistit počítačový server s minimálními hardwarovými specifikacemi pro běh podobných aplikací. Toto kritérium není nijak konkrétní, protože ve své podstatě systém může běžet i na „historických“ strojích, ale nevyvarujeme se tak zbytečných potíží s rychlostí přístupu (omezení rychlosti disků a procesoru) a možných havárií hardwaru.

Specifikuji tedy pouze jakési doporučené minimální hardwarové požadavky:

Procesor:

Intel Pentium 4 2.4GHz nebo AMD Athlon 64 +2800

Operační paměť:

Minimálně 1 GB

Diskový prostor:

Minimálně 5 GB (pro skripty a soubory)

Minimálně 20 GB (pro databázi)

Co se týká softwarových požadavků, nelze než doporučit operační systém založený na jádře UNIXu. Já osobně doporučuji instalaci operačního systému Ubuntu, které je v současné době ve verzi 9.04⁴. Pokud si stáhnete verzi pro server, získáte

⁴ Ubuntu 9.04 „Jaunty Jackalope“

automaticky sestavený systém pro internetové aplikace „LAMP⁵“, což je kombinace všech nutných prostředků, pro provoz internetového serveru.

5.2 Zavedení systému

Pro správný chod informačního systému a odladění všech potencionálních chyb je zapotřebí celý systém řádně otestovat. V případě, že systém projde testováním, přijde na řadu implementace do běžného provozu, které je naplánováno podle následujícího harmonogramu:

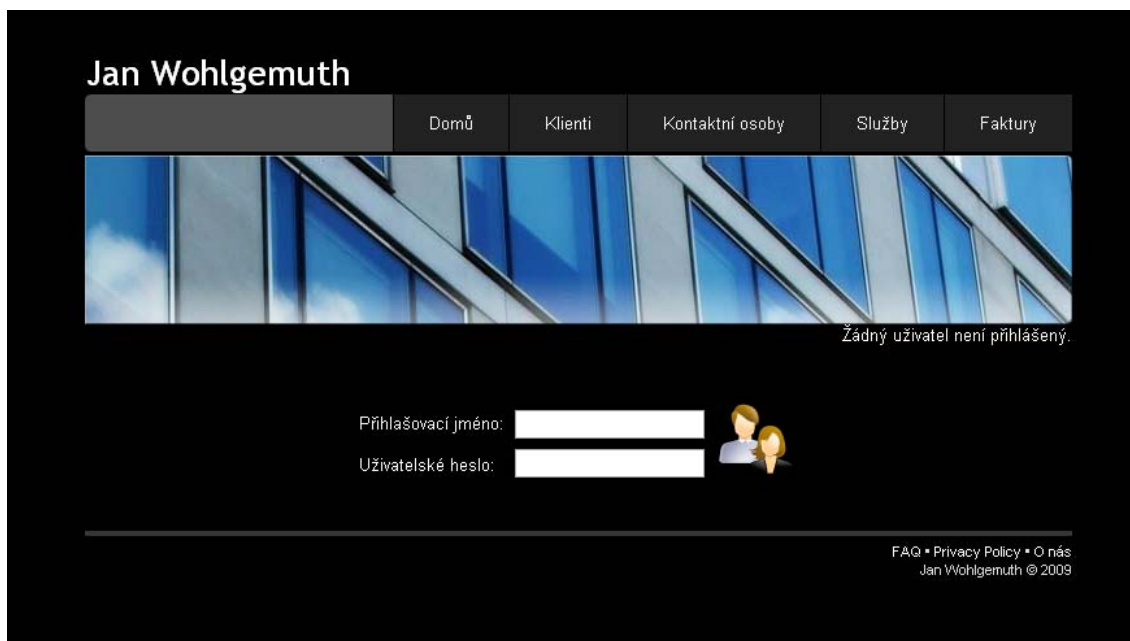
Květen - Červen 2009	Testování jednotlivých modulů
Červen 2009	Testování celku IS
Červenec - Srpen 2009	Implementace do firemních procesů, zaškolení pracovníků
31.srpna 2009	Ostrý start, fakturace přes IS, zavedení všech zakázek do systému

Tabulka č. 7 Harmonogram zavedení IS

5.3 Design systému

Pro ilustraci zde nabízím náhled do dvou oken systému. První z nich nabízí pohled do úvodní stránky pro přihlašování uživatelů. V horní liště se nachází navigační menu, které slouží pro rychlé přepínání mezi jednotlivými moduly systému.

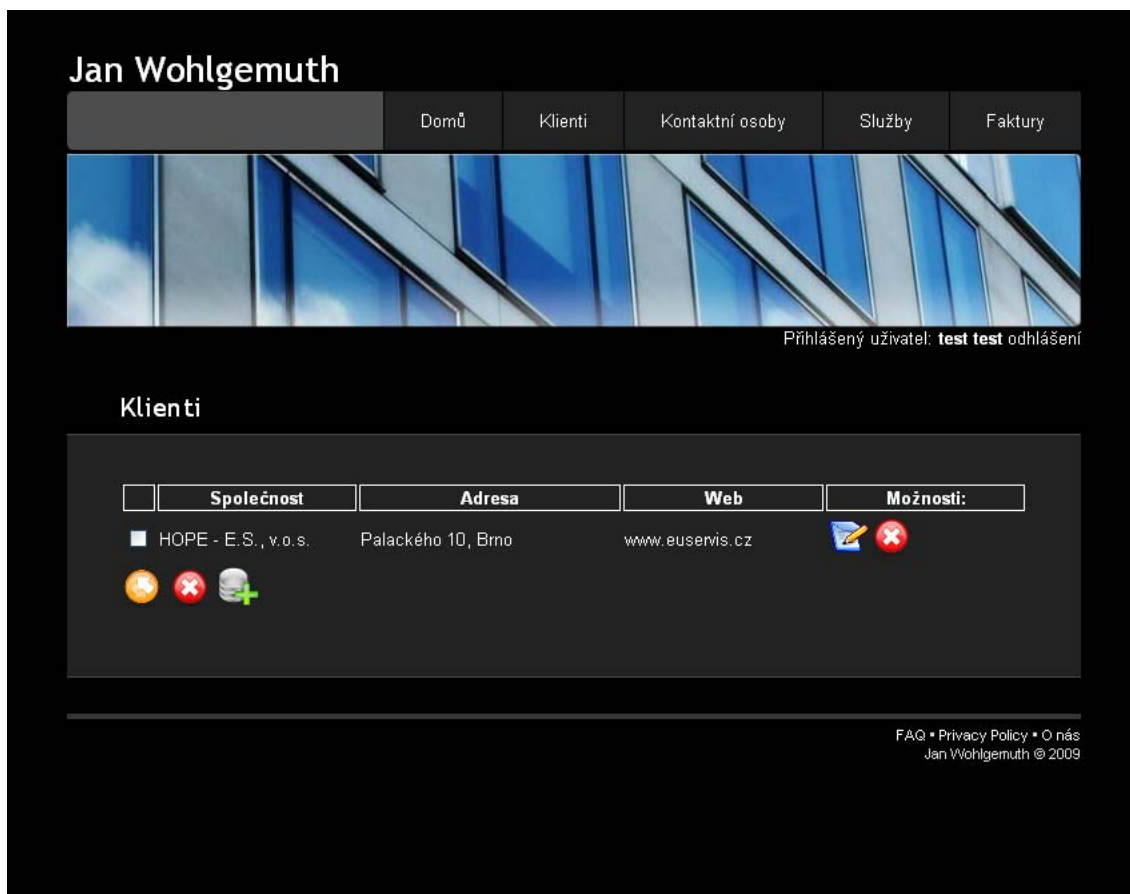
⁵ LAMP – Linux, Apache, MySQL, PHP



Obrázek č. 10 Úvodní obrazovka systému - přihlášení

Jako druhou obrazovku jsem vybral náhled do výpisu klientů. Zobrazeny jsou zde pouze nejdůležitější údaje o klientovi, které slouží pouze pro rychlé přesunutí k požadovaným údajům. Po rozkliknutí konkrétního klienta se můžeme dostat na všechny další záznamy spojené s tímto klientem (faktury, kontaktní osoby). Grafické ikony napomáhají v rychlých změnách v záznamech. U každého klienta můžeme zvolit buď výmaz klienta nebo editaci jeho údajů. Ikony pod tabulkou klientů slouží pro přidání nového klienta nebo pro hromadné operace – výmaz více klientů současně. Uvedené schéma tabulky je aplikováno i na další moduly systému, takže je jednoduché pro pochopení a uživatel se tak nemusí seznamovat s několika typy tabulek a formulářů.

Jako barvu pozadí jsem zvolil černou, která sice nepůsobí nijak optimisticky, ale lépe kontrastuje s bílým textem. Osvěžit vzhled mají barevné ikony.



Obrázek č. 11 Obrazovka systému - výpis klientů

5.4 Návrh možných rozšíření

V této kapitole bych se rád zamyslel nad budoucím vývojem informačního systému mé společnosti. Protože nepatřím mezi skeptické podnikatele a očekávám růst společnosti, nabízí se otázka zhodnocení možnosti rozšiřování systému.

Mezi první moduly, které bych rád do systému zabudoval, patří modul pro řízení nákladů na zaměstnance – „timesheety“. Rád bych, kdyby v budoucnosti každý uživatel systému mohl jednotlivým službám přiřazovat čas strávený jejich udržováním, zálohováním a jinou činností, která bude od klienta požadována. Tento modul by ohromně usnadnil vyúčtování mezi mnou a mými zaměstnanci, ale zároveň umožnil strukturování nákladů směrem ke klientovi.

Z dalších možných rozšíření by mohl přijít systém pro řízení ostatních nákladů – základ ekonomického software, kdy by se do systému zadávaly platby třetím stranám

a systém by tak mohl sloužit pro kompletní přehled nákladů a tržeb, což by umožnilo efektivnější finanční řízení.

Kromě nepřeborného množství rozšiřujících modulů nesmíme zapomínat ani na stávající funkční systém a umožnit jeho životaschopnost za pomoci okrašlovacích změn grafického návrh, ikon a další úpravy grafického a funkčního charakteru. Již v současném systému je například zabudována funkce na okamžité nalezení polohy adresy na serveru *www.mapy.cz*. Z dalších zlepšení je to přímé vytáčení telefonních čísel pomocí VoIP operátorů, které je možné, pokud se nachází na uživatelském počítači takto vybavený software. Tyto drobné změny způsobí větší oblíbenost systému, což je nepochybně podstatný aspekt v každé firemní struktuře.

Závěr

Prvotním cílem této práce bylo zanalyzovat chod mého podnikání pro potřeby následného návrhu a implementace informačního systému. Na základě analýzy vznikl konceptuální návrh systému, který byl následně implementován pomocí popsaných technologií. Výsledný systém se zatím nachází ve stádiu testování a neustále jsou u něj objevovány drobné chyby a nedostatky. Na odstranění těchto nedostatků budu pracovat i po odevzdání této práce.

Sekundárním cílem, který již nevyplývá ze zadání, ale uložil jsem si ho já osobně, bylo pokusit se zmapovat a sumarizovat nejjednodušší a nejstručnější technologie pro vytvoření podobného menšího informačního systému pro podporu podnikání malých a středních společností. V tomto směru jsem chtěl nastínit nejen postupné kroky (analýzu požadavků, návrh a následnou implementaci), ale také ty jednodušší nástroje, které je v těchto krocích nutné aplikovat.

Jestli jsem splnil u druhý „sekundární“ cíl své práce ukáže až čas, ale myslím si, že v tomto ohledu je dále co zlepšovat a sám bych se jistě přimlouval pro zpracování podrobnější instruktážní brožurky, která by napomáhala k větší informační gramotnosti drobných podnikatelů a jejich IT pomocníků.

Myslím si, že doba kdy se faktury psaly do starých formulářů na zažloutlém papíře a účetní sumarizace se počítaly na kalkulačce pomocí metody tužka/papír, jsou již dávno ty tam a každý, kdo se pokouší prorazit na trh s vlastním nápadem a chutí podnikat, bude nucen umět podobný systém minimálně používat. Ovšem mnohem lépe pro něj bude, když ho bude moci sám navrhnout či podílet se na jeho tvorbě. Vždyť takový systém bude sloužit právě jemu.

Literatura

- [1] LACKO, Branislav. *Význam projektového řízení pro automatizační praxi* [online]. 2005 [cit. 2009-05-08]. Dostupný z WWW: <http://www.odbornecasopisy.cz/index.php?id_document=30570>.
- [2] KRYL, Milan. TOP 10 chyb webových aplikací. Blog [online]. 2005 [cit. 2009-05-03]. Dostupný z WWW: <<http://kryl.info/clanek/237-top-10-chyb-webovych-aplikaci>>.
- [3] KRESLÍKOVÁ, Jitka. Řízení projektů informačních systémů. Opora IRP. FIT VUT v Brně, Brno 2006.
- [4] *Wikipedie : Message-Digest algorithm* [online]. 18. 5. 2009 [cit. 2009-05-05]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Message-Digest_algorithm>.

Seznam použitých obrázků

OBRÁZEK Č. 1 ŽIVOTNÍ CYKLUS PROJEKTU Z HLEDISKA PROJEKTOVÉHO ŘÍZENÍ [3]	13
OBRÁZEK Č. 2 ŽIVOTNÍ CYKLUS PROJEKTU Z HLEDISKA VÝVOJE PRODUKTU [3].....	14
OBRÁZEK Č. 3 PŘÍKLAD USE CASE DIAGRAMU - NÁVRH SYSTÉMU RESTAURACE	21
OBRÁZEK Č. 4 PŘÍKLAD ER DIAGRAMU - NÁVRH DATABÁZE SYSTÉMU RESTAURACE	22
OBRÁZEK Č. 5 VÝVOJ VERZÍ SPECIFIKACE BPMN	23
OBRÁZEK Č. 6 MODEL NAPOJENÍ MODULŮ SYSTÉMU	36
OBRÁZEK Č. 7 USE CASE MODEL SYSTÉMU - NEJVYŠŠÍ ÚROVEŇ	37
OBRÁZEK Č. 8 USE CASE MODEL - Use CASE MODULU KLIENT	38
OBRÁZEK Č. 9 ER DIAGRAM INFORMAČNÍHO SYSTÉMU	39
OBRÁZEK Č. 10 ÚVODNÍ OBRAZOVKA SYSTÉMU - PŘIHLÁŠENÍ	46
OBRÁZEK Č. 11 OBRAZOVKA SYSTÉMU - VÝPIS KLIENTŮ	47

Seznam použitých tabulek

TABULKA Č. 1 TABULKA NESPLŇUJÍCÍ 1.NF	16
TABULKA Č. 2 TABULKA BEZ PROBLÉMOVÝCH TELEFONNÍCH ČÍSEL OPATŘENÁ KLÍČEM V PODOBĚ ID	17
TABULKA Č. 3 TABULKA TELEFONNÍCH ČÍSEL SPOJENÉ POMOCÍ KLÍČE ID OSOBY	17
TABULKA Č. 4 TABULKA VÝROBKŮ NESPLŇUJÍCÍ PODMÍNKY 2.NF.....	18
TABULKA Č. 5 TABULKA VÝROBKŮ ODKAZUJÍCÍ NA VÝROBCE POUZE POMOCÍ KLÍČE ID VÝROBCE.....	18
TABULKA Č. 6 TABULKA VÝROBCŮ S KLÍČEM V PODOBĚ ID	18
TABULKA Č. 7 HARMONOGRAM ZAVEDENÍ IS	45

Seznam použitých zkratek

ASP - Active Server Pages je technologie programování webových stránek vyvinutá firmou Microsoft.

CSS - Cascading Style Sheets – tabulky kaskádové styly

GPL - General Public Licence – všeobecná veřejná licence

ICT - Information and Communication Technologies je souhrnným označením technické a programové infrastruktury podniku

IIS - Internetová a informační služby

MySQL - databázový systém vytvořený švédskou firmou MySQL AB. Je poskytován pod bezplatnou licencí GPL a pod licencí komerční.

PHP - je rekurzivní zkratkou PHP: Hypertext Preprocessor. Jedná se o kriptovací programovací jazyk, určený především pro programování webových stránek.

UML – Unified modeling language, viz. kapitola 2.2.2

Seznam příloh

Příloha 1. Datový slovník

Přílohy:

Příloha č. 1: datový slovník

Klienti

Sloupec	Typ	Nulový	Komentáře
ID_klient	int(11)	Ne	
ID_zamce	int(11)	Ne	FK - zamestnanci
ico	varchar(10)	Ne	
dic	varchar(12)	Ne	
jmeno	varchar(35)	Ne	
ulice	varchar(35)	Ne	
cislo_popisne	varchar(9)	Ne	
mesto	varchar(50)	Ne	
psc	varchar(5)	Ne	
telefon	varchar(14)	Ne	
email	varchar(35)	Ne	
web	varchar(40)	Ne	
cislo_uctu	varchar(20)	Ne	

Kontaktní osoby

Sloupec	Typ	Nulový	Komentáře
<u>ID_osoby</u>	int(11)	Ne	
ID_klienta	int(11)	Ne	FK - klienta
jmeno	varchar(35)	Ne	
prijmeni	varchar(50)	Ne	
titul	varchar(30)	Ne	
email	varchar(50)	Ne	
telefon	varchar(15)	Ne	
mobil	varchar(15)	Ne	
funkce	varchar(20)	Ne	

Komunikace

Sloupec	Typ	Nulový	Komentáře
<u>ID_komunikace</u>	int(11)	Ne	
ID_zamce	int(11)	Ne	FK - zamestnanci
ID_osoby	int(11)	Ne	FK - osoby
datetime	datetime	Ne	
zpusob	varchar(15)	Ne	
popis	text	Ne	

Služby

Sloupec	Typ	Nulový	Komentáře
<u>ID_sluzby</u>	int(11)	Ne	
ID_klienta	int(11)	Ne	FK - klienta
nazev_sluzby	varchar(35)	Ne	
popis_sluzby	text	Ne	

Položky služeb

Sloupec	Typ	Nulový	Komentáře
ID_polsluzby	int(11)	Ne	
ID_sluzby	int(11)	Ne	FK - sluzby
jmeno_polozky	varchar(50)	Ne	
popis_polozky	text	Ne	
prilozeny_soubor	varchar(50)	Ne	
info	text	Ne	
datetime_vzniku	datetime	Ne	
datetime_posledniuprav	datetime	Ne	

Faktury

Sloupec	Typ	Nulový	Komentáře
ID_faktury	int(11)	Ne	
ID_klienta	int(11)	Ne	FK - klienta
datum_vystaveni	date	Ne	
datum_plneni	date	Ne	
datum_splatnosti	date	Ne	
zaplaceno	tinyint(1)	Ne	
cislo_faktury	varchar(9)	Ne	

Položky faktury

Sloupec	Typ	Nulový	Komentáře
<u>ID_polfaktury</u>	int(11)	Ne	
ID_faktury	int(11)	Ne	FK - zamestnanci
nazev	varchar(80)	Ne	
cena_jednotka	varchar(8)	Ne	
jednotka	varchar(8)	Ne	
pocet_jednotek	varchar(8)	Ne	
dph	int(11)	Ne	

Zaměstnanci

Sloupec	Typ	Nulový	Komentáře
ID_zamce	int(11)	Ne	
login	varchar(15)	Ne	
password	varchar(32)	Ne	
jmeno	varchar(30)	Ne	
prijmeni	varchar(40)	Ne	
email	varchar(50)	Ne	
telefon	varchar(15)	Ne	
icq	varchar(15)	Ne	
skype	varchar(20)	Ne	
ulice	varchar(25)	Ne	
cislo_popisne	varchar(6)	Ne	
mesto	varchar(50)	Ne	
psc	varchar(5)	Ne	
prava	varchar(5)	Ne	
cislo_uctu	varchar(35)	Ne	
datum_narozeni	datetime	Ne	

Log

Sloupec	Typ	Nulový	Komentáře
ID_log	int(11)	Ne	
ID_zamce	int(11)	Ne	FK - zamestnanci
popis	varchar(50)	Ne	
datetime	datetime	Ne	
ip_adresa	varchar(15)	Ne	