

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

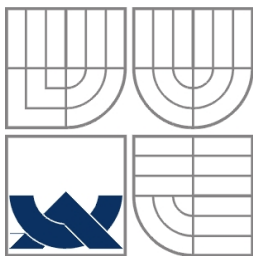
INFORMAČNÍ SYSTÉM MENŠÍ STAVEBNÍ
SPOLEČNOSTI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

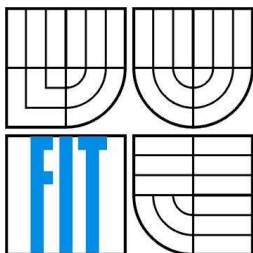
AUTOR PRÁCE
AUTHOR

TOMÁŠ HENKL

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM MENŠÍ STAVEBNÍ SPOLEČNOSTI

INFORMATION SYSTEM OF SMALL-SCALE BUILDING ENTERPRISE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tomáš Henkl

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vladimír Bartík, Ph.D.

BRNO 2007

Abstrakt

Tato práce se zabývá problematikou informačních systémů. Úkolem projektu je analyzovat, navrhnout a realizovat informační systém menší stavební společnosti, která se zabývá výstavbou a pronájmem velkokapacitních stanů a pódíí. Vše pomocí programovacího jazyku PHP, HTML, CSS, JavaScript a databázového systému MySQL.

Klíčová slova

Informační systém, databáze, PHP, HTML, CSS, MySQL

Abstract

This work is concerned with question of information systems. The task of the project is to analyze, design and realize information system of small-scale building enterprise, which deals with construction and hire of stages and high-density tents. Everything by means of programming languages PHP, HTML, CSS, JavaScript and database system MySQL.

Keywords

Information system, database, PHP, HTML, CSS, MySQL

Citace

Tomáš Henkl: Informační systém menší stavební společnosti, bakalářská práce, Brno, FIT VUT v Brně, 2007

Informační systém menší stavební společnosti

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením ing. Vladimíra Bartíka, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Henkl
Datum

© Tomáš Henkl, 2007.

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

Obsah

Obsah	1
Úvod	3
1 Informační systémy	4
1.1 Informační systém	4
1.2 Databáze	5
2 Analýza požadavků a návrh řešení	6
2.1 Požadavky na aplikaci	6
2.1.1 Obecné požadavky	6
2.1.2 Přístup k aplikaci	7
2.1.3 Data v informačním systému	7
2.1.4 Vyhledávání a archivace	9
2.1.5 Práce s daty	10
2.1.6 Statistiky	10
2.2 Návrh řešení	11
2.2.1 Jazyk UML	11
2.2.2 Diagram užití	11
2.2.3 Zjednodušený diagram tříd (ER-diagram)	12
2.2.4 Převod modelu na tabulky databáze	13
3 Implementace	15
3.1 Implementační prostředky	15
3.1.1 MySQL	15
3.1.2 PHP	16
3.1.3 HTML	16
3.1.4 JavaScript	17
3.1.5 XML	18
3.2 Spuštění aplikace	18
3.3 Uživatelské prostředí	18
3.3.1 Výběr uživatele	18
3.3.2 Hlavní okno systému	19
3.3.3 Správa dat	20
3.3.4 Vyhledávání	21
3.3.5 Archivace	21
3.3.6 Statistiky	22
3.3.7 Tisk	22

3.4	Problémy při implementaci	22
4	Možná rozšíření	24
	Závěr	25
	Literatura	26
	Seznam příloh	27

Úvod

Informační systémy, i když si to mnozí z nás neuvědomují, jsou čím dál tím více součástí dnešního života každého z nás. Setkáváme se s nimi snad na všech větších státních úřadech, těžko by jsme si už dnes bez nich dokázali představit život na většině středních a snad všech vysokých školách a veliké i středně velké firmy by se bez informačních systémů asi těžko obešli bez újmy na efektivitě. Postupem času se stanou součástí i menších firem. Důvod je jednoduchý. Informační systémy, pokud jsou dobře navrženy a zpracovány, šetří nejenom čas, ale v nezanedbatelné míře i peníze. Právě takovýto informační systém menší společnosti, která se zabývá stavbou a pronájmem pódíí a velkokapacitních stanů je obsahem této práce.

V první části se budeme zabývat objasněním pojmu informační systém, dále se tato práce čtenáři pokusí přiblížit pojem databáze, která je nezbytnou součástí takovýchto projektů, a nezapomene definovat další důležité pojmy.

V další části bude rozebrána problematika konkrétního informačního systému, který je předmětem tohoto projektu. Podíváme se na problém přístupu k projektu. Seznámíme se s jakými daty a položkami bude databáze plněna a budou popsány operace se záznamy v databázi.

Ve třetí kapitole bude nastíněn podrobnější popis jednotlivých implementačních nástrojů, které byli použity při tvorbě programové části projektu. Bude zde také popsána implementace informačního systému menší stavební společnosti. Dále v textu jsou popsány programované součásti aplikace.

Čtvrtá kapitola se zabývá jednotlivými možnými vylepšeními některých prvků programu a také možnosti implementace nových prvků.

1 Informační systémy

[3] Obecný systém jako takový, je cíleně vytvořený soubor komponent, mezi kterými existují určité vztahy, a které splňují nějaký cíl. Skládá se z atributů (veličiny jež charakterizují určitý prvek systému), událostí (změna atributu nebo změna konfigurace systému - například komponenty) a časových množin (hodnoty vztažené k určitému okamžiku). Systémy obecně dělíme na tvrdé a měkké. Tvrdé systémy jsou spojovány s jedním specifickým (strukturovaným) problémem (většinou technické systémy), naopak v měkkých systémech vystupuje celá řada faktorů, jsou obecnější (člověk je aktivním prvkem systému=individualita - problém protože každý má jiné znalosti a jinak uvažuje). Existují další druhy dělení:

- uzavřené × otevřené (podle toho, zda nastává interakce s okolím),
- deterministické × stochastické (tzn. jednoznačné nebo statistické chování),
- statické × dynamické (tzn. lineární nebo diferenciální -systém si pamatuje vnitřní stav),
- spojité × diskrétní (podle časových událostí, případně ex. také kombinované).

V systémech může nastat zpětná vazba, kdy výstupní veličina opětovně ovlivňuje vstupní veličinu, a tedy i samotný systém. Každý systém má tedy tendence být nestabilní, což nemusí nutně znamenat nevýhodu nebo poruchu, u mnoha systémů je tento efekt dokonce velice žádaný.

1.1 Informační systém

Je podle [3] speciální případ systému pro sběr, udržování, zpracování a poskytování informací a dat. Příkladem informačního systému může být telefonní seznam nebo kniha došlé pošty. Takovýto systém nemusí být nutně automatizovaný pomocí počítačů, může se jednat klidně o papírovou podobu systému. Jde tedy přesně o konceptuální, otevřený systém s uzavřeným cyklem, kde jeho základním úkolem je zpracování dat a zabezpečení komunikace. Koncový uživatel používá informační systém dvěma způsoby. Uloží data (informace) a nalezne data (informace).

- **Data** jsou jakékoli poznatky či fakta, používané pro popis nějakého jevu nebo vlastnosti pozorovaného objektu. Jsou vyjádření skutečností formálním způsobem tak, aby je bylo možno přenášet nebo zpracovat. Lze je získávat měřením nebo pozorováním. Dělí se na data spojitá a data atributivní. Data spojitá se přímo vztahují k nějaké spojité stupnici, zatímco data atributivní nikoliv.
- **Informace** jsou výsledek vyhodnocování smyslových vjemů, zpracování nebo organizace dat a proto mohou být ovlivněny uživatelem. Je možno je chápat jako data s nějakým přidaným významem (data + význam). Podle filozofů to jsou sdělení, které odstraňuje nejistotu nebo nevědomost.

1.2 Databáze

Databáze je určitá uspořádaná množina dat (informací) uložená na paměťovém médiu. Předchůdcem databáze by si každý mohl představit kartotéku. Umožňovaly uspořádávání dat podle různých kritérií a zatřídování nových položek. Veškeré operace s nimi prováděl přímo člověk. Správa takových kartoték byla v mnohém podobná správě dnešních databází.

V širším smyslu jsou součástí databáze i softwarové prostředky. Tyto prostředky umožňují definice dat, vytváření slovníku dat, manipulace s daty, zajištění bezpečnosti a integrity dat, zotavení po chybách, souběžný přístup (transakční zpracování) a zajištění co nejvyšší výkonnosti. V české odborné literatuře nazývají systém řízení báze dat (SRBD). Běžně se označením databáze – v závislosti na kontextu – myslí jak uložená data, tak i software (SRBD).

Z hlediska způsobu ukládání dat a vazeb mezi nimi můžeme rozdělit databáze do základních typů:

- Hierarchické databáze – 2.polovina 60. let 20.století a dále
- Síťová databáze - konec 60. let 20.století a dále
- Relační databáze – 1970 teorie, 1975 Systém R (IBM) až doposud
- Objektové databáze –v dnešní době
- Objektově relační databáze – pomalý nástup již dnes

Data v databázi jsou ve své podstatě známá fakta o věcech které nás zajímají a chceme si je uchovat v elektronické podobě s přibližně stejnou strukturou. Data v databázi jsou charakteristická většinou velkým množstvím podobných dat, perzistentním chováním (přetrvávání dat), sdílením s jinými uživateli, bezpečností přístupu a omezení možnosti manipulace.

2 Analýza požadavků a návrh řešení

2.1 Požadavky na aplikaci

Požadavky na aplikaci by měly objasnit, co bude aplikace vykonávat, jaké bude mít hlavní funkce. Je to jedna s fází návrhu informačního systému. Součástí této fáze je i konzultace s cílovým uživatelem, z důvodu ujasnění si celé koncepce.

2.1.1 Obecné požadavky

Hlavním důvodem vzniku tohoto informačního systému je zjednodušení, zefektivnění a zpřehlednění chodu menší společnosti, která se pomalu zavádí na český trh a její růst bez elektronického informačního systému zmenšuje její naději na úspěch.

Cílovým uživatelem bude malá firma, která se zabývá stavbou a pronájmem podíí a jejich zastřešení, velkokapacitních stanů, tribun a posezení na různé zábavní, sportovní a kulturní akce. Působnost je na celém území České republiky, převážně však na severní a jižní Moravě. Firma má dva (může mít v budoucnu i více) hlavní stálé zaměstnance, kteří celou firmu řídí a sjednávají zakázky, a několik (5-15) nasmlouvaných pracovníků na stavbu. Zákazníci budou moci zadávat zakázky pomocí internetu, pracovníci se budou moci registrovat na práce vypsané z jednotlivých zakázek odkudkoliv, kde právě budou a zaměstnanci nebudou muset trávit celý den v kanceláři aby měli všechny potřebné údaje nadosah. Z toho důvodu by se mělo jednat o víceuživatelskou internetovou aplikaci dostupnou z různých míst republiky i světa.

S aplikací budou pracovat jak lidé, kteří jsou znalí výpočetní techniky, tak i absolutní laici. Proto je nezbytně nutné aby veškeré ovládací prvky i struktura celého informačního systému byly jednoduché a celá práce s ním otázkou pouhé intuice.

Aplikace musí být správně strukturovaná a proto je důležité dodržet následující pravidla:

- Zajistit různé úrovně práv přístupů
- Ke každému zákazníkovi připojit správné objednávky
- Pracovníkům přidat práce které si zaregistrují
- Objednávkám přidat správné zboží, které bylo objednáno a vystavit správnou fakturu pro platbu
- Po ukončení prací přidat pracovníkům hodiny podle délky práce a vrátit půjčené zboží do skladu
- Kontrolovat volné a půjčené zboží
- Editovat zákazníky, pracovníky, zaměstnance, zakázky i zboží

Pokud do systému nějaká data vložíme, měli bychom je najít v nezměněné podobě kdykoliv je budeme potřebovat při kterémkoliv použití aplikace. Smazána mohou být pouze, pokud tak někdo učiní.

Pokud se budou některé položky přidávat nebo budeme měnit stávající položky, měli bychom si dávat pozor na postup, tak abychom se nedopustili porušení struktury dat. V případě že budeme mazat jednotlivé položky, musíme s nimi odstranit i data která jsou na ně vázána a která by tímto ztratila opodstatnění uchovávat je.

2.1.2 Přístup k aplikaci

Jak již bylo napsáno výše, jedná se o klasický informační systém pro předem nespecifikovaný počet uživatelů. Z tohoto důvodu je potřeba rozdělit uživatele do několika různých skupin podle možností přístupu do systému a práce s daty. Každá skupina by měla mít možnost přístupu jen do té části informačního systému, do které má oprávněný vstup (př. Zákazníci nesmí mít možnost měnit údaje o pracovnících a to jakýmkoliv způsobem). První skupina budou stálí zaměstnanci. Ti budou mít administrátorská práva, přístup skoro ke všem nastavením, editacím a odstraňování informací ze systému. Druhá bude skupina zákazníků, kteří budou moci pouze editovat jejich osobní data a informace týkající se jejich zakázek. Třetí skupinou budou pracovníci, jejich přístup bude jen k jejich osobním datům a k registraci na vypsane práce.

2.1.3 Data v informačním systému

Mezi hlavní a nejdůležitější data v systému budou patřit navzájem propojené prvky (tabulky) databáze: zákazník, akce, stavba, zaměstnanec, práce. Každý tento prvek by měl obsahovat pro něj specifická data, ale na druhou stranu se tyto prvky i navzájem ovlivňují.

Zákazník

Zákazník bude v systému chápán jako firma, která má nějakou adresu kde společnost sídlí. Jméno, telefon a e-mail konkrétní osoby, která se o objednávku stará ze strany zákazníka (jednatele). V případě soukromé osoby, se jedná o její adresu a jednatelem je osoba samotná. Každý ze zákazníků musí mít jedinečné přihlašovací jméno (login) a z důvodu bezpečnosti i heslo. Bude zde i položka pro výpočet slevy, například za objem uzavřených zakázek. Detailní struktura zákazníka:

- Login a heslo
- Jméno firmy nebo zákazníka
- Město, ulice a směrovací číslo sídla(bydliště)
- Jméno, telefon a e-mail jednatele
- Počet zakázek

Zákazníci se mohou do systému přidávat sami pomocí registračního formuláře, kde tyto údaje vyplní. V systému je možno mít i zákazníky bez jakýchkoliv zakázek, protože zaregistrování nebude přímo podmíněno objednávkou.

Akce

Další důležitou tabulkou, která bude v databázi uložena, bude akce. Zadáním přesného místa, data a času začátku, data a času konce, zákazník specifikuje konkrétní akci. Pokud bude chtít zákazník cokoli speciálního o akci sdělit, může k tomu využít položku poznámka. Akce bude jednoznačně určena svým identifikačním číslem. Každý zákazník může mít specifikováno několik akcí třeba i na stejnou dobu a na stejné místo. Pokud bude akce zadána podle přání zákazníka, potvrdí ji a akce bude postoupena k dalšímu zpracování, které má na starost administrátor. Vyřízena je položka která, má signalizovat, že akce byla již vyřízena .

Detailní data akce:

- Id akce
- Zákazník který si akci objednal
- Místo konání akce
- Datum a čas začátku a konce akce
- Poznámka
- Potvrzeno a vyřízeno

Akce je první částí pro vytvoření celé objednávky. Zákazník si poté vybere stavby které si chce na danou akci pronajmout a nechat postavit. Položka bude v systému uložena i pokud nebude obsahovat akce, ale nepůjde potvrdit, nemá smysl aby se akcí, kde se nemá nic stavět, zabýval někdo ze zaměstnanců.

Stavba

Je položka která je v systému velice důležitá. Je v ní zaznamenáno, kterou konkrétní stavbu si zákazník chce pronajmout na kterou akci a podle toho se potom vytváří práce pro pracovníky. Zákazník si zde vybere typ stavby a její rozměry (potažmo počet). Zákazník se zde dozví i ceny a dobu případné stavby. Detail:

- Id stavby
- Akce na které se bude stavět
- Typ stavby
- Šířka a délka
- Cena a doba stavby

Po přidání budou jednotlivé stavby podle typu vloženy do tabulky půjčené stavby a jejich případnému objednání na stejnou dobu systém nedovolí.

Práce

Práci do systému vloží zaměstnanec (administrátor). Vychází ze zákazníkem potvrzené akce. Z jedné akce se vytváří dvě práce, postavení stavby a po skončení akce, její zbourání. Důležitou informací je vedoucí stavby. Ten bude vybrán z řad pracovníků, který bude moci práci, na kterou bude přidělen, upravovat (jinak to bude moci pouze administrátor), počet lidí kteří se mohou na práci přihlásit a doba trvání práce. Detail práce:

- Id práce a id akce která se zpracovává
- Typ práce (stavba nebo bourání)
- Datum a čas začátku
- Vedoucí, počet lidí a doba trvání práce
- Dopravce materiálu

V případě smazání práce se smaže i její protějšek (pokud se smaže stavba, musí se smazat i bourání a naopak). Dopravce nemusí být zadán, to informační systém vyhodnotí jako vlastní dopravu materiálu.

Zaměstnanec

Zaměstnance (chápe se jako pracovník i stálý zaměstnanec) do systému bude moci přidat jen zaměstnanec s právy administrátor, protože není možné, aby se jako zaměstnanec mohl přihlásit každý. Identifikace bude probíhat podle loginu, který bude mít každý zaměstnanec jiný, a autentizace bude probíhat pomocí hesla. Nezbytné jsou údaje o jménu a bydlišti zaměstnance. Důležitým prvkem jsou práva jednotlivých zaměstnanců, která je rozdělí na pracovníky a administrátory. Dalším parametrem jak jsou zaměstnanci děleni jsou jejich schopnosti (ne všichni pracovníci mohou být vedoucí práce). Detail :

- Id, login a heslo
- Jméno
- Město, ulice, směrovací číslo, telefon, e-mail
- Práva
- Hodinová mzda a počet odpracovaných hodin
- Schopnosti

Vzhledem k tomu, že v tomto informačním systému musí být alespoň jeden zaměstnanec s právy administrátora. Nelze jej, pokud je poslední, ze systému odstranit.

2.1.4 Vyhledávání a archivace

Vzhledem k postupnému zaplňování databáze informačního systému bude potřeba v důležitých datech a v takových datech, které budou v systému ve větším množství, umět vyhledávat přesně ty údaje, které nás zajímají (i za podmínky, že třeba neznáme přesně jejich název). Ne zobrazovat

zbytečné záznamy, pokud si to nebudeme vysloveně žádat. Mezi takové údaje budou bezesporu patřit údaje o zákaznících a jejich zakázkách, zaměstnancích, půjčených stavbách, fakturách,

Neméně důležitou funkcí systému by měla být archivace údajů mimo informační systém (např. do externího souboru mimo databázi), které nejsou již tolik důležité, ať už pro svoji zastaralost, nebo které nebudeme chtít mít aktivně v systému z jiného důvodu.

2.1.5 Práce s daty

Čistý informační systém bude obsahovat pouze administrátorský účet a sortiment zboží, který firma používá ke svému podnikání. Systém se bude postupně plnit daty, a proto je nezbytně nutné mít možnost tyto data upravovat a mazat. Všechny údaje musí mít možnost vkládat, editovat a odstraňovat uživatel s administrátorskými právy. Zákazník může editovat pouze údaje, které do systému sám zadal a každý uživatel může měnit svoje osobní data.

Pokud bude chtít někdo vložit nějaké nové údaje do systému, musí se tak díť pomocí co nejjednodušších operací. Měl by se dozvědět, které informace jsou nezbytné pro správné zadání nových dat a které jsou na vůli zadávajícího, jestli je chce zadávat. Kontrola správnosti zadaných údajů by měla být automatickou součástí, zvláště pokud půjde o zadávání od zákazníků a zadavatel by měl být upozorněn na chybně zadané údaje. Při zadávání práce, by měl systém nabídnout všechny pracovníky, kteří mají v systému nastaveno že mohou být vedoucí a taky všichni volní dopravci.

Při editaci položek by měla platit stejná pravidla jako při vkládání nových údajů: vyplněné povinné údaje, kontrola správnosti dat, jednoduchost a přehlednost. Údaje, které již v systému jsou uloženy by se měly při editaci zobrazovat v příslušných kolonkách, aby se nemusely vypisovat znovu údaje, které nejsou předmětem změny.

Při odstraňování položek na kterých jsou závislá jiná data je nezbytné dbát na odstranění i souvisejících údajů (př. když odstraním akci, tak musím odstranit i všechny stavby které jsou na tuto akci objednány z objednaných staveb), aby se v systému nehromadily zbytečné údaje, které k ničemu neslouží a jen zdržují vyhledávání potřebných údajů.

2.1.6 Statistiky

Statistiky jsou nezbytnou součástí každého rozhodování kam bude firma investovat, jestli koupit další pódia nebo nové stany. Statistiky by měly být přehledně zpracovány, nejlépe pomocí vhodného druhu grafu.

V tomto informačním systému se bude jednat o statistiky jednotlivých druhů staveb a jejich využití v závislosti na čase. Další možnou statistikou je celkový počet a rozsah zakázek za jednotlivá období, která může hodně napovědět kolik pracovníků si má firma na jaké období nasmlouvat.

2.2 Návrh řešení

Bude zde popsán modelovací jazyk, který jsem použil při modelování (UML) a některá zásadní pravidla pro převod modelu na databázi.

2.2.1 Jazyk UML

Návrh aplikace může teoreticky vzniknout pouze za použití tužky, papíru a mozku. Model ale bude používat jen vámi známou syntaxi a sémantiku, ve které se patrně nikdo jiný nevyzná. Je proto lepší využít některý z analytických jazyků.

Unified Modeling Language (UML) má již ve svém názvu slovo unifikovaný, protože jedním z jeho cílů je sjednocení používaných výrazových prostředků. UML je jazykem s bohatou sémantikou a syntaxí, který usnadňuje návrh a vizualizaci různých typů aplikací. Navíc tvůrci jazyka UML nebyli natolik naivní ani sebevědomí, aby věřili, že standardními elementy jazyka UML pokryli kompletně veškeré požadavky návrhářů, a přímo do jazyka zabudovali mechanismy rozšíření, které usnadňují řízenou deklaraci a definici nových elementů jazyka (citace z[4]).

Historie jazyka není nijak dlouhá, ale nevznikl jen tak ve vzduchoprázdnu. Předcházela mu snaha mnoha znalců o vytvoření kvalitní metody, která by umožňovala kvalitní modelování. V 1996 se pánové Jacobson, Booch a Rumbaugh sešli ve firmě Rational Corp, vybrali nejlepší vlastnosti z existujících konkurenčních jazyků a spojili je do jazyka UML. V roce 1997 byl jazyk přijat jako průmyslový standard a postupem času se stal nejrozšířenějším analytickým jazykem.

Model v UML se skládá z různých diagramů, jež představují pohledy na různé části sémantického základu navrhované aplikace. Sémantickým základem je souhrn specifikací aplikace, který vymezuje působnost, v němž se může návrh pohybovat. Diagram ve vizuální formě zobrazuje právě jednu konkrétní informaci o aplikaci. Žádný dvourozměrný diagram nemůže zachytit komplexní aplikaci v celku, ale soustředí se vždy právě na jeden důležitý poznatek. Velice známé a hojně využívané jsou diagramy užití (use-case) a diagramy tříd.

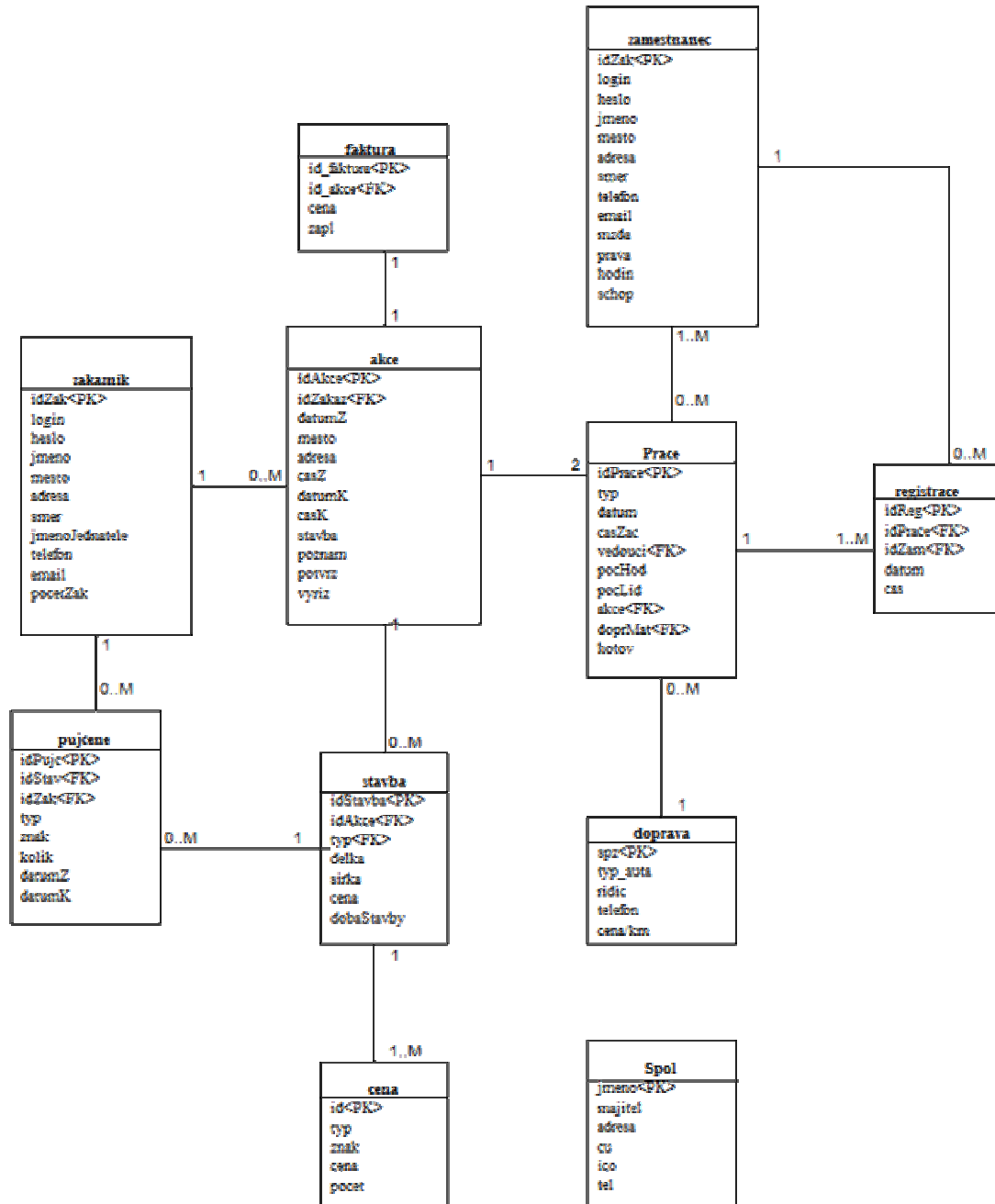
2.2.2 Diagram užití

Tato část UML je nejméně rozpracovanou částí celé metodiky. Use-case diagram vyjadřuje vztahy mezi účastníky vně systému a účastníky uvnitř systému. Jsou zde zachyceny hlavní funkční požadavky na systém.

V této fázi analýzy se ještě nezabýváme technologickými aspekty řešení a používáme pouze pojmy přirozeného jazyka a termíny z problémové oblasti, abychom co nejvěrněji a pro zákazníka co nejsrozumitelněji sestavili funkční obraz celého systému.

2.2.3 Zjednodušený diagram tříd (ER-diagram)

Použití E-R (entity-relationship) diagramu je snahou představit jednotlivé entity systému a vztahy mezi nimi. Na obrázku 2.1 je znázorněn ER-diagram informačního systému menší stavební společnosti.



Obrázek 2.1: ER diagram projektu

2.2.4 Převod modelu na tabulky databáze

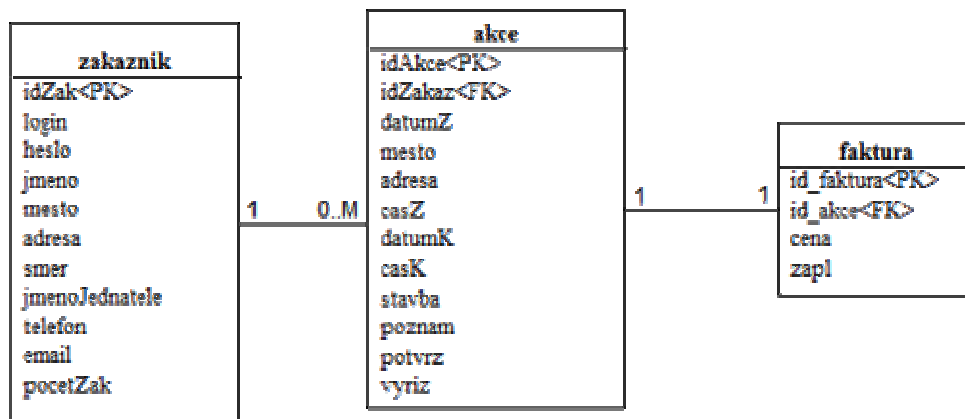
Pokud chceme převést model na tabulky databáze, je nezbytné dodržovat pravidla, která byla pro převod vytvořena. Normalizace je pojem, který značí proces převodu modelu na tabulky databáze a její součástí je i dodržování těchto pravidel. Pomocí normalizace můžeme zjednodušit databázovou strukturu do stavu, který bude vhodný pro užití relační databáze, která má být v našem případě, výsledkem tohoto procesu.

Při převodu ER-diagramu vznikají mezi entitami vztahy s různou kardinalitou přechodů. Mezi nejdůležitější patří vztahy s kardinalitou 1:1, 1:M a M:M. Jednoduše to znamená kolik, prvků jedné entity může mít entita na ní závislá.

Příklad převodu vztahu s kardinalitou 1:M a 1:1

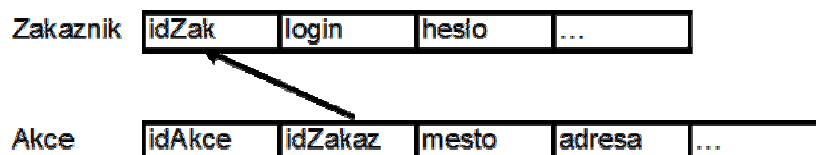
Pokud je mezi entitami vztah 1:M, znamená to, že jedna entita (X) může mít více entit na ni závislých (Y), ale pokud se na tento vztah podíváme z pohledu závislých (Y) entit, může mít každá pouze jednu závislou entitu(X).

V případě vztahu 1:1 může mít každá entita jen jednu entitu na ni závislou a to i z pohledu závislé entity.



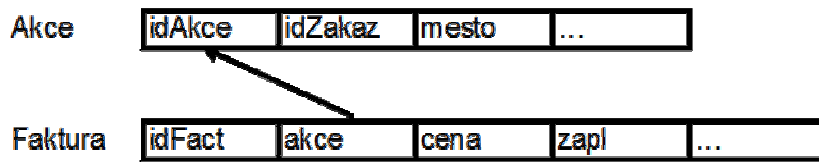
Obrázek 2.2: Ukázka vztahu 1:M a 1:1

Při převodu vztahu 1:M musíme vložit nový sloupec do tabulky u které je naznačena kardinalita N, kam vložíme cizí klíč, který se bude okazovat na primární klíč druhé tabulky.



Obrázek 2.3 Převod vztahu 1:M do podoby tabulky

Pokud převádíme vztah 1:1 musíme vložit nový sloupec do jedné z tabulek. Tento sloupec bude cizím klíčem odkazující se na primární klíč v druhé tabulce.



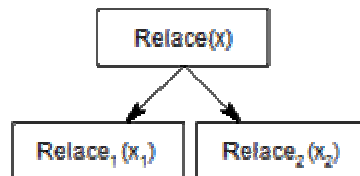
Obrázek 2.4 Převod vztahu 1:1 do podoby tabulky

Mezi hlavní problémy chybného návrhu můžeme zařadit opakování informace (redundance), nemožnost reprezentovat určitou informaci a složité kontroly integritních omezení. Těmto nedostatkům pomáhá předejít normalizace.

Při normalizaci se aplikuje takzvaná transformace na normální formy (NF). Těchto transformací je celkem šest, ale poslední dvě (čtvrtá a pátá se používají jen ojedinele). Označují se číselně a se vzrůstajícím číslem roste i jejich důkladnost (kvalita). Každá další NF musí splňovat požadavky normální formy předešlé.

1. První normální forma určuje základní požadavek relačního modelu dat, při pohledu na složitost dat. Jedná se o atomičnost hodnot jednoduchých atributů.
2. Druhá normální forma znamená, že je v první normální formě a každý její neklíčový atribut je plně závislý na celém primárním klíči a ne pouze na jeho jednotlivých částech.
3. Třetí normální forma značí že je splněna 2.NF a zároveň nejsou na sobě závislé neklíčové atributy.
4. Boyce-Coddova normální forma předpokládá existenci netriviální závislosti X,Y kde X je superklíč. Superklíčem je každá nadmnožina kandidátního klíče.
5. Čtvrtá normální forma určuje vlastnosti z hlediska vícehodnotových atributů.
6. Pátá normální forma definuje vlastnosti z pohledu závislosti na spojení.

Při převodech nižších forem na vyšší normální formy se nevyhneme dekompozici. Jedná se o rozpad schématu relace na několik schémat, které nezapříčiní vznik nových atributů, avšak žádný atribut neubude. Při dekompozici se musíme zbavit redundance (opakování informace), musí být zachovány závislosti a nesmí dojít k žádné ztrátě při zpětném spojení. Typickým příkladem je rozpad původního schématu na dvě nová schémata.



Obrázek 2.5 Dekompozice kde $x = x_1 \cup x_2$

3 Implementace

Tato kapitola ukáže jaké implementační prostředky byly použity a popíše způsob implementace jednotlivých částí systému, jeho instalace a zprovoznění.

3.1 Implementační prostředky

Pro implementaci informačního systému jsem zvolil:

- Databázový systém MySQL
- Skriptovací programovací jazyk PHP
- Značkovací jazyk HTML
 - CSS - Cascading Style Sheets
- Programovací jazyk JavaScript
- Značkovací jazyk XML

3.1.1 MySQL

Vzhledem k postupnému přibývání většího množství dat, bylo potřeba ukládat tyto data do databáze. Pro potřeby tohoto systému jsem zvolil opensource databázový systém MySQL. Jedná se o systém vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael Widenius a David Axmark. MySQL je multiplatformní databáze.

Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními. MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, triggerly, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních verzích.

Velmi oblíbená a často používaná je kombinace MySQL, PHP a Apache jako základní software webového serveru. Má vysoký podíl na v současné době používaných databázích. Hlavními důvody jsou:

- Snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy)
- Jedná se o volně šiřitelný software
- Veliká rychlost vykonávání SQL dotazu
- Neomezený počet tabulek (může být omezen operačním systémem)

Pro tyto vlastnosti je využíván ve velké většině neziskových organizací, škol, univerzit a poskytovatelů internetu.

3.1.2 PHP

Jedná se o skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML či WML, což je velmi výhodné pro tvorbu internetových (webových) aplikací. PHP lze ovšem také použít k tvorbě konzolových a desktopových aplikací.

PHP skripty jsou prováděny na straně serveru, k uživateli je přenášěn až výsledek jejich činnosti. Syntaxe jazyka kombinuje hned několik programovacích jazyků (C, Java, Pascal). PHP je platformě nezávislý, tzn. že skripty fungují bez úprav na mnoha různých operačních systémech. Obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k většině databázových serverů (mj. MySQL), podporu spousty internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP, ...). Patří mezi jazyky, kde není nutné předem definovat typ proměnných, navíc jakákoli proměnná může kdykoli změnit svůj typ. S verzí PHP 5 se výrazně zlepšil přístup k objektově orientovanému programování, který je podobný Javě.

Od roku 1994 je PHP jedním z nejpoužívanějších způsobů tvorby dynamicky generovaných WWW stránek. Jeho tvůrce (Rasmus Lerdorf) jej vytvořil pro svou osobní potřebu. Sada skriptů byla vydána ještě v téže roce pod názvem Personal Home Page Tools, zkráceně PHP. Později se systém PHP spojil s programem Form Interpreter stejného autora. Tak vzniklo PHP/FI 2.0. Zeev Suraski a Andi Gutmans v roce 1997 přepsali parser a zformovali tak základ PHP3. Současně byl název změněn na dnešní podobu PHP hypertext procesor.

PHP3 vyšlo v roce 1998, bylo rychlejší a obsahovalo více funkcí. Také běželo i pod operačním systémem Windows. V roce 2000 vychází PHP verze 4, o čtyři roky později pak verze 5 s vylepšeným objektovým přístupem, podobným jazyku Java. Zatím poslední je verze 5.2.1.

3.1.3 HTML

V roce 1989 spolupracovali Tim Berners-Lee a Robert Caillau na propojeném informačním systému pro CERN, výzkumné centrum fyziky poblíž Ženevy ve Švýcarsku. V té době se pro tvorbu dokumentů obvykle používal TeX, Postscript a také SGML. Berners-Lee si uvědomoval, že potřebují něco jednoduššího a v roce 1990 byl tedy navržen jazyk HTML a protokol pro jeho přenos v síti - HTTP (HyperText Transfer Protocol - přenosový protokol hypertextu). V pozdější době došlo k velkému rozvoji webu a bylo nutné pro HTML definovat standardy. První verze standardu byla 0.9 vydaná v roce 1991 a poslední verze 4.01 vyšla na konci roku 1999 [3].

Je charakterizován množinou značek, které jsou většinou párové (také nazývané „tagy“, rozlišujeme počáteční a koncové značky, koncová značka má před názvem značky znak lomítka), a jejich atributů. Mezi značky se uzavírají části textu dokumentu a tím se určuje význam (sémantika) obsaženého textu. Názvy jednotlivých značek se uzavírají mezi úhlové závorky ("<" a ">"). Část dokumentu uzavřená mezi značkami tvoří tzv. element (prvek) dokumentu.

Součástí obsahu elementu mohou být další vnořené elementy. Atributy jsou doplňující informace, které upřesňují vlastnosti elementu.

CSS - Cascading Style Sheets

Hlavním smyslem CSS je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů, se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují jen obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí.

Mezi výhody použití CSS patří:

- Rozsáhlejší možnosti
- Konzistentní styl
- Dynamická práce se styly
- Kratší doba načítání stránky

Výhodou CSS oproti starému formátování v HTML je, že veškerý design a formátování se načítá z jednoho souboru .css, který je většinou společný pro celou aplikaci. To znamená, že pokud máte v plánu změnu designu webu, stačí změnit pouze jeden soubor a změna se aplikuje na všechny dokumenty, ve kterých je soubor připojen. Také se soubor CSS uloží do mezipaměti prohlížeče a pokud není změněn, tak se načítá pouze jednou a zobrazení stránek se velmi urychlí (citace[3]).

Hlavní nevýhodou CSS je zatím stále špatná podpora v nejrozšířenějších prohlížečích. Různé prohlížeče interpretují stejný CSS kód jinak a je někdy velmi obtížné jej napsat tak, aby se ve všech prohlížečích výsledek zobrazil stejně.

Pomocí CSS bude dělán design celého informačního systému.

3.1.4 JavaScript

JavaScript je programovací jazyk, který se používá v internetových stránkách. Zapisuje se přímo do HTML kódu, což je velká výhoda, protože je to jednoduché. JavaScript je klientský skript. To znamená, že se program odesílá se stránkou do prohlížeče klienta a teprve tam je vykonáván. (Protikladem klientských skriptů jsou skripty serverové, které jsou vykonávány na serveru a na klienta jdou už jen výsledky.).

Existují i jiné jazyky klientských skriptů, například VBScript. Jsou ale tak málo používané, takže když se dnes mluví o "skriptech", myslí se tím JavaScripty. Většina lidí, kteří se s ním nesetkali, si myslí, že se jedná o druh (jednodušší) Javy, ale není to tak. Jedná se o jiný programovací jazyk, který má pouze podobnou syntaxi.

3.1.5 XML

XML (eXtensible Markup Language, česky-rozšiřitelný značkovací jazyk) je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a široké spektrum různých typů dat.

Vlastnosti jazyka:

- Standardní formát pro výměnu informací
- Mezinárodní podpora
- Vysoký informační obsah
- Snadná konverze do jiných formátů

Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů, umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí. Prezentace dokumentu (vzhled) se potom definuje připojeným stylem (citace[3]).

3.2 Spuštění aplikace

Aplikace nemá vlastní grafické uživatelské rozhraní, proto je třeba využívat některý z internetových prohlížečů, které jsou zdarma k dispozici na internetových stránkách. Aplikace je tvořena a testována v několika prohlížečích:

- Opera v. 9.10
- Mozilla Firefox v. 2.0
- Internet Explorer v. 6 a 7

Vzhledem k tomu, že tyto prohlížeče používá drtivá většina všech uživatelů, považoval jsem za dostatečné aby aplikace správně fungovala v těchto výše zmíněných prohlížečích.

3.3 Uživatelské prostředí

Vzhledem k typu projektu je nezbytně nutné rozdělení přístupu pro různé skupiny uživatelů. Pokud uživatel nebude přihlášen, budou mu k dispozici pouze internetové stránky společnosti se všeobecnými údaji, formulář pro registraci zákazníků, přihlašovací formulář a možnost pro zapomenuté heslo zákazníka.

3.3.1 Výběr uživatele

Při vstupu do informačního systému se bude muset každý uživatel, pomocí svého přihlašovacího jména a hesla, přihlásit. Systém sám pozná zda se jedná o jeden z možných typů povoleného přístupu (zákazník, zaměstnanec-administrátor, nebo zaměstnanec-pracovník) a poté přesměruje uživatele do části informačního systému, která je pro něj určena.

Přihlaseení

Login : Heslo :

[Registrace](#) | [Zapomenute heslo](#)

Obrázek 3.1: Přihlašovací formulář

Zde je možná registrace. Zákazník vyplní všechny potřebné údaje (mimo jiné i přihlašovací jméno a heslo) a po úspěšné registraci se může ihned přihlásit do systému.

Pokud zákazník zapomene heslo, bude mít možnost si nechat vygenerovat nové a nechat si jej zaslat na svůj email, který uvedl při registraci.

3.3.2 Hlavní okno systému

Hlavní okno systému je takřka stejné pro všechny uživatele. Je rozděleno do několika částí, z nichž každá se trochu liší podle typu uživatele.

- Hlavička
- Navigační menu sekcí
- Pole pro zobrazení dat
- Patička

The screenshot shows the main interface for an administrator. At the top, there are two summary tables: 'Aktuální práce' (Current Work) and 'Nevyplacená faktury' (Unpaid Invoices). The 'Aktuální práce' table has columns for ID, date, status, location, and people. The 'Nevyplacená faktury' table has columns for ID, customer name, invoice number, due date, and amount. On the right, the user 'tomas henkl' is logged in with a 'Logout' button. A sidebar on the left contains a menu with categories like 'Zákazníci', 'Zaměstnanci', 'Práce', 'Dopravce', 'Stavby', 'Osobní', and 'Společnost'. The main content area is titled 'Přehled zákazníků' (Customer Overview) and includes search filters for name, company name, and month, along with a 'Najdi' (Find) button. Below the filters is a table of customers with columns for ID, login, name, location, address, phone, company name, ID number, email, and number of orders. The table contains two entries.

id	login	jméno	lokace	adresa	telefon	jméno firmy	id	email	počet zak.
1	toma	toomas henkl	Olomouc	První 1	77000	Tomas Henkl	712516910	toomas@stredobdel.cz	1
2	paol	Pavel Paol	Olomouc	Levná 3	77200	Pavel Paolova	308123765	paol@stredobdel.cz	0

Obrázek 3.2: Hlavní okno (administrátor)

Hlavička

Horní část stránky je v případě administrátora zaplněna informacemi o nejbližších pracích, které jsou v systému zaznamenány. Jsou zde také informace o fakturách, které ještě nebyli zákazníky zaplacený. V ostatních případech je zde pouze jméno, datum a políčko logout (odhlásit).

Navigační menu sekcí

Navigační menu je seznam odkazů na hlavní části systému, který je uživateli přístupný. Administrátor zde má odkazy do sekce zákazníků, zaměstnanců, prací, staveb, dopravců, osobních údajů a údajů firmy. Jedná se tedy o komplexní zprávu všech prvků systému.

Zákazník má pomocí menu zpřístupněno zadávání nových zakázek, přehled zakázek které již byli dokončeny, správu faktur, správu osobních údajů. Z pohledu pracovníka je možné se pomocí hlavního menu podívat na práce a osobní údaje.

Pole pro zobrazení dat

Většina operací s tímto informačním systémem se děje právě v tomto poli. Jedná se o největší zobrazovanou část, kde se zobrazují veškeré operace které uživatel provádí a veškeré informace, které uživatel od systému požaduje. V horní části se zobrazuje menu pro navigaci v aktuální sekci, pomocí kterého může uživatel získat veškeré dostupné informace.

Patička

Spodní část informačního systému je použita pro mapu celého systému. Je to z důvodu rychlého zobrazení potřebných dat.

3.3.3 Správa dat

Pro správné fungování celého informačního systému je správa dat velice důležitá. Pokud by se některá data ukládala špatným způsobem, do špatných tabulek nebo neúplná, pokud bychom při odstraňování nepotřebných dat neprovedli tuto operaci právně, celý informační systém by přestal plnit svoji správnou funkci.

Nová data

Při ukládání nových dat je nesmírně důležité aby všechny zadávané údaje odpovídali datům které do informačního systému chceme ukládat, a proto je každý vstup kontrolován zdali jsou všechny údaje vyplněny ve správném tvaru a zdali nepřesahují zadané meze. Důkladná kontrola je zavedena hlavně pokud jde o zadávání zakázek ze strany klienta, který nemusí být obeznámen se všemi postupy a nabídkami ve firmě.

Pro vkládání nových dat jsou pro uživatele nachystány přehledné formuláře, které jsou velice pečlivě popsány. Je zde zvýrazněno co musí uživatel vyplnit, jaká data a v jakém formátu jsou požadována. Při špatném zadání, je na chybu upozorněn a může špatně zadanou část formuláře opravit.

Úprava položek

Úprava vložených dat je další funkcí systému. Modifikaci je možno provést takřka se všemi daty, které jsme do systému vložili a to za pomoci velice podobných formulářů jako při vkládání. Políčka jsou již předem vyplněna a stačí jen upravit danou položku.

V případě upravování osobních údajů si ovšem lze změnit všechny údaje, ale přihlašovací jméno (login) musí být stále jedinečné. Heslo se mění v jiném formuláři než osobní údaje, musí být zadáno nejdříve původní heslo a potom dvakrát heslo nové. Veškeré změny se projeví okamžitě.

Úprava všech položek je velice jednoduše dostupná, nachází se na příslušném řádku u každé zobrazované položky jako odkaz.

Odstranění dat

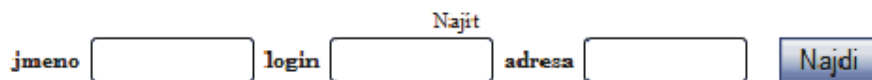
Je nejkritičtější operací s daty. Musí být dodržena veškerá pravidla pro odstraňování dat ze systému (databáze). Pokud budeme mazat zákazníka nesmíme zapomenout vymazat všechny akce objednané tímto zákazníkem a všechny stavby na všech těchto akcích. Podobně je tomu i při odstraňování prací, pokud se smaže stavba, musí se smazat i příslušné bourání, při odstraňování zaměstnanců nesmí jít odstranit administrátor, který je poslední nebo který je zrovna přihlášen.

Pokud chce uživatel některá data odstranit, stačí u konkrétní položky klepnout na odkaz smazat a operace bude provedena.

3.3.4 Vyhledávání

Pro rychlé vyhledání v konkrétní tabulce, u kterých se to předpokládá, je připraveno vyhledávání. Není nutné vyhledávat v položkách u kterých víme, že nepřesáhnou počet deset. Příkladem může být třeba nezaplacené faktury nebo rozpracované práce.

Vyhledávat se může podle jednoho nebo podle více polí a to aniž bychom znali celé konkrétní vyhledávané slovo. Stačí zadat pouze počáteční písmeno vyhledávaného slova. Probíhá pomocí SQL příkazu LIKE místo použití rovnítka.



The image shows a search form with three input fields and a button. The first field is labeled 'jméno', the second 'login', and the third 'adresa'. Above the second field, the word 'Najit' is written. To the right of the third field is a blue button with the text 'Najdi'.

Obrázek 3.4: Formulář pro vyhledávání (v zaměstnancích)

3.3.5 Archivace

Archivace jednotlivých dat v databázi je velice důležitou funkcí systému. Pokud by se databáze nějakým způsobem vymazala, bylo by možné z archivovaných dat rekonstruovat zálohovanou část databázi. Není snahou archivovat všechna data uložená v databázi. Tímto procesem prochází jen data která by již nešla znovu obnovit.

Mezi data která jsou archivována patří:

- Zákazníci
- Akce
- Faktury
- Zaměstnanci

- Stavby

Archivace probíhá na popud administrátora, který pomocí odkazu v sekci společnost exportuje jednotlivě každou tabulku zvlášť do samostatného souboru. K archivaci jsem použil, dnes již hojně využívaný značkovací jazyk, XML.

3.3.6 Statistiky

Statistiky jsou implementovány v sekci staveb a jedná se o přehled využití jednotlivých typů staveb v průběhu roku. Data k tomuto účelu jsou získávána z tabulky (pujcene), která obsahuje veškeré informace o půjčených stavbách a je proto k tomuto účelu nejvhodnější. Vše je sestaveno pomocí HTML a PHP, ne však GD funkcí, proto výsledkem nebude obrázek, ale HTML konstrukce.

Statistiky jsou umístěny v sekci stavby. Pomocí formuláře je možno zobrazit více grafů v jednom požadavku pro srovnání využití jednotlivých staveb v nastaveném roce. Tato funkce je dostupná pouze z administrátorského účtu.

3.3.7 Tisk

Jako důležitý bod informačního systému lze považovat nějaký druh výstupu, v jiné než elektronické formě. Z tohoto důvodu byl přidán tisk důležitých věcí, které je dobré mít v papírové formě. Vše je nastaveno pomocí CSS.

Jedná se především o tisk práce, který je dostupný buď z náhledu práce nebo po provedení práce. Obsahuje veškeré stavby, které se budou stavět, datum a místo práce, seznam pracovníků, kteří se zaregistrovali, kdo dopravuje materiál a jeho telefonní číslo.

Tisk objednávky pro zadavatele je položka dostupná až po potvrzení akce zadavatelem. Obsahuje jméno objednatele, adresu, kde se akce koná, kdy se koná a co si zákazník objednal.

Tisk faktury pro zaplacení objednávky obsahuje číslo faktury důležité údaje o zákazníkovi a o stavební společnosti, údaje o stavbách, jejich cenu, celkovou cenu a další důležité údaje pro zaplacení faktury. Tato volba je dostupná u všech faktur (zaplacených i nezaplacených) v informačním systému.

3.4 Problémy při implementaci

Při implementaci jsem se setkal s několika menšími problémy, které vznikly díky nezkušenosti při tvorbě velkých samostatných projektů. Mezi největší problémy patřilo:

- Kontrola půjčených staveb – při návrhu aplikace jsem na tuto skutečnost zapomenul, proto jsem při implementaci dlouho řešil jak budu kontrolu provádět. Nakonec jsem přidal do systému další tabulku (pujcene).
- Kontrola volných pracovníků – jedná se o kontrolu zda-li se pracovník nezaregistroval na dvě práce ve stejnou dobu na jiné místo. Po mnohých konzultacích s možným uživatelem tohoto

systemu jsem se rozhodl tento problém zatím neřešit z důvodu jeho složitosti. Kontrolu musí provést pracovník nebo administrátor.

Vzhledem k charakteru aplikace, jedná se o internetovou aplikaci, jsem se samozřejmě nevyhnul problémům při zobrazení internetových stránek na kterých bude systém fungovat. Jedná se především o prohlížeč Internet Explorer 6 a 7, které nedodržují standardy pro zobrazování internetových stránek.

4 Možná rozšíření

Sklad součástí

Každá stavba je složena ze součástí, na které se musí rozložit aby bylo možné stavbu skladovat a přepravovat. Tato část by se zabývala možnými půjčeními jednotlivých součástí pro potřeby nějaké jiné podobné společnosti, která by si nechtěla půjčit celou stavbu a taky pro potřeby společnosti, která by tímto měla přehled o všech součástkách. Bylo by to svázáno s půjčováním staveb. Z nedostatku informací od společnosti a nedostatku času nebyla tato část implementována.

Zasílání nových prací e-mailem

Pokud by pracovník měl zájem, tak by se jemu poslal e-mail ihned, pokud by zaměstnanec vytvořil práci s veškerými informacemi o nově vzniklé práci. Ten by díky tomu věděl o každé práci ihned po jejím vytvoření.

Tato funkce není obtížná na implementaci, ale je zatím zpochybněna její užitá hodnota. Dnes ještě ne každý má možnost si posílat e-mail na telefon nebo jiný způsob, tak aby to věděl ihned a nemusel kvůli tomu na internet, potom je již jedno jestli se dívá na e-maily nebo rovnou do tohoto systému.

Rozšíření statistik

Mezi další potenciální údaje pro které by se dali vytvořit statistiky by mohly patřit celkový objem půjčených staveb v penězích za určité časové období.

Statistiky by se dali implementovat i pro další důležitá data, která by uživatel potřeboval, ale jednalo by se pouze o použití stejného algoritmu s jinými daty a kvůli nedostatku času nejsou další statistiky k dispozici.

Závěr

Tento informační systém má za úkol pomáhat, zjednodušovat, zpřehledňovat a standardizovat práci v malé společnosti, která žádný takový systém nemá. Nejedná se přímo o práci na zakázku, ale chtěl jsem si při tvorbě systému vyzkoušet jaké je tvořit informační systém pro konkrétního uživatele. Snažil jsem se o co možná nejužší spolupráci s touto společností, aby celá tato aplikace neskončila mezi nepoužitelnými projekty někde „šuplíku“.

Výsledek se zatím ideálnímu příliš neblíží, ale je možné další práci tento projekt upravit do podoby, která bude uživateli vyhovovat ve všech důležitých bodech jeho práce a bude možné ji z pohledu programátora rozumným způsobem realizovat. Zatím se samozřejmě nemůže rovnat profesionálním pracím, to ale nebude možné bez zkušeností získaných v takovýchto projektech. Z tohoto důvodu budu na projektu dále pracovat. Jsem domluven i s jedním ze zaměstnanců společnosti, který mi svoji spolupráci přislíbil i nadále.

Tato práce pro mě byla velikým přínosem, zvláště pak právě spolupráce s koncovým uživatelem. V neposlední řadě práce s jazykem SQL při formulacích dotazů v systému MySQL. Udělal jsem velký pokrok v programování webových aplikací pomocí jazyka PHP. Bylo to mé první setkání s jazykem JavaScript, který jsem se alespoň trochu naučil.

Celá aplikace byla kompletně testována a během tohoto procesu se neobjevili žádné závady, které by znamenali nefunkčnost systému. I přes nedostatky, které aplikace jistě obsahuje, a porovnání se zadanými požadavky, jsem z větší části přesvědčen, že se mi podařilo naplnit všechny body zadání úspěšně.

Literatura

- [1] Zendulka J., Rudolfová I.: Databázové systémy, Brno, 2006, Vysoké učení technické v Brně, Fakulta informačních technologií, [online],
URL<https://www.fit.vutbr.cz/study/courses/IDS/private/IDS_predn.pdf>
- [2] Hruška T.,Křivka Z.: Analýza, návrh a implementace informačního systému, Brno, 2007,Vysoké učení technické v Brně, Fakulta informačních technologií, [online],
URL<<https://www.fit.vutbr.cz/study/courses/WAP/private/opory/OporaPISAnalyzaNavrhImplementace.pdf>>
- [3] Wikipedie otevřená encyklopedie. (květen 2007)
URL <<http://cs.wikipedia.org/>>
- [4] Stein, R.: Návrh aplikací v jazyce UML - Unified Modeling Language. [online], 5 2003,
URL <<http://interval.cz/clanky/navrh-aplikaci-v-jazyce-uml-unified-modeling-language/>>

Seznam příloh

Příloha 1. Manuál viz přiložené CD

Příloha 2. Zdrojové texty viz přiložené CD

Příloha 3. CD