

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

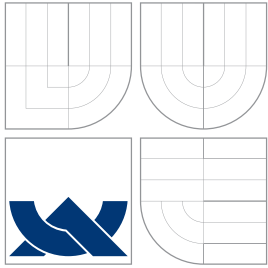
INFORMAČNÍ SYSTÉM FIREMNÍ NÁŘAĎOVNY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

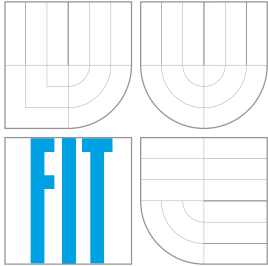
AUTOR PRÁCE
AUTHOR

PAVEL SVOBODA

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 FIREMNÍ NÁŘAĐOVNY

INFORMATION SYSTEM FOR A POTTING SHED OF A FIRM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL SVOBODA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VLADIMÍR BARTÍK, Ph.D.

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Svoboda Pavel**

Obor: Informační technologie

Téma: **Informační systém firemní nářadovny**

Kategorie: Databáze

Pokyny:

1. Analyzujte požadavky kladené na informační systém firemní nářadovny. Informační systém by měl zajišťovat základní operace s databází nářadí a zaměstnanců, kteří si nářadí půjčují. Dále by měl zajišťovat např. zobrazení dat ve formě tiskových sestav a bezpečné přihlašování uživatelů.
2. Na základě analýzy navrhnete koncepci takového informačního systému. Využijte modelovací jazyk UML.
3. Implementujte navrženou koncepci a ověřte funkčnost systému.
4. Zhodnoťte dosažené výsledky, porovnejte je s již existujícími podobnými systémy. Diskutujte možné pokračování v tomto projektu.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Body 1-2

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).


Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bartík Vladimír, Ing., Ph.D.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Požetšchova 2
L.S.


doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Pavel Svoboda**
Id studenta: 84135
Bytem: Stavy 905, 735 34 Stonava
Narozen: 27. 09. 1985, Karviná
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Informační systém firemní nářad'ovny
Vedoucí/školitel VŠKP: Bartík Vladimír, Ing., Ph.D.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.


Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel


.....
Autor

Abstrakt

Bakalářská práce pojednává o návrhu a implementaci informačního systému firemní nářadovny. Informační systém umožňuje vkládat zaměstnance a inventární nářadí, zapůjčovat zboží, upravovat či smazat údaje. Technická zpráva uvádí přehled a stručný popis použitých nástrojů a technik, konkrétně HTML, PHP, JavaScript, CSS, MySQL a Apache. Dále uvádí popis a implementaci tiskových sestav za využití exportování do souboru PDF, generování asociačních pravidel z frekventované množiny a import poštovních směrovacích čísel ve formátu CSV. Je také popsán návrh pomocí diagramů jazyka UML, Use Case diagram a Class diagram. Na závěr je uvedeno zhodnocení a rozšíření systému, diskuze nad podobnými systémy nebo problémy, které vznikly při implementaci.

Klíčová slova

Informační systém, firemní nářadovna, webová aplikace, HTML, PHP, JavaScript, CSS, MySQL, Apache, UML, Apriori, tiskové sestavy, (U)FPDF, import z CSV.

Abstract

The main goal of this work is to design and implement information system for potting shed of a firm. Information system allows you to create an employees or inventory stuff, lending stuff, modification or deleting stuff. The document contents a list and short description of used technologies as HTML, PHP, JavaScript, CSS, MySQL and Apache. Description and implementation of the print lists by using exporting to PDF files, generation association rules from frequented set and importing ZIP code saved in CSV format. It also contents description of the system via UML diagrams, Use Case diagram and Class diagram. On the end of document there is a short summary of the project and discussion about innovation of the system, comparing with another systems and problems occurring during the implementation.

Keywords

Information system, potting shed of a firm, web application, HTML, PHP, JavaScript, CSS, MySQL, Apache, UML, Apriori, print lists, (U)FPDF, import from CSV.

Citace

Pavel Svoboda: Informační systém firemní nářadovny, bakalářská práce, Brno, FIT VUT v Brně, 2007

Informační systém firemní nářad'ovny

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval zcela samostatně pod vedením pana Ing. Vladimíra Bartíka, Ph.D. a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Svoboda
12. května 2007

Poděkování

Rád bych tímto poděkoval panu Ing. Vladimíru Baríkovi, Ph.D. za odborné vedení mé práce, za rady, pomoc a čas, který mi během vypracovávání této práce věnoval.

© Pavel Svoboda, 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

1	Úvod	3
1.1	Informační systém	3
1.2	Cíl firemní nářadovny	3
1.3	Analýza požadavků	3
1.4	Přehled	4
2	Použité nástroje a techniky	5
2.1	HTML	5
2.2	PHP	6
2.3	JavaScript	6
2.4	CSS	7
2.5	MySQL	7
2.6	Apache	7
3	Třída (U)FPDF	9
3.1	Úvod	9
3.2	GNU - General Public License	9
3.3	Historie	9
3.4	Využití	9
3.5	Instalace	10
3.6	Boj formátů	11
	3.6.1 Druhy formátů	11
	3.6.2 Počet výskytu	11
4	Asociační pravidla	12
4.1	Využití	12
4.2	Teorie	12
4.3	Typy asociačních pravidel	13
4.4	Algoritmus Apriori	13
4.5	Generování asociačních pravidel	15
5	Návrh systému	17
5.1	UML	17
5.2	Program	17
5.3	Use case diagram	18
	5.3.1 Tvorba	18
5.4	Class diagram	20

6 Implementace	22
6.1 Funkce	22
6.1.1 Vložit	22
6.1.2 Tabulky	22
6.1.3 Vyhledávání	22
6.1.4 Půjčit a vrátit zařízení	23
6.1.5 Smazat a editovat	23
6.2 Práva	23
6.3 Import PSC	24
6.4 Tiskové sestavy	24
6.5 Struktura souborů	25
7 Zhodnocení	26
7.1 Podobné systémy	26
7.2 Rozšíření informačního systému	26
7.3 Problémy při implementaci	26
8 Závěr	29
Seznam použitých zdrojů	31
Seznam příloh	32
A Instalace	33
B Ukázka vzhledu pro přihlášení	34
C Ukázka vzhledu tabulky	35
D Obsah přiloženého CD	36

Kapitola 1

Úvod

1.1 Informační systém

Informační systémy by se daly zařadit mezi důležité hodnoty v rozvoji společnosti. Podle počtu, rozsahu a možností informačního systému se dá usoudit, jak na tom jednotlivé firmy, podniky, města, kraje, či stát jsou. Je velmi smutné, že někteří z nich ještě nepochopili jaké možnosti nám informační systémy dávají a hlavně kolik času a práce ušetří.

Informační systémy se stále zdokonalují. To, co platilo před půl rokem jako novinka, dnes při větší zakázce informačního systému považují větší firmy za standard. Proto velikost a rozsah jedné aplikace roste. Není divu, že některá menší (ale moderní) firma by také ráda zahodila sešity, papíry a tužky a vrhla se do plného využití informačního systému, který by za ní zpracovával například docházku, nákup, prodej, personální údaje o zaměstnancích. . .

1.2 Cíl firemní nářad'ovny

Nabídnout zaměstnancům a vedení firmě informační systém, který ji umožní lepší manipulaci s zapůjčeným zbožím, správu osobních informací zaměstnanců a možnost lepšího přeskupení zboží uloženého v místnostech pomocí funkce pro zjišťování nejčastěji zapůjčených kombinací daného zboží a ohodnocení silných transakcí. Tím docílit efektivnější práci s nářadím, urychlení práce a zbavení se zastaralých věcí jako je tužka a papír.

1.3 Analýza požadavků

Informační systém firemní nářad'ovny obsahuje správu všech zaměstnanců, celkového nářadí a aktivit zapůjčených věcí (nářadí). Zaměstnanci mohou být vkládáni do systému, kde je zapotřebí zadat základní údaje o zaměstnanci. Tyto údaje mohou být posléze i upravovány či smazány. Aplikace zahrnuje seznam všech nářadí, které mohou být zapůjčeny. Tato nářadí sebou nesou určité údaje o sobě, které upřesňují jejich informace. Nářadí mohou být taktéž upravena nebo úplně smazána ze systému. Každé nářadí může být zapůjčeno určitou osobou při specifikaci počtu kusů pro zapůjčení. Toto nářadí se pak zobrazí v seznamu „Aktivit“ daného zaměstnance. Nářadí může být vráceno buď částečně, když zaměstnanec vrátí nářadí jen zčásti, nebo úplně.

Informační systém obsahuje správu zaměstnanců do jednotlivých úrovní s omezenými právy, bezpečné přihlašování, tiskové sestavy všech zaměstnanců a všech nářadí v systému, kontrolu nad vkládanými městy a jejich příslušných poštovních směrovacích čísel, dolování

jednoúrovňových booleovských asociačních pravidel z transakčních dat a to základním algoritmem Apriori a generování asociačních pravidel z frekventovaných množin.

1.4 Přehled

Tato technická zpráva odhalí použité nástroje a techniky v kapitole 2, které byly zapotřebí k tvorbě informačního systému. Vysvětlí možnost využití třídy (U)FPDF pro tiskové sestavy v kapitole 3 a jejího bližšího použití v aplikaci. Aplikování algoritmu Apriori a generování asociačních pravidel, která jsou popsána v kapitole 4. V kapitole 5 je popsán návrh systému v jazyce UML a kapitole 6 přístup k implementaci a vysvětlení vybraných funkcí. Problémy, které se vyskytly při implementaci jsou popsány v kapitole 7, která navíc ještě obsahuje diskuzi nad již známými programy a informačními systémy nebo možná vylepšení této aplikace. Na konec kapitola 8 obsahuje celkové shrnutí a přínos.

Kapitola 2

Použité nástroje a techniky

Využil jsem běžné programovací jazyky a techniky pro tvorbu informačního systému. Mezi které patří jazyk HTML (blíže v podkapitole 2.1) pro tvorbu struktury a obsahu stránek, jazyk CSS (blíže v podkapitole 2.4) pro tvorbu vzhledu dokumentu, objektově orientovaný skriptovací jazyk JavaScript (blíže v podkapitole 2.3) pomocí něhož jsou ovládány některé interaktivní prvky grafického rozhraní, jako jsou tlačítka a textová pole, další skriptovací programovací jazyk PHP (blíže v podkapitole 2.2) určený pro dynamickou tvorbu internetových stránek. Na závěr MySQL (blíže v podkapitole 2.5) relační databázový systém pro uchování dat v tabulkách a webový server Apache (blíže v podkapitole 2.6) pro odladění kódu.

2.1 HTML

HTML je zkratka z anglického HyperText Markup Language, značkovací jazyk pro hypertext. HTML patří mezi jedny z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje zobrazovat stránky na Internetu. Tento jazyk patří pod podmnožinu již dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML¹.

HTML jazyk bude obsahovat text dokumentu, propojený značkami, které tento jazyk definuje. Tyto značky mohou být párové nebo nepárové. Pomocí těchto značek se formátuje text, vytváří vzhled a funkce stránky. Obsah HTML dokumentu pak může obsahovat: text, nadpisy, odstavce, drážky, číslované seznamy, tabulky, obrázky, interaktivní prvky. Částečně může obsahovat i vzhled dokumentu, ale tato možnost má řadu nevýhod a nedostatků. Proto se od tohoto upouští a definují se raději kaskádové styly CSS.

Je nezbytné si povšimnout, že vývoj jazyka HTML byl pozastaven už v roce 1999. Jeho vývoj započal v roce 1989 a první funkční verze byla na světě v roce 1991. Osm let je relativně krátká doba na to, aby se mohl tento jazyk dostatečně rozvinout, ale přesto finální verze má označení 4.01. Nyní se vedou spory zda přejít na novější verzi XHTML². Důvodem je ohromné množství dokumentů, které neodpovídají specifikaci HTML. Jelikož současně prohlížeče tolerují uživatelské chyby v značkování (neuzavření, překřížení či špatné zanoření značek), musí se snažit tyto chyby eliminovat tím, že začnou „myslet“ za uživatele a pokoušejí se tyto nedostatky vyrušit. Často tato snaha vede k úplně jinému výsledku, než si uživatel představoval. Tento princip je velmi náročný a pro méně výkonná zařízení

¹Standard Generalized Markup Language - je univerzální značkovací meta-jazyk, který umožňuje definovat značkovací jazyky jako své vlastní podmnožiny.

²Zkratka anglického eXtensible HyperText Markup Language - rozšiřitelný značkovací jazyk pro hypertext.

nepoužitelný. Tato méně výkonná zařízení potřebují přesnou definici struktur a značek. Kdyby se přistoupilo na přesnou definici jazyka HTML, většina internetových stránek by přestala fungovat. Výjimku by tvořila skupina těch dokumentů, které prošly verifikací a validací. [12]

2.2 PHP

PHP³ jak již bylo řečeno je skriptovací programovací jazyk. Tento jazyk se syntaxí přibližuje hned několika programovacím jazykům (Perl, C, Pascal nebo Javě). PHP byl z počátku napsán v Perlu, posléze z důvodu zvýšení rychlosti přepsán do jazyka C. S novou verzí PHP 5 se výrazně zlepšil přístup k objektově orientovanému programování a nyní se více podobá Javě. PHP jazyk není závislý na operačním systému a výsledné skripty mohou bez jakýkoliv úprav fungovat na různých operačních systémech. Jazyk PHP obsahuje rozsáhlou podporu pro zpracování textu, grafiky, práci se soubory, databází nebo podporu celé řady protokolů (HTTP, FTP, POP3).

Nejčastěji se jazyk PHP využívá spolu s databázovým serverem (např. MySQL) a webovým serverem Apache k tvorbě internetových aplikací. Pokud prohlížeč požádá o webovou stránku ze serveru, server tuto stránku nejprve sestaví a posléze odešle. Existují dvě technologie, „klientské“ a „serverové“. Jelikož PHP skripty jsou prováděny na straně serveru a k uživateli je přenášěn až výsledek jejich činností, patří jazyk PHP mezi serverové technologie. Internetový prohlížeč pak nemá ani tušení, co se všechno na serveru odehrálo, je mu zaslán pouze výsledek, který se zobrazí. [18]

2.3 JavaScript

Je objektově orientovaný skriptovací jazyk. Nyní je běžně využíván jako interpretovaný programovací jazyk pro internetové stránky. Je vkládán přímo do HTML kódu stránky. Pomocí něho jsou pak ovládána různá tlačítka, pole a vytvářeny animace a efekty obrázků.

Server zašle spolu s HTML stránkou prohlížeči i nějaké kusy programového kódu a ten se ve vhodné chvíli na klientském počítači spustí. Tato vhodná chvíle může nastat při určité události: pokud uživatel zmáčkne tlačítko, přejede myší přes odkaz. . . JavaScript by se proto dal zařadit do technologie „klientské“, která je oproti PHP spouštěna na cílovém počítači. Provádění kódu tak bývá rychlejší a zatíženost serveru menší než v případě provádění kódu na straně serveru. [13]

Jeho nevýhodou je možnost vypnutí na straně klienta v prohlížeči. Proto by měly být internetové aplikace funkční i bez podpory JavaScriptu. Další jeho nevýhodou je právě stahování programového kódu k uživateli. Toto stahování sebou nese jisté riziko. Programový kód může fungovat tak, jak byl navržen, ale může také představovat jako spyware⁴, napadnout systém jako virus nebo způsobit daleko větší škodu jako krádež informací, smazání osobních dat. . . Prevence proti tomu je zrušit podporu JavaScriptu úplně nebo potvrzovat každé spuštění skriptu na stránkách. To má za následek neustále klikání na okénko se zprávou „povolit“ při každém načtení stránky. [8]

³PHP - je zkratka od Personal Home Page

⁴Spyware - je obecný pojem používaný k popisu softwaru, který se chová určitým způsobem, například zobrazuje reklamu, shromažďuje osobní informace nebo mění konfiguraci počítače, obvykle bez získání vašeho předchozího souhlasu. [9]

2.4 CSS

CSS⁵ je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML, XML. Hlavním smyslem je možnost oddělit vzhled dokumentu od jeho struktury a obsahu. Ve vývoji tento jazyk pokulhává. Je sice pravda, že první verze byla již v roce 1996, ale větší pozorností a uplatnění CSS proběhlo mnohem později.

CSS pracuje tak, že obsah webu je uložen v jednom souboru a veškerý vzhled a formátování se načítá z jednoho nebo více souborů uložených pod příponou .css. Tento soubor může být uložen do mezipaměti prohlížeče a pokud není změněn, tak se načítá pouze jednou a zobrazení stránek se pak velmi urychlí. CSS nabízí větší možnosti formátování než HTML. HTML třeba nenabízí žádnou funkci pro formátování bloku textu - určení vzdálenosti od okrajů stránky. Jistou nevýhodou je, že ne všechny prohlížeče jasně dodržují definici CSS, proto ne každý prohlížeč zobrazí výsledek úplně stejně. [11]

2.5 MySQL

MySQL⁶ je relační databázový systém typu SŘBD⁷ založený na jazyce SQL. Který umožňuje uchovávat informace v databázi. Data v databázích se uchovávají ve formě tabulek. Každý sloupec je definován typem dat, které mohou být v tabulce uloženy. Řádky tabulky jsou pak tvořeny jednotlivými záznamy. Pro jednoznačnou identifikaci daného řádku se využívá hlavního sloupce označeného jako primární klíč. Tento primární klíč pak slouží k odkazování z jiných tabulek pomocí cizích klíčů. Mezi tabulkami mohou existovat relace, které vytvářejí vztah mezi jednotlivými sloupci. [14]

Jednou z předních výhod MySQL je, že je k dispozici jako open source, tedy program šířený zdarma. K dalším výhodám MySQL patří podpora všech hlavních platforem, vysoký výkon i rychlost a vynikající kompatibilita s jinými systémy, zejména se serverovým programem Apache (o kterém bude zmínka v následující podkapitole) a skriptovacím PHP (který byl zmíněn v podkapitole 2.2). MySQL se díky své relativní jednoduchosti poměrně snadno učí. Mezi nevýhody MySQL jednoznačně patří nedostatečný výkon v opravdu náročných (zatěžovaných) webových aplikacích. Tehdy se používají konkurenční databáze, například Oracle⁸ a jiné. [3]

Pro jednoduchou správu MySQL databází se používá nástroj PhpMyAdmin. PhpMyAdmin je Open Source program napsaný v PHP, který umožňuje zálohování, vytváření tabulek, vkládání, editaci a mazání záznamů v tabulkách, vytváření databází apod. PhpMyAdmin je pokročilý nástroj pro kompletní správu MySQL systému přes webové rozhraní. [20]

2.6 Apache

Apache⁹ HTTP Server je softwarový webový server s otevřeným kódem a proto podporuje moderní operační systémy včetně UNIX a Windows NT. Předností Apache je především že je nabízen jako open source, tedy zdarma a také schopnost zpracovávat nejnovější protokoly,

⁵Cascading Style Sheets - tabulky kaskádových stylů.

⁶MySQL - je zkratka z angl. My Structured Query Language = systém pro řízení databází. [20]

⁷SŘBD - Systém řízení báze dat.

⁸Databázový systém Oracle je vyvíjen firmou Oracle Corporation, moderní multiplatformní databázový systém s velice pokročilými možnostmi zpracování dat, vysokým výkonem a snadnou škálovatelností. [16]

⁹Apache - vzniklo ze spojení A PATCHy server

je vysoce konfigurovatelný, je možno psát vlastní moduly s použitím Apache API... Není divu, že jeho podíl na trhu je v roce 2006 (podle společnosti Netcraft) téměř 70%.

Při vývoji aplikací slouží hlavně k možnosti spustit a otestovat internetové aplikace na vlastním počítači (není třeba složitě nahrávat a testovat na vzdáleném serveru). Existují dva způsoby konstrukce virtuálních serverů: buď na základě IP adresy (IP-based) nebo využitím vlastností protokolu HTTP 1.1 (na základě jména - Name-based). Apache svou bezpečností a spolehlivostí může posloužit také pro jiné účely. Jedním z příkladů je sdílení souboru z osobního počítače dál na internet. Uživatel, který má Apache nainstalovaný na svém počítači může nahrát soubory do hlavní dokumentové složky Apache, která bude sdílena na internetu. [10, 4]

Kapitola 3

Třída (U)FPDF

3.1 Úvod

Pro tiskové sestavy jsem využil třídu (U)FPDF, která je volně dostupná na internetu, na stránce www.fpdf.org pod licenci GNU - General Public License, GNU GPL. Umožňuje sestavit výstup do formátu PDF¹.

3.2 GNU - General Public License

Tato licence zaručuje, že software pod touto licencí se může volně šířit, vytvářet kopie, modifikovat jej nebo jeho části použít v nových programech. Pokud je software upraven a distribuován je nutné, aby příjemci věděli, že se nejedná o originál a musí být přiložena licenční smlouva. Kvůli volné ochraně, redistributoři pokud chtějí vydat softwarový patent, musí zajistit, že každý patent musí být buď vydán s tím, že umožňuje každému volné užití, anebo nesmí být vydán vůbec. [6]

3.3 Historie

První verze 1.01 FPDF vznikla v roce 2001. Následovala rozšíření jako byla vkládání obrázků, přidávání vlastních fontů, práce s tabulkami - metody pro nastavování počátků tabulky nebo získání zpět pozici tabulky, vkládání dat do vlastností PDF výstupu jako je autor, klíčová slova, nadpis... V roce 2002 nová správa s formátem výstupu A4, A3, dále možnost definovat okraje v tabulce, ukončení podpory pro PHP 3 a podporování už pouze jen PHP 4, vytváření odkazů v dokumentu PDF nebo v roce 2003 a 2004 přidáno několik dekodování pro východní jazyky a práci s výstupem. [7]

3.4 Využití

Třída (U)FPDF umožňuje vytvořit výstupní soubor v formátu PDF. Jelikož jsem zvolil jako hlavní kódování UTF-8, musel jsem ještě tuto třídu rozšířit o třídu UFPDF, která je taktéž volně dostupná na internetu, na stránce <http://acko.net/blog/ufpdf?page=1> pod licenci GNU, neboť třída FPDF neumožňuje práci s kódování UTF-8.

¹Formát PDF - Portable Document Format (Formát pro přenositelné dokumenty) někdy často špatně zaměňován se zkratkou Printable Document Format (Formát pro tiskové dokumenty).

Jedním z hlavních důvodů proč jsem využil tuto třídu pro tiskovou sestavu jsou:

- Výstup bude vypadat na všech počítačích stejně a to díky tomu, že při vytváření konečného PDF souboru se definují hlavní věci jako je typ písma, velikost písma, záhlaví, zápatí a hlavně velikostí okrajů tak, aby výstup vypadal na všech počítačích stejně.
- Přenositelnost - formát PDF lze otevřít v programu Acrobat (Adobe) Reader, který je volně dostupný na internetu a je zadarmo. Není třeba se strachovat při rozesílání inventárního náradí či seznamu zaměstnanců emailem, zda výstupní soubor půjde či nepůjde danému člověku otevřít.
- Zapouzdřenost - třída obsahuje základní příkazy pro práci s výstupním souborem v PDF. Tyto příkazy mohou být dále upraveny či zdokonaleny.
- Rozšiřitelnost - možnost rozšířit třídu o své vlastní funkce.

3.5 Instalace

K plnému využití výstupu do souboru PDF v češtině je zapotřebí si obstarat své vlastní písma. Balíček třídy FPDF obsahuje složku dokumentace, základní písma pro práci vygenerovaná z anglických písem a tedy neobsahují háčky a čárky, na konec výukový program. Tento balíček neobsahuje programy potřebné pro vygenerování vlastního písma, ty je třeba stáhnout zvlášť. Zato balíček třídy UFPDF již tyto programy obsahuje implicitně. Pro změnu neobsahuje žádná hotová písma. Tato písma je třeba si vygenerovat pomocí obsažených programů. K vygenerování písma je zapotřebí dvou programů a české fonty (*.TTF). Pro balíček FPDF jsou to tyto programy a skripty *ttf2pt1.exe* a *makefont.php* s využitím spouštěcího programu *php.exe* a pro balíček UFPDF jsou to tyto upravené programy a skripty *ttf2ufm.exe* a *makefontuni.php* s využitím spouštěcího programu *php.exe*.

Postup pro generování písma pro třídu UFPDF je následovný:

1. Vygenerování tabulky znaků do souboru .ufm za pomoci programu *ttf2ufm.exe* a vstupního písma. Soubor .ufm bude obsahovat seznam znaků, velikost daného znaku, jeho číselnou podobu a jmenné označení. Ukázka použití, kde *myfont.ttf* je písmo pro vygenerování (např: arial.ttf):

```
ttf2ufm.exe -a -F myfont.ttf
```

2. Spuštění skriptu *makefontuni.php* s vstupní hodnotou písma .ttf a vygenerovaného souboru .ufm. Skript vygeneruje soubor .php a dva binární zabalené soubory. Soubor .php bude obsahovat základní informace o písmě, které bylo generováno a hlavně převede seznam znaků z souboru .ufm do pole potřebné pro třídu UFPDF. Ukázka spuštění skriptu:

```
php.exe -q makefontuni.php myfont.ttf myfont.ufm
```

Jako poslední krok stačí nakopírovat výsledné soubory .php, .z a .ctg.z do adresáře, kde se nacházejí písma pro třídu (U)FPDF. Písma mohou být do třídy načtena a používána pod názvem MyFont. Ukázka:

```
pdf->AddFont('MyFont', '', 'myfont.php');
```

3.6 Boj formátů

Tvorba tiskové sestavy do souboru PDF nabízí novou možnost rychlého rozšiřování dokumentů po internetu. Při bližším zkoumání každý člověk pochopí, že formát PDF má a bude mít vždy dost svých příznivců. Dokumenty na internetu se vystavují právě převážně v formátu PDF.

3.6.1 Druhy formátů

Formát PDF se nedá srovnat v formáty jako je DOC², který je hlavně využíván lidmi, kteří mají převážně nainstalovaný operační systém Windows nebo s otevřeným formátem ODF³ nebo PS⁴, který je sice volně dostupný a využívají ho hlavně příznivci Linuxu či Unixu, ale „zatím“ v České republice nezažil dostatečný rozvoj a mohl tak konkurovat ostatním formátům. Ještě bych měl zmínit formáty (X)HTML, který umožňuje ukládání rozsáhlejších prací, ale jeho hlavní nevýhodou je, že konkrétní způsob zobrazení stránky přenechá na libovůli prohlížeče, potažmo uživatele. Má to jisté „výhody“. Zobrazený dokument si čtenář může upravit tak, aby mu vyhovoval. Na druhou stranu mu znemožní zaručit u všech lidí přesně stejný vzhled. [17]

Typickým příkladem rozdílů je chování při zvětšování dokumentu, např. pro lepší čitelnost pro osoby s vadou zraku:

- Při zvětšení PDF dokumentu se pouze zvětšuje měřítko, vzhled dokumentu zůstává zachován; při vysokém zvětšení se na obrazovku nevejde celý sloupec textu a je třeba horizontální posuv pro čtení celého řádku.
- Pro zvětšení HTML dokumentu stačí zvětšit velikost použitého písma, text se přeformátuje tak, aby se stále vešel do šířky okna prohlížeče, při tom se ovšem vzhled dokumentu změní, řádkové zlomy jsou mezi jinými slovy, dokumentu se zvýší počet řádků textu atd. (Poznámka: novější verze formátu PDF umožňují vytvářet dokumenty, které se mohou také přeformátovat stejným způsobem, tato možnost však obvykle využívána není.)

3.6.2 Počet výskytu

Jako poměr počtu výsledků jsem použil velmi dobrý a všem známý fulltextový vyhledávač Google, který vyhledává webové stránky v internetu. Jedná se o stránku: <http://www.googlefight.com/index.php>, která ze dvou klíčových slov vytvoří graf, kde jako velikost udává počet výskytů daného slova na internetu. Člověk si pak může udělat představu kolikrát se dané slovo „přibližně“ na internetu objevuje.

Příkladem je formát PDF, který se na internetu vyskytuje v desítkách miliard výskytů. Oproti tomu formát DOC ve stovkách milionech výskytů. Je zase pravda, že formát PDF, může zahrnovat tabulkové výstupy, čili bych k formátům DOC měl připočítat formát XLS, který se vyskytuje „pouze“ v desítkách milionů výskytů.

²DOC - formát firmy Microsoft produktu MS Word

³ODF - neboli OASIS Open Document Format for Office Applications (OASIS otevřený formát dokumentu pro kancelářské aplikace) [15]

⁴PS - PostScript je programovací jazyk určený ke grafickému popisu tisknutelných dokumentů [19]

Kapitola 4

Asociační pravidla

Asociační pravidla se zabývají hledáním zajímavých asociací nebo korelací nad velkými množinami datových položek. Tyto asociace mohou pomoci k růstu firem nebo k lepšímu marketingu.

Mezi typické příklady pro získávání asocičních pravidel je analýza nákupního košíku, v mém případě to bude analýza zapůjčených věcí ve firemní nářadovně. Asociační pravidla hledají různé vazby mezi jednotlivým zapůjčeným zbožím a zjišťují, která zboží si zaměstnanec nejčastěji zapůjčí dohromady. Zapůjčili si zaměstnanec brusku a vrtačku, s jakou pravděpodobností si také zapůjčí sadu vrtáků a kotoučů?

4.1 Využití

V firemní nářadovně se tato analýza využije zejména ve skladování věcí a tedy k zefektivnění práce se zapůjčeným zbožím. Většina podniků má své zboží rozmístěno v různých skladech a odděleních, popřípadě i sekcích. Při přenosu nejčastěji používaných zboží blíž k sobě dojde k rychlejšímu odbytu a lepší efektivitě práce.

Ve firemní nářadovně jsem tuto analýzu využil k tomu, aby ředitel nebo jiný oprávněný uživatel mohl zařídit přemístění nejčastěji zapůjčených zboží blíže k sobě. Je velmi nelogické, nastane-li situace, že právě již zmíněná bruska s vrtačkou se stanou nejčastěji zapůjčeným zbožím, ale sada vrtáků a kotoučů bude umístěna v jiném skladě, kde zaměstnanec musí zdolat 10 minutovou trasu jen pro vyzvednutí této sady.

4.2 Teorie

Máme-li množinu všech položek nabízených v inventáři, pak každá z nich může být reprezentována booleovskou proměnnou, která reprezentuje zapůjčení nebo nezapůjčení zařízení zaměstnancem. Každý zaměstnanec může pak být reprezentován bitovým vektorem. Tyto vektory mohou být analyzovány za účelem získávání asocičních pravidel.

$$\text{bruska} \Rightarrow \text{vrtačka} \quad (s = 2\%, c = 60\%)$$

kde s značí minimální podporu a c spolehlivost, což jsou metriky zajímavosti pravidla. Minimální podpora 2% zde znamená, že ve 2% transakcí byly tyto dvě položky společně. Spolehlivost 60% znamená, že 60% zaměstnanců, kteří zapůjčili brusku, si také zapůjčili i vrtačku. Asociační pravidlo je považováno za zajímavé, je-li splněna podmínka minimální podpory a minimální spolehlivosti. [2]

Získávání asociačních pravidel probíhá v následujících krocích:

1. Nalezení frekventovaných množin, tedy množin položek, které splňují podmínku minimální podpory.
2. Generování silných asociačních pravidel z frekventovaných množin. Tato pravidla musí splňovat podmínku minimální podpory a spolehlivosti.

4.3 Typy asociačních pravidel

Existují různé typy asociačních pravidel. Asociační pravidla mohou být klasifikována podle následujících kritérií:

- Podle typu hodnot v pravidlech: Jestliže nás zajímá pouze přítomnost nebo nepřítomnost položky, pak se jedná o booleovská asociační pravidla. Jestliže však pravidlo popisuje asociace mezi kvantitativními položkami nebo atributy, pak mluvíme o kvantitativních asociačních pravidlech. Příkladem může být následující pravidlo:

$$\text{věk} = 30..39 \wedge \text{příjem} = 40\,000 - 45\,000 \Rightarrow \text{koupí (LCD TV)}$$

- Podle dimenzí obsažených v pravidlech: Booleovská asociační pravidla jsou jedno dimenzionální, obsahují pouze dimenzi koupí. Naopak výše zmíněné asociační pravidlo je více dimenzionální, protože obsahuje celkem tři dimenze.
- Podle úrovně abstrakce v pravidlech: Některé metody jsou schopny získat asociační pravidla nad různými úrovněmi abstrakce. Takovým pravidlům říkáme víceúrovňová. Příkladem mohou být následující pravidla pro dvě různé úrovně abstrakce:

$$\text{věk} = 30..39 \Rightarrow \text{koupí (TV)}$$

$$\text{věk} = 30..39 \Rightarrow \text{koupí (LCD TV)}$$

- Podle dalších rozšíření asociačních pravidel: Dalšími rozšířeními může být korelační analýza, maximální vzory nebo uzavřené množiny.

4.4 Algoritmus Apriori

Algoritmus Apriori - jedná se o nejjednodušší variantu asociačních pravidel. Můžeme jej zařadit mezi dolování jednoúrovňových booleovských asociačních pravidel z transakčních dat.

Budeme-li brát v úvahu, že množinu obsahující právě k položek, nazveme ji k -množinou. Apriori je algoritmus pro získávání frekventovaných množin. Název vychází z faktu, že využívá předchozí znalost o dříve získaných frekventovaných množinách. V každé iteraci získané frekventované k -množiny jsou použity pro generování $(k+1)$ -množin. Každá iterace tak vyžaduje průchod databází.

Proces se skládá ze dvou kroků, spojovacího a vylučovacího:

- **Spojovací krok** - K nalezení L_k (všech frekventovaných k -množin), kandidáti na frekventované množiny (C_k) jsou vygenerovány spojením množiny z L_{k-1} . Necht' l_1 a l_2 jsou množiny L_{k-1} . Algoritmus předpokládá, že položky v množině jsou lexikograficky

seřazeny. Ke spojení dvou (k-1)-množin dojde, pokud jejich prvních k-2 prvků je shodných. Výsledná množina vzniklá spojením l_1 a l_2 bude mít prvky $l_1[1], l_1[2], \dots, l_1[k-1], l_2[k-1]$, kde $l_1[i]$ je i-tý prvek množiny l_1 a musí platit, že $l_1[k-1] < l_2[k-1]$.

- **Vylučovací krok** - C_k je nadmnožinou L_k , tzn., že její prvky buď jsou nebo nejsou frekventované. Zjišťování výskytu každého kandidáta v databázi je velmi neefektivní, proto využíváme Apriori vlastnost. Žádná (k-1)-množina, která není frekventovaná, nemůže být součástí frekventované k-množiny. Tedy, jestliže některá (k-1)-podmnožina není frekventovaná, pak není frekventovaná ani kandidátní k-množina a můžeme ji odstranit z C_k .

Algoritmus Apriori:

1. V první iteraci se prochází databáze pro každou položku a spočítá se její podpora. C_1 se tedy bude skládat ze všech položek databáze.
2. Ze vstupu se načte minimální podpora, kterou předpokládáme, že bude mít hodnotu 22% (počet výskytů bude 2). L_1 budou tvořit položky C_1 splňující minimální podporu.
3. K vygenerování C_2 dojde spojením množiny z L_1 dle výše popsaného postupu.
4. Proběhne výpočet podpory kandidátů z C_2 a jsou odstraněny nefrekventované položky na základě minimální podpory.

Dochází k cyklu, který generuje kandidáty pro C_{k+1} , při spojení množin z L_k . Položky **3** a **4** jsou generovány tak dlouho dokud $C_{k+1} = \emptyset$.

Ukázková tabulka transakční databáze je na obrázku 4.1 s vektorem zapůjčených položek. Jeden řádek této tabulky je tvořen v prvním sloupci zaměstnancem a jeho vektorem zapůjčených zařízení. Zaměstnance si v transakční databázi jsou označeni zástupným číslem ZXXX, kde XXX je pořadové číslo v tabulce. Vektor zapůjčených zařízení se rozumí seznam aktivit daného zaměstnance. První položka této tabulky by mohla vypadat následovně. Z001 zaměstnanec, který odpovídá dejme tomu v seznamu zaměstnanců v databázi s označením 1, tj. Ing. Pavel Lincer. Vektor zapůjčených položek tohoto zaměstnance je (I1, I2, I5), což znamená, že má zapůjčené nářadí s identifikačním číslem 1, 2 a 5 (I1, I2 a I5). To může představovat zařízení například 1. Pila, 2. Vrtačka a 5. Lopata. Při porovnání s posledním záznamem tabulky Z009, což představuje jiného zaměstnance, který má zapůjčené podobné položky jako zaměstnanec s označením Z001, tj. 1. Pila, 2. Vrtačka, ale třetí zapůjčenou položku má jinou a to dejme tomu 3. Kladivo.

Ukázka funkčnosti na obrázku 4.1 a pseudokódu pro algoritmus Apriori:

Vstupem: Transakční databáze D a minimální hodnota podpory min_supp

Výstup: L - frekventované množiny v D

1. $L_1 = \text{nalezeni_mnoziny}(D) \setminus \setminus \text{1-mnozina}$
2. **for** ($k = 2; L_{k-1} \neq \emptyset; k++$) {
3. $C_k = \text{apriori_gen}(L_{k-1}, \text{min_supp}) \setminus \setminus \text{generování nových kandidátů}$
4. **for each** transakce $t \in D$ { $\setminus \setminus$ projdi D pro zjištění počtu výskytů
5. $C_t = \text{subset}(C_k, t);$

Zaměstnanec	Vektor aktivit
Z001	I1, I2, I5
Z002	I2, I4
Z003	I2, I3
Z004	I1, I2, I4
Z005	I1, I3
Z006	I2, I3
Z007	I1, I3
Z008	I1, I2, I3, I5
Z009	I1, I2, I3

Tabulka 4.1: Transakční databáze D - zaměstnanci a vektor zapůjčených zařízení

```

6.      for each kandidát  $c \in C_t$ 
7.           $c.\text{počet\_výskytů}++$ ;
8.      }
9.       $L_k = \{c \in C_k | c.\text{počet\_výskytů} \leq \text{min\_supp}\}$ ;
10. }
11. return  $L = U_k \cdot L_k$ 

```

V prvním kroku jsou vygenerovány frekventované 1-množiny. Dále se v jednotlivých iteracích spouští funkce `Apriori_gen`, která provádí spojovací a vylučovací fázi, tedy generuje kandidáty. Poté se prochází databáze a pro každého kandidáta se spočítá počet jeho výskytů. Kandidáti, jejichž podpora je vyšší než minimální, jsou uloženi do L_k .

4.5 Generování asociačních pravidel z frekventovaných množin

Toto generování se provádí s využitím rovnice pro výpočet spolehlivosti:

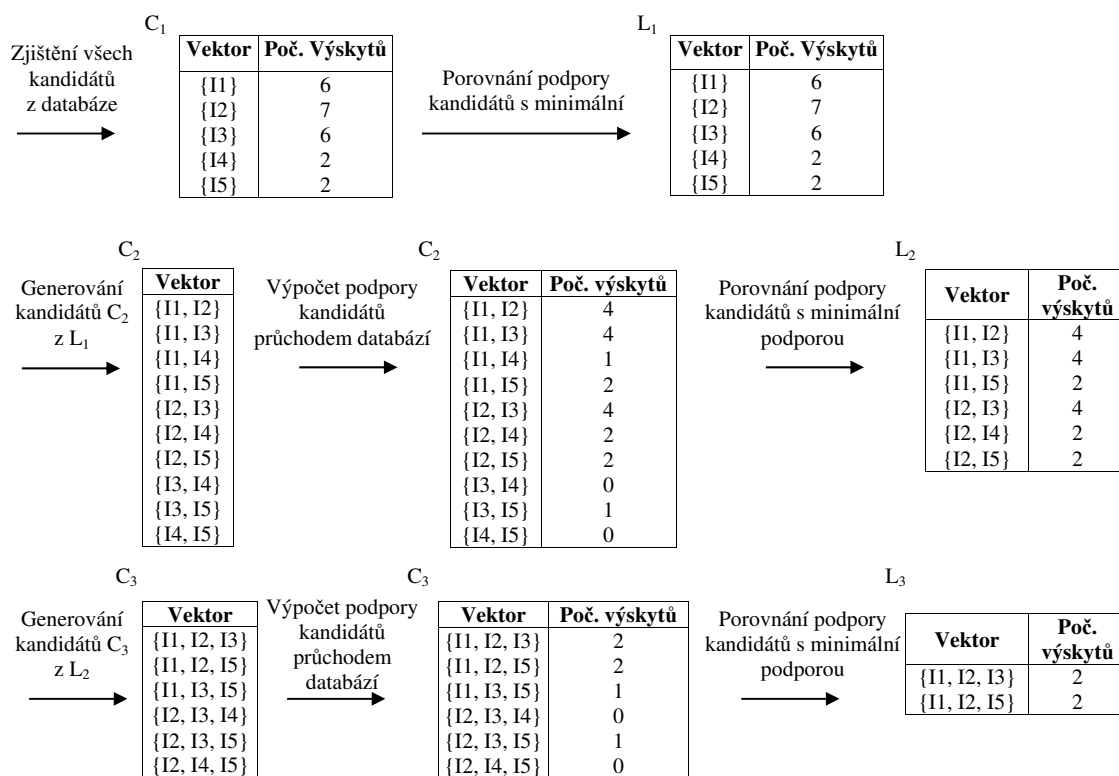
$$\text{conf}(A \Rightarrow B) = P(B|A) = \frac{s(A \cup B)}{s(A)}$$

Na základě této rovnice se postupuje při generování asociačních pravidel v následujících krocích:

- Pro každou frekventovanou množinu l , generuj všechny její neprázdné podmnožiny.
- Pro každou podmnožinu s , vygeneruj pravidlo $s \Rightarrow (l - s)$ a podle rovnice vypočti jeho spolehlivost. Pokud spolehlivost je vyšší než minimální, pak je pravidlo silné.

Protože jsou pravidla generována z frekventovaných množin, je pro ně automaticky splněna podmínka minimální podpory.

Příklad generování asociačních pravidel: Ukázka pro vygenerování asociačních pravidel pro množinu $\{I1, I2, I3\}$. Kde neprázdnými podmnožinami jsou: $\{I1, I2\}$, $\{I1, I5\}$, $\{I2, I5\}$, $\{I1\}$, $\{I2\}$, $\{I5\}$. Budou vygenerována tato asociační pravidla:



Obrázek 4.1: Funkčnost algoritmu Apriori

$I1 \wedge I2 \Rightarrow I5; c = 2/4 = 50\%$	$I1 \wedge I2 \Rightarrow I5; c = 4/2 = 100\%$
$I1 \wedge I5 \Rightarrow I2; c = 7/2 = 100\%$	$I1 \wedge I5 \Rightarrow I2; c = 2/7 = 29\%$
$I2 \wedge I5 \Rightarrow I1; c = 6/2 = 100\%$	$I2 \wedge I5 \Rightarrow I1; c = 2/6 = 33\%$

Je-li minimální hodnota spolehlivosti rovna 70%, pak výstupem budou tři silná pravidla. Pro minimální hodnotu spolehlivosti 50% budou tyto pravidla čtyři.

Kapitola 5

Návrh systému

5.1 UML

UML¹ je unifikovaný modelovací jazyk, který má vlastní grafickou syntaxi a sémantiku. Rozkládá systém na různé části a zároveň určuje vzájemné závislosti nebo vztahy s okolím. UML diagramů existuje několik typů lišících podle toho, jaké se pomocí nich plánují či zpracovávají úlohy. Tyto diagramy se od sebe odlišují především repertoárem použitých značek, způsobem jejich vzájemného propojení a s nimi související sémantikou.

UML jazyk definuje různé předměty (elementy), relace a diagramy. Tyto předměty mohou být například abstrakce modelu, mohou popisovat chování, seskupení položek nebo poznámky. K tomu aby tyto předměty mohly být propojeny, UML jazyk definuje relace. Tyto relace propojují různými způsoby různé předměty. Relace pak mohou mít charakter: Asociace, Závislost, Generalizace nebo Realizace. Diagramy zachycující různé pohledy na modelovací systém. Těchto typů diagramů je ohromné množství, například **Use Case diagram - diagram případů použití, Class diagram - diagram tříd, Diagramy objektů, diagram komponent, diagram aktivit...** [21]

V návrhu informačního systému jsem využil jazyka UML k popisu systému a to za použití Use Case diagramu a Class diagramu. Pro návrh relační databáze by se spíš hodil ER diagram než Class diagram, ve kterém sice nepřímě lze popsat entity, atributy a vztahy, ale například primární atributy už bez použití rozšíření vyjádřit nelze.

5.2 Program

Pravděpodobně nejznámějším a také nejpropracovanějším produktem, který pracuje s jazykem UML, je systém Rational Rose².

Pro tvorbu svých diagramů jsem ale využil jiný a méně známý produkt. Use case diagram a Class diagram jsem vytvořil v produktu Enterprise Architect (EA) verzi 6.5, který je sice zpoplatněný, ale existuje třicetidenní trial verze plného produktu s omezeným exportem finálního produktu. EA podporuje specifikaci UML verzi 2.0.

¹UML - Unified Modeling Language

²Rational Rose je vyvíjen firmou IBM Corporation produkt je dostupný na www-306.ibm.com/software/rational

5.3 Use case diagram

Use Case diagram, neboli případ užití, se využívá k zachycení funkcí, které systém nabízí svému okolí a definic využití těchto funkcí. Velmi často je využíván k prvnímu nahlédnutí do problému, slouží k rychlému zorientování a zhodnocení vstupů a výstupů ze systému. [1, 22]

Každý Use Case diagram je tvořen z následujících UML prvků:

- Use Cases - případy užití → zachycující funkčnost systému, stručné pojmenování funkce, např. zpracování dat, reakce na poruchu, apod.
- Actors - aktéři → aktéři pracující se systémem (vstupují do něj, používají jej), události mimo systém, např. obsluha, jevy zachycené čidly
- Collaboration - shluky speciálních funkcí, collaboration specifikuje role a atributy potřebné k dokončení speciálních funkcí
- Boundary - oblasti → definuje komponenty nebo podsystémy zahrnující Use cases, které jsou v tomto podsystému aplikovány
- Package - balíčky → obsahují seznamy jmen - tj. entit, které mohou obsahovat další balíčky
- Relationships (include/extend) - vztahy mezi jednotlivými Use Cases
- Associations - asociace → vazba mezi Aktérem a Use Case

5.3.1 Tvorba

Use case diagram obsahuje rozčlenění zaměstnanců do jednotlivých skupin podle jednotlivých oprávnění. Informační systém byl navržen pro firmu, která má rozčleněné zaměstnance do více úrovní. Jednotlivá postavení ve firmě sebou nesou určitou zodpovědnost a práva. Ukázka těchto práv pro šéfa zaměstnanců je na obrázku č. 5.1.

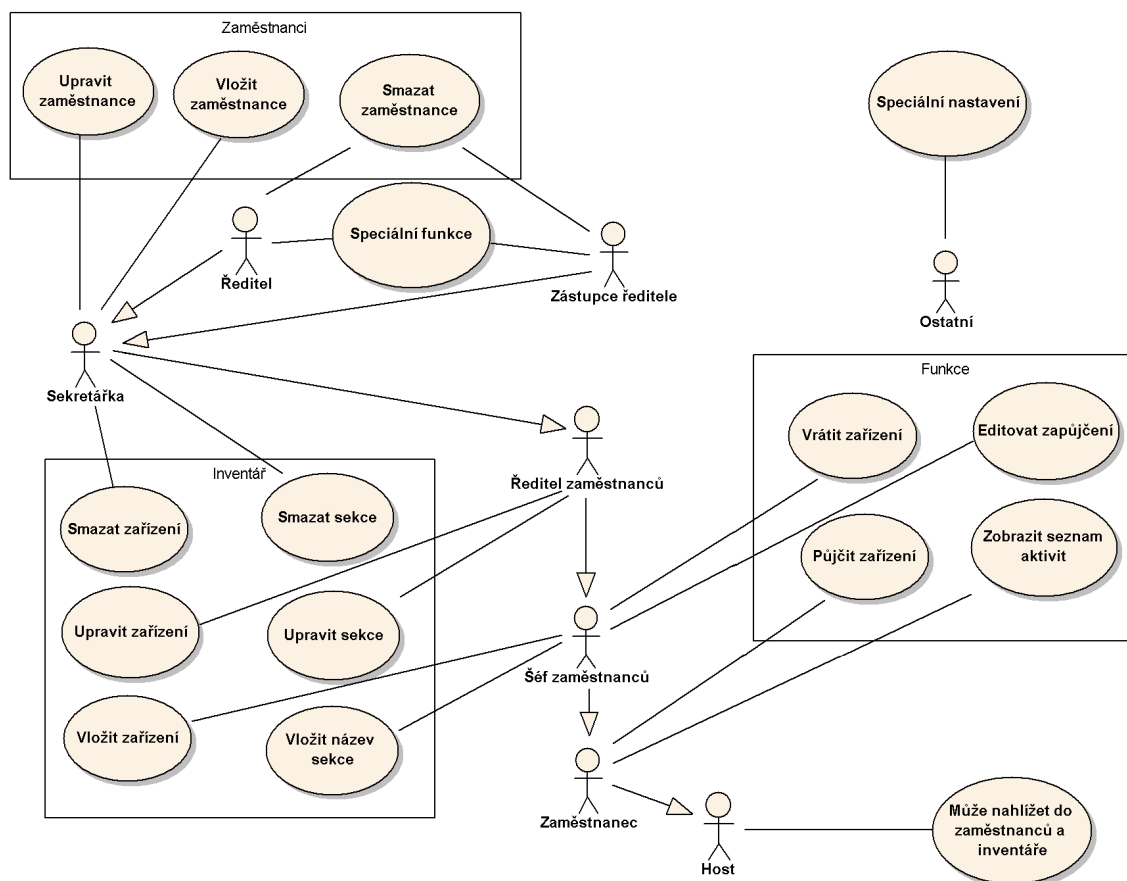
Vložit oprávnění			
<input type="radio"/> - ředitel	<input type="radio"/> - zástupce ředitele	<input type="radio"/> - sekretářka	<input type="radio"/> - vlastní
<input type="radio"/> - Ředitel zaměstnanců	<input type="radio"/> - Šéf zaměstnanců	<input type="radio"/> - zaměstnanec	<input type="radio"/> - host
Zaměstnanci	Inventář úpravy	Inventář	Funkce
<input type="checkbox"/> - vložit	<input checked="" type="checkbox"/> - vložit	<input checked="" type="checkbox"/> - vložit	<input checked="" type="checkbox"/> - půjčit
<input type="checkbox"/> - upravit	<input type="checkbox"/> - upravit	<input type="checkbox"/> - upravit	<input checked="" type="checkbox"/> - vrátit
<input type="checkbox"/> - smazat	<input type="checkbox"/> - smazat	<input type="checkbox"/> - smazat	<input checked="" type="checkbox"/> - zobrazit
			<input checked="" type="checkbox"/> - editovat

Obrázek 5.1: Ukázka oprávnění pro šéfa zaměstnanců

Zaměstnanci, kteří mohou obsluhovat tento systém jsou zobrazení na obrázku č. 5.2. Zaměstnanci na vyšších úrovních dědí vlastnosti od zaměstnanců na nižší úrovni:

- zaměstnanec - má právo pouze si zapůjčit zařízení z celkového seznamu a dále nahlédnout do zapůjčených zařízení tzv. aktivit

- šéf zaměstnanců - může upravovat délku zapůjčených zařízení v aktivitách, přijímat zpátky zapůjčené zařízení od zaměstnanců, může vkládat nová zařízení do databáze a také vkládat nové údaje do inventáře - jako jsou sklady, místnosti a typy zařízení
- ředitel zaměstnanců - může upravit již vložené údaje v inventáři a v databázi zařízení
- sekretářka - může vkládat zaměstnance, upravovat údaje od zaměstnanců, úplně vymazat věci z databáze nebo údaje v inventáři
- zástupce ředitele a ředitel - mohou smazat zaměstnance a mají plnou kontrolu nad správou aplikace a jsou jim přiřazeny speciální funkce v aplikaci
- host - speciální typ osoby, která nemá žádné výhradní právo pro správu aplikace, může pouze nahlížet do seznamu zaměstnanců a do databáze zařízení
- vlastní - speciální typ osoby, které jsou přidělena zvláštní práva do systému



Obrázek 5.2: UML use case diagram (model užití) uživatelů

Obyčejný zaměstnanec nemá právo smazat jiného zaměstnance z databáze (pokud mu toto právo ale nebylo přiděleno). Také jsem v diagramu zohlednil možnost „krádeže“ zařízení a to tak, že zaměstnanec má právo si zařízení zapůjčit, ale vrátit jej může pouze pod dohledem nadřízeného (šéfa zaměstnanců), tj. osoby, která má právo přijímat zapůjčené zboží do nářadovny.

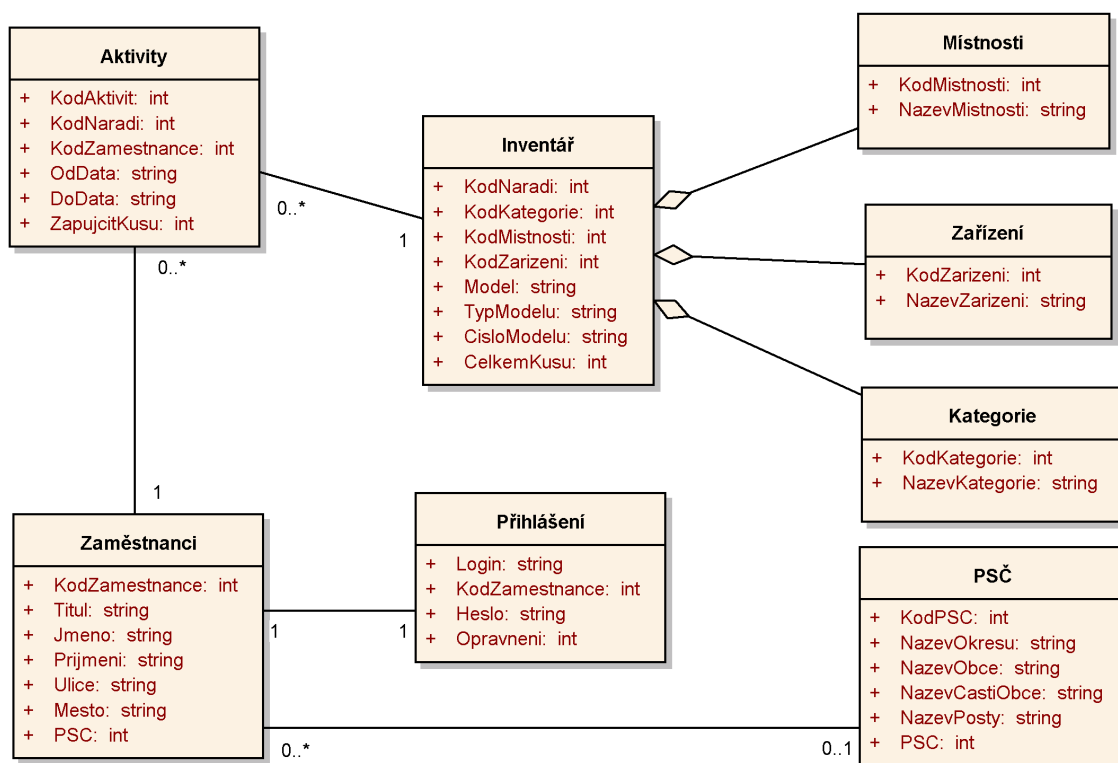
Typ „vlastní“ je speciální skupina zaměstnanců, která má jiná oprávnění, než která jsou předdefinována v systému. Tento případ zaměstnanců by se měl vyskytovat jen v ojedinělých případech, jelikož narušuje podstatu struktury oprávnění v systému.

Mezi další speciální skupiny zaměstnanců patří „ředitel“ a „zástupce ředitele“, kteří mají plná práva vstupu do systému. S tímto souvisí i speciální práva s používáním databáze jako je importování databáze poštovní směrovacích čísel (blíže v podkapitole 6.3) nebo generování asociačních pravidel z frekventovaných množin (o kterém pojednávala kapitola 4).

5.4 Class diagram

Class diagram - diagram tříd je graf symbolů tříd, rozhraní a dalších strukturních prvků propojených statickými vztahy jako je asociace, závislost, agregace, kompozice, generalizace, realizace. [22]

Class diagram informačního systému je na obrázku 5.3. Tento obrázek reprezentuje tabulky v databázi, jejich propojení a atributy jednotlivých entit.



Obrázek 5.3: Class diagram - diagram tříd

Class diagram se skládá z těchto prvků:

- asociace - je relace mezi dvěma či více třídami zakreslená úsečkou
- závislost - je relace mezi prvky, v níž změna jednoho elementu má vliv na závislý element, kreslíme ji přerušovanou šipkou
- agregace - vyjadřuje složení entity ze skupiny komponentních entit, vztah kreslíme úsečkou na konci zakončenou prázdným kosočtvercem

- kompozice - speciální případ agregace, kdy každá komponentní třída smí náležet pouze jednomu celku, vztah kreslíme úsečkou zakončenou plným kosočtvercem
- generalizace - je vztah mezi obecnější a specifitější entitou, vztah rodiče a potomka, vztah značíme plnou čarou zakončenou šipkou
- realizace - vztah mezi rozhraním a implementační třídou, který znázorníme čárkovanou šipkou

Existují různé typy návrhu. Metoda, kdy je návrh prováděn shora-dolů, která nejdříve identifikuje entity a pak jim přiřadí atributy nebo metoda návrhu zdola-nahoru nejdříve se najdou atributy a ty se přiřazují do entit, které v podstatě představují tabulky.

Implementace dnešních databází je většinou postavena na přístupu zvaném relační model. Relační model organizuje data do tabulek, tzv. relací, skládajících se z řádků a sloupců. Přestože relační teorie má své přímé principy návrhu databáze, používá se často z důvodu jednoduchosti návrhu ER model, který se následně do relačního modelu transformuje. [5]

Kapitola 6

Implementace

6.1 Funkce

6.1.1 Vložit

Při vkládání ať už zaměstnanců nebo zařízení do inventáře, musí být správně vloženy všechny požadované vstupní údaje. Po odeslání vstupních údajů nastává kontrola, zda byly tyto údaje správně vyplněny. Kontroluje se, zda je řetězec neprázdný, tj. zda došlo k nevyplnění určité položky. Kontrola také obsahuje test na dané položky, zda vkládaný údaj je číslo, zejména u vkládání zaměstnanců a to PSČ¹ nebo v inventáři při položce celkový počet kusů. Pokud nastala chyba na výstupu se zobrazí chybová hláška s oznámením, které položky byly nesprávně vyplněny a zpětně je zavolána funkce pro vkládání, která jako parametr obsahuje všechny vstupní položky pro vkládání tak, aby nedošlo k ztrátě vložených údajů. Výjimku tvoří vkládání zaměstnanců, kde oprávnění, jméno a heslo se zpátky nepředávají a musí se znova vyplnit. Je to v závislosti na tom, že vložené jméno a heslo může uživatel zapomenout a při opravě chybného údaje toto jméno a heslo zapomenout.

Při vkládání osobních údajů zaměstnance jako jsou město a PSČ informační systém nabízí možnost zkontrolovat tyto vložené údaje. Při zmáčknutí tlačítka „Zkontroluj město“ respektive „Zkontroluj PSČ“, se vygeneruje tabulka, která dané město respektive PSČ obsahuje. Uživatel si tak může zkontrolovat zda vložený údaj souhlasí s databází (podrobnosti v podkapitole 6.3). Při nezadání údajů je uživatel upozorněn aby vložil nějaký text pro kontrolu.

6.1.2 Tabulky

Tabulky jsou zobrazeny po 30 údajích. Tabulky obsahují sloupce s údaji a jeden funkcionální sloupec, který umožňuje s každým záznamem pracovat. Tabulky zaměstnanců a inventáře mohou být seřazeny.

6.1.3 Vyhledávání

Aplikace umožňuje vyhledat zaměstnance nebo zařízení dle vstupních údajů. Tyto údaje jsou nepovinné, takže není třeba vyplnit všechny položky. Při vyplňování je možné zadat i část obsaženého řetězce v databázi, není tedy nutné znát přesný a celý název hledaného

¹PSČ - poštovního směrovacího čísla

zaměstnance či zařízení v databázi. Po vygenerování se zobrazí tabulka s hledanými údaji. Tato tabulka obsahuje i funkcionální sloupec, který umožňuje ihned s vyhledanými daty pracovat.

6.1.4 Půjčit a vrátit zařízení

Zařízení lze půjčit po kliknutí na menu „Inventář“ a vybráním zařízení pro zapůjčení a kliknutí na odkaz „Půjčit“ nebo kdekoliv se tento odkaz u zařízení nachází. Následně je třeba vyplnit základní údaje o vlastnosti zapůjčení. Za prvé je třeba specifikovat „kdo“ si vybrané zařízení půjčuje. Pokud nemám právo minimálně vracet zařízení do skladu, tj. jsem obyčejný zaměstnanec, zařízení mohu zapůjčit pouze na sebe. Zde je ošetřena možnost, aby zaměstnanec nemohl samovolně zapůjčovat věci na někoho jiného. Pokud ale toto právo mám, tj. na úrovni šéfa zaměstnanců, mohu zadávat zapůjčené zboží i ostatním zaměstnancům. Dále je třeba upřesnit od kterého data a na jak dlouho bude zařízení danému uživateli zapůjčeno a následně poslední položkou je počet kusů. Posléze stačí kliknout na tlačítko „Zapůjčit“ a zařízení bude připsáno uživateli, který zařízení zapůjčil a z celkového inventáře ubude počet položek, které si zaměstnanec zabral. Pokud počet položek dosáhl nulové hodnoty zobrazí se hláška, že zařízení není na skladě a tlačítko „Zapůjčit“ není aktivní.

Vrátit zařízení lze taktéž více způsoby. Nejčastěji používaným způsobem bude vyhledat daného zaměstnance kliknutím na odkaz „Aktivity“ nebo pokud má uživatel právo na zobrazení svých aktivit, může zobrazit své aktivity rovnou z menu. Následně se zobrazí celkové aktivity daného uživatele. Poté stačí zatrhnout všechna zařízení, specifikovat počet kusů, které budou vráceny a kliknout na tlačítko vrátit. Provede se jedna ze dvou specifických funkcí, které zaručí buď částečné vrácení nebo úplné vrácení zařízení.

6.1.5 Smazat a editovat

Informační systém nabízí možnost smazat a editovat věci. Tuto funkci je možno využít jen v případě, že máte dostatečná práva pro smazání a editování položek. Odkaz pro smazání nebo editování položek se nachází v tabulce, konkrétně v pravém sloupci pojmenovaném „Funkce“. Pokud uživatel smaže zaměstnance, automaticky budou smazány všechny jeho aktivity. Při smazání zařízení z inventáře, jsou taktéž smazány všechny aktivity všech uživatelů, kteří měli toto zařízení zapůjčené. Pokud dojde k smazání údajů v inventáři, jako je kategorie, místnost nebo zařízení vzniknou nezařazené položky. Tyto položky pak budou v inventární tabulce zobrazeny úplně na konci. Zařadit zpět do systému mohou uživatelé, kteří disponují právem pro editaci položek v inventáři. Po kliknutí na editování dané položky v inventáři, je třeba zařadit záznam do příslušné kategorie, místnosti nebo zařízení. Záznam bude po uložení odstraněn z seznamu nezařazených položek.

6.2 Práva

Každý uživatel má přidělená práva, co smí a nesmí dělat (viz. ukázka obrázek č. 5.1). Tyto vstupní práva se pak převedou na třinácti bitové číslo, kde každý bit znamená určité oprávnění. Je-li příslušný bit v jedničce, uživatel bude mít přiděleno toto oprávnění. Pro změnu je-li v nule, toto oprávnění mu bude odebráno. Toto číslo pro uživatele jako je ředitel s plnými právy vypadá následovně „111111111111“, pro uživatele s nulovými právy „000000000000“ a pro uživatele s různými právy „100110110111“.

Práva jsou pak využívána při generování funkcí. Před každou funkcí, např. vložit zaměstnance, se nejprve zavolá funkce „kontrola“ s parametrem, který udává akci, která se má provést (v tomto případě „vložit“). Funkce provede bitový součin čísla pro vkládání s bitovým číslem uživatele. Vráti-li výsledek jedničku - tj. uživatel má právo vložit zaměstnance, funkce vrátí hodnotu **true** a následně zavolá funkci pro vložení zaměstnance. Pokud ale výsledek vrátí nulu - tj. uživatel nemá právo vložit zaměstnance, funkce vrátí hodnotu **false** a funkce pro vložení zaměstnance nebude zavolána a vypíše hlášku ve stylu: „Uživatel nemá právo pro vložení xyz akce“, kde xyz je název akce - „vložit“, kterou uživatel zvolil.

6.3 Import PSČ

Aplikace umožňuje importovat soubor s seznamem poštovních směrovacích čísel, soubor ve tvaru CSV². Seznam směrovacích čísel byl pořízen z volně dostupné databáze České pošty na stránkách http://www.cpost.cz/data/psc/xls_pcobc.zip ve formátu XLS³. Soubor byl následně upraven od nepotřebných sloupců, uspořádán do výsledné podoby a exportován do souboru CSV. Výsledné sloupce musí obsahovat tyto položky: Název okresu; Název obce; Název části obce; Název pošty; PSČ. Jednotlivé sloupce jsou odděleny středníkem.

Importovat a zároveň smazat předchozí databázi všech poštovních směrovacích čísel mohou ovládat pouze ti zaměstnanci, kteří mají plná práva jako je ředitel (viz. již zmíněná podkapitola 6.2). Tato speciální práva byla popisována v podkapitole 5.3.1.

Aplikace při kontrole poštovního směrovacího čísla využívá databázi, která je uložena na serveru. Tato databáze není primárně na serveru nahrána. Při instalaci aplikace je třeba při prvním spuštění tento import provést. Popřípadě až bude zapotřebí někdo z plnými právy provede import až za běhu aplikace. Do té doby nebude obsahovat databáze žádný záznam a kontrola PSČ respektive města bude zobrazovat hlášku o nenalezení daného slova v databázi.

6.4 Tiskové sestavy

Aplikace umožňuje tisknout pouze vybrané sloupce tabulky a těmto sloupcům určovat jejich velikost. Tato dynamická změna tabulky vede k zvětšení či zmenšení výsledného výstupu zobrazení ve formátu PDF.

K sestavení tabulky pro tiskovou sestavu je třeba vytvořit nový objekt. Tento objekt je typu: PDF_MySQL_Table. V této třídě je definovaná funkce „Table“, která jako parametr přebírá řetězec, tj. dotaz výběru nad tabulkou, pole s definicí hlavičky tabulky a pole s dynamickou šířkou sloupců. Tato funkce se postará o vytvoření příslušné tabulky. Dotaz výběru nad tabulkou je klasický MySQL dotaz nad tabulkou. Je-li použito v aplikaci seřazení podle určitého sloupce, výsledný PDF soubor toto seřazení bude akceptovat a rozšíří dotaz nad tabulkou o parametr seřazení. Jinak přiřadí jako standardní hodnotu pro zaměstnance seřazení podle „Příjmení“ a pro inventář seřazení dle „Modelu“.

Při vytvoření nového objektu typu PDF_MySQL_Table se musí definovat používané písmo, jejich přezdívký a text hlavičky ve výsledném souboru. Tato třída zajistí správné vysázení hlavní hlavičky v záhlaví a v zápatí číslování stránek výsledného souboru. Poté co je sestaven objekt se zavolá funkce „Table“ s příslušným parametrem. Tato funkce zajistí vložení sloupců hodnot pro tisk do pole. Dále zavolá funkci pro přepočtení šířky sloupců

²CSV - Comma-separated values, hodnoty oddělené čárkami

³Tabulkový formát pro operační systém Windows a Mac z kancelářského balíku Microsoft Office

(celé tabulky). Pokud byla překročena velikost stránky, tj. součet čísel pro dynamickou šířku sloupců je větší jak velikost šířky textu pro tisk, nedochází k přesnému definování sloupců, ale k poměru těchto hodnot tak, aby byla dodržena tabulka pro vysázení do velikosti šířky pro text. To má za následek přesného vysázení tabulky na stránku, ale k zdeformování vnitřních sloupců tabulky. Tato anomálie se dá vyřešit právě buď změnou velikosti šířky sloupců nebo odstraněním nepotřebného sloupce z výsledné tiskové sestavy. Funkce „Table“ zajistí vysázení hlavičky tabulky a jednotlivých řádků tabulky.

6.5 Struktura souborů

Informační systém se skládá z těchto souborů a adresářů:

1. index.php - ve kterém se nachází hlavní rozhodovací centrum, co se kde zobrazí
2. zamestnanci.php, inventar.php, inventar_upravit.php a aktivity.php - obsahují funkce, které jsou zapotřebí pro vkládání, editování a smazání
3. apriori.php - zpracovává data a vypočítává jejich ohodnocování a pravděpodobnost
4. fpdf.php a ufpdf.php spolu s adresářem Fonts - obsahuje třídu FPDF, respektive rozšíření třídy FPDF o UFPDF a potřebné soubory pro vygenerování souboru do formátu PDF jakou jsou písma
5. tisk.php - zpracovává parametry jako je počet sloupců pro tisk, nadpisy v tabulce, šířku jednotlivých sloupců pro tisk
6. tisk_class.php - zpracované parametry tiskne do jednotlivých buněk v tabulce, zde je funkce na přepočítání šířky tabulky, aby při tisku více sloupců, které jsou hodně široké nenastal případ, že buňky v tabulce by se zobrazily mimo okraj stránky a nemohly být řádně vytisknuty
7. include.php, connect.php, menu.php, kalendar.php, script.sql a style.css - soubory potřebné pro práci s aplikací

Kapitola 7

Zhodnocení

7.1 Podobné systémy

Různé modifikace aplikace firemní nářad'ovny jsou velmi častým produktem na trhu. Nejčastějším produktem jsou tzv. skladové systémy a ekonomické a účetní aplikace. Tyto aplikace jsou hlavně zaměřeny na daňové odpisy, vystavování faktur, dodacích listů, prodejek, účtenek... V těchto případech se nejedná o aplikace, které by se staraly o možnost zapůjčení zboží či jejich správu, i když některé aplikace tuto možnost nabízejí, jako spíš se jedná o možnost zjednodušení účetnictví a vedení zařízení v podniku.

7.2 Rozšíření informačního systému

Tento informační systém je pouhou malou aplikací. Při rozšiřování této aplikace by bylo dobré se zaměřit na myšlenku spojit tuto aplikaci s ekonomickým programem při zachování původní myšlenky informačního systému firemní nářad'ovny s možností zapůjčování zboží ve firmě. Systém by ještě navíc obsahoval položku cen. Kупní cena zařízení, jeho reálná cena, aby se vědělo jaké jmění má firma uloženu v hmotných statcích, dále stáří zařízení, jeho kategorie. Rozšířit počet položek pro identifikaci zařízení v informačním systému, tj. přidáním většího počtu položek pro popis jednoho záznamu v databázi zaměstnanců nebo inventáře.

Bezpečnost práce s přihlášenými uživateli. Běží-li tato aplikace jako webová aplikace je třeba zajistit větší bezpečnost. Příkladem by mohlo být šifrování, autentizace, využití hashovacích funkcí.

Jeho funkční rozšíření by mohlo být možnost rezervace zboží. Zaměstnanec by si mohl rezervovat zařízení dopředu, ale přiřazené by mu bylo později. Zde by bylo nutné vyřešit spor, kdy nezbude žádné zařízení na skladě. Jak přiřadit poslední zboží, když si zaměstnanec rezervoval zboží?

Využít funkce, která by řešila upozorňování vedení na zakoupení nového zboží, které je na skladě nejčastěji zapůjčované a často se nachází na pokraji nulové hodnoty. To by vedlo k lepšímu využití zapůjčených zboží.

7.3 Problémy při implementaci

Největší problém a možná i jediný velký při řešení bakalářské práce vznikl při sestavování tiskových sestav. Problémem bylo kódování. Je to zcela běžný problém se kterým se setkává-

me při vytváření jakýchkoliv aplikací snad všude. Chyba nastala při generování výsledného souboru PDF, kdy výstupní soubor vygeneroval místo spřežkových znaků dva „paznaky“. To bylo způsobeno zvoleným kódováním UTF-8. Chybu by se dalo odstranit změnou celého kódování ve všech souborech a v databázi nebo překódování každého výstupního znaku pomocí nějaké funkce do jiného kódování, ale já jsem se postavil k tomuto problému jinak a snažil jsem se aplikaci zprovoznit za jakoukoliv cenu v kódování UTF-8 a změnu na jiné kódování jsem bral jako krajní případ. To mě vedlo k rozšíření třídy FPDF o třídu UFPDF, která dokázala zpracovávat znaky v UTF-8 kódování. Výsledek však ještě nebyl zdaleka hotov, jelikož i písmo, které bylo zapotřebí pro vygenerování výsledného souboru neobsahovalo české znaky. Musel jsem si proto opatřit i konvertory písem pro třídu UFPDF, které byly odlišné od konvertorů písem FPDF.

Když byla třída rozšířena a písmo vygenerována, výsledek vypadal mnohem lépe. Ale znovu nebyl zcela dokonalý, jelikož nastal vizuální problém písmenka malé „č“, které se nesmírně podobalo velkému písmenku „Č“, přesněji řečeno písmenku malé „č“ bylo zobrazeno za použití písma arial.ttf jako velké C s tečkou - „Č“. Tuto anomálii se mi po dlouhé době studování podařilo také odstranit. Zkoušel jsem různé typy písma aplikovat do vstupního souboru v domněnku, že chyba je ve vstupním písmu. Výsledek byl znovu chybný, písmeno „č“ se mi nepodařilo změnit. Snažil jsem se pochopit v čem mohla vzniknout tato chyba.

Chyba byla způsobena programem, který generoval seznam znaků ze vstupního písma. Jednalo se tedy o program *tf2ufm.exe*. Tento program vždy přiřadil nesprávnou číselnou hodnotu k písmenu malé „č“ a špatně určil jeho velikost. Našel jsem na internetu seznam možných řešení, všechny postupy ale nevedly k správnému výsledku. Postup bude podobný, jak již byl naznačen v kapitole 3.5, s tím rozdílem, že dojde k menší úpravě výsledného vygenerovaného souboru. Tím dojde k správnému zobrazení písmena „č“. Řešení, které jsem použil a které vedlo ke správnému výsledku bylo následující:

1. První krok bude stejný. Vygenerovat tabulku znaků do souboru .ufm.
2. Nyní dojde k již zmíněné úpravě vygenerovaného kódu. Je třeba tento soubor editovat. Najít řádek, který bude obsahovat položku *Cdot*¹.
3. Je třeba změnit číselnou hodnotu pod označením G XXX, kde XXX představuje číslo, které bylo na tomto řádku nalezeno, na novou hodnotu G 254. Číslo G 254 označuje právě malé písmenku „č“. Tím se zaměnil špatný znak na správný.
4. Ještě je třeba upravit velikost vygenerovaného písmenka. Tato změna není nutná, pokud není potřeba znát přesnou velikost písmenka. Změna bude provedena na téže řádce, ale hodnotu WX XYZ, kde XYZ je číslo, změnit podle svého uvážení. Pokud špatný znak byl malý a správné písmenku by mělo být větší, je třeba tuto hodnotu zvětšit. V opačném případě zmenšit.
5. Pokračovat v generování souboru .php, jak již bylo zmíněno v kapitole 3.5.

Jelikož v aplikaci používám dva druhy písma, musel jsem provést změny hodnot v obou vygenerovaných tabulkách.

Ukázka změny pro písmo arial.ttf, kterou jsem provedl:

U 266 ; WX 722 ; N Cdot ; G 410 ; ⇒ U 266 ; WX 555 ; N Cdot ; G 254 ;

¹V textu jsou rozlišovaná malá a velká písmenka.

Ukázka změny pro písmo times.tff, kterou jsem provedl:

U 266 ; WX 667 ; N Cdot ; G 410 ; \Rightarrow U 266 ; WX 501 ; N Cdot ; G 254 ;

Jelikož třída UFPDF není zcela dokončena, může obsahovat nějaké chyby. Existují různé neoficiální verze, které opravují některé již známé chyby. Program pro vygenerování písma, jak jsem již zmínil, není zcela funkční a nemusí vždy vygenerovat správnou tabulku znaků z písma. Je tedy vždy nutné si pořádně zkontrolovat všechny výsledné vygenerované znaky, zda jsou správně zobrazeny.

Kapitola 8

Závěr

Cílem této bakalářské práce bylo vytvořit informační systém firemní nářaďovny. Tento systém jsem vytvořil a rozšířil ho o některé speciální funkce. Snažil jsem se uživateli nabídnout přehlednou, jednoduchou a nenáročnou aplikaci pro spravování zapůjčených věcí ve firmě. Systém je navržen tak, aby uživateli práci s aplikací co nejvíce zpříjemnil pomocí jednoduchého ovládaní.

Tento informační systém splňuje základní požadavky, které jsou zapotřebí pro vedení firemní nářaďovny. Dokáže vkládat zaměstnance, zařízení do inventáře a hlavně zapůjčovat a vracet zpátky dané nářadí. Obsahuje jednoduchou správu přihlášení pro zabezpečené přihlášení do systému a již zmíněné tiskové sestavy, generování asociačních pravidel nebo databázi poštovních směrovacích čísel.

Informační systém jsem po funkční stránce na závěr otestoval a programově prověřil pomocí W3C validátorů¹, který otestoval správnou syntaxi a sémantiku stránek. Tato kontrola proběhla úspěšně a proto jsem mohl vložit odkaz do levého hlavního panelu, že tyto stránky prošly kontrolou a jsou správně zapsané.

Touto bakalářskou prací jsem si rozšířil obzory v oblasti programování složitějších PHP stránek. Naučil jsem se používat jednoduché dolování dat z databáze.

¹Služba, která pomáhá zkontrolovat webový dokument více na domovských stránkách <http://validator.w3.org/>

Seznam použitých zdrojů

- [1] Uml use case diagrams:
<http://www.andrew.cmu.edu/course/90-754/umlucdfaq.html>. Poslední změna 30. 11. 2006. [cit. 2007-05-11].
- [2] Jaroslav Zendulka a kol. Získávání znalostí z databází. Technical report, VUT - Fakulta informačních technologií, Brno, Říjen 2006. Studijní opora do předmětu IDS.
- [3] Adaptic. Mysql. <http://www.adaptic.cz/znalosti/slovnicek/mysql.htm>. Poslední změna 2005. [cit. 2007-05-11].
- [4] Apache. Apache http server project. <http://httpd.apache.org/>. Poslední změna 2005. [cit. 2007-05-11].
- [5] Hana Kozelková. Editor er diagramů s podporou převodu do relačního modelu. Master's thesis, Univerzita Karlova, 2006. [online].
- [6] Ladislav Lhotka. Český překlad obecné veřejné licence gnu.
<http://staff.cesnet.cz/~lhotka/gnugpl-cz.html>, 2007. [cit. 2007-05-11].
- [7] FPDF library. Historie fpdf. <http://www.fpdf.org/en/histo.php>. Poslední změna 31. 12. 2004. [cit. 2007-05-11].
- [8] Osborne McGraw Hill. Javascript security.
<http://www.devarticles.com/c/a/JavaScript/JavaScript-Security/>. Poslední změna 4. 10. 2004. [cit. 2007-05-11].
- [9] Microsoft. Co je spyware?
<http://www.microsoft.com/cze/athome/security/spyware/spywarewhat.msp>. Poslední změna 2007. [cit. 2007-05-11].
- [10] Wikipedie otevřená encyklopedie. Apache http server.
<http://cs.wikipedia.org/wiki/Apache>. Poslední změna 17. 4. 2007. [cit. 2007-05-11].
- [11] Wikipedie otevřená encyklopedie. Cascading style sheets.
http://cs.wikipedia.org/wiki/Cascading_Style_Sheets. Poslední změna 29. 4. 2007. [cit. 2007-05-11].
- [12] Wikipedie otevřená encyklopedie. Html. <http://cs.wikipedia.org/wiki/HTML>. Poslední změna 18. 4. 2007. [cit. 2007-05-11].

- [13] Wikipedie otevřená encyklopedie. Javascript. <http://cs.wikipedia.org/wiki/Javascript>. Poslední změna 30. 4. 2007. [cit. 2007-05-11].
- [14] Wikipedie otevřená encyklopedie. Mysql. <http://cs.wikipedia.org/wiki/Mysql>. Poslední změna 17. 4. 2007. [cit. 2007-05-11].
- [15] Wikipedie otevřená encyklopedie. Opendocument. <http://cs.wikipedia.org/wiki/ODF>. Poslední změna 25. 2. 2007. [cit. 2007-05-11].
- [16] Wikipedie otevřená encyklopedie. Oracle. <http://cs.wikipedia.org/wiki/Oracle>. Poslední změna 1. 1. 2007. [cit. 2007-05-11].
- [17] Wikipedie otevřená encyklopedie. Pdf - portable document format. http://cs.wikipedia.org/wiki/Portable_Document_Format. Poslední změna 5. 5. 2007. [cit. 2007-05-11].
- [18] Wikipedie otevřená encyklopedie. Php. <http://cs.wikipedia.org/wiki/Php>. Poslední změna 2. 5. 2007. [cit. 2007-05-11].
- [19] Wikipedie otevřená encyklopedie. Postscript. <http://cs.wikipedia.org/wiki/Postscript>. Poslední změna 2. 5. 2007. [cit. 2007-05-11].
- [20] Artic Studio. Co je to databáze mysql? <http://www.artic-studio.net/slovnicek-pojmu/databaze-mysql/>. Poslední změna 2005. [cit. 2007-05-11].
- [21] Root.cz Pavel Tišnovský. Pdf - portable document format. <http://www.root.cz/clanky/nastroje-pro-tvorbu-uml-diagramu/>. Poslední změna 4. 7. 2005 [cit. 2007-05-11].
- [22] Dušan Kolář Zbyněk Křivka. Principy programovacích jazyků a objektově orientovaného programování. Technical report, VUT - Fakulta informačních technologií, Brno, Únor 2006. Studijní opora do předmětu IPP.

Seznam příloh

- A Instalace
- B Ukázka vzhledu pro přihlášení
- C Ukázka vzhledu tabulky
- D Obsah přiloženého CD

Příloha A

Instalace

K tomu abyste mohli zprovoznit tento informační systém, musíte nainstalovat pár základní věcí. Je nutné nainstalovat webový server Apache, databázový server MySQL a hypertextový preprocesor PHP. Snadnou a rychlou možností pro operační systém Windows Vám poslouží program EasyPHP verze 1.8. Velmi jednoduchá instalace, kde stačí spustit instalační soubor `easyphp1-8_setup.exe`, který je přiložený ve složce „Programy“ na odevzdaném CD. Nebo stáhnout z internetu nejnovější verzi ze stránek: <http://www.easyphp.org/>. Nainstalovat a spustit.

Nyní je třeba vytvořit databázi. K tomu využijete program PHPMyAdmin, kde po otevření vytvoříte novou databázi v UTF-8 kódování. Přepnete se do této databáze a kliknete na položku SQL. Zde máte dvě možnosti jak importovat databázi. Zaprvé otevřete si soubor `script.sql` uložený na finálním CD v složce „Aplikace“ a dvojhmatem Ctrl + C a Ctrl + V vložíte do tabulky pro SQL dotaz, vyberete kódování UTF-8 a zmáčknete „proved“ . Nebo zadruhé kliknete na procházet, najdete cestu k souboru `script.sql`, který je uložený na finálním CD, zmáčknete „OK“, vyberete kódování UTF-8 a zmáčknete „proved“ . Databáze by měla být úspěšně načtena.

Dalším krokem je zkopírovat aplikaci do složky webového serveru. V tomto případě to bude složka „www“ v programu EasyPHP. Do ní zkopírujete všechny soubory ze složky „Aplikace“ nacházející se na finálním CD.

Posledním krokem je nastavení údajů pro přihlášení k serveru MySQL. Otevřete zkopírovaný soubor `connect.php` v složce „www“, v tomto souboru je třeba změnit hodnoty proměnných `$DBuzivate1`, `$DBheslo`, `$DB` a `$DBpath` podle nastavení vašeho serveru. Pokud změníte i server na kterém běží aplikace, je třeba upravit hodnotu proměnné `$DBserver` dle adresy, na kterém databázový server poběží.

Poté stačí spustit jakýkoliv prohlížeč a zadat adresu <http://localhost> (popřípadě adresu serveru). Následně by se měla zobrazit úvodní stránka aplikace s přihlášením. Pro vstup do aplikace je nutné použít implicitní přihlašovací údaje.

Uživatelské jméno je „admin“ a heslo také „admin“. Nyní můžete aplikaci plně využívat.

Příloha B

Ukázka vzhledu pro přihlášení



Obrázek B.1: Ukázka informačního systému - přihlášení

Na obrázku **B.1** je zobrazen vzhled informačního systému, konkrétně úvodní stránka s přihlášením.

Příloha C

Ukázka vzhledu tabulky

Firemní nářadovna

HOME Mé aktivity Odhlásit

Hlavní menu

- Zaměstnanci
- Vložit
- Vyhledat
- PSČ
- Inventář
- Aktivity
- Apriori

Stav přihlášení

Přihlášen:

W3C HTML 4.01

ID	Titul	Jméno	příjmení	Ulice	Město	PSČ	Funkce
6		Bílý	Eda	3456	Albrechtice	38463	Edit Smazat Aktivita
10	Ing.	Blbý	Igor	Lesná 546/67	Brno	18374	Edit Smazat Aktivita
11		Hrozný	Miro	Ovesná 4567/7	Praha	73940	Edit Smazat Aktivita
1	Ing.	Lincer	Pavel	Ěp. 905	Stonava	76543	Edit Smazat Aktivita
3		Lincer	Karel	Krnov 87	Petřvald	73782	Edit Smazat Aktivita
2		Malý	Jan	Stavy 1023	Karviná	98724	Edit Smazat Aktivita
7		Matovský	Marek	Drmec 33	Havířov	83649	Edit Smazat Aktivita
4		Novák	Ondra	Malina čp. 456	Suchá	44682	Edit Smazat Aktivita
8	Ing.	Novotný	Michal	Hrnečná 146	Ostrava	76364	Edit Smazat Aktivita
12	Ing.	Svoboda	Krutý	Hrubá 80/8	Hradec	32748	Edit Smazat Aktivita
14		Široký	Kuba	Májová 67	Jablunkov	83654	Edit Smazat Aktivita
9		Škaredý	David	Oves 397/33	Tmava	84635	Edit Smazat Aktivita
13		Úzký	Hugo	Ráj 9/6	Louky	45625	Edit Smazat Aktivita
5		Zelený	Miloš	Krpavá 984	Petrovice	27593	Edit Smazat Aktivita

Vytvořit vlastní velikosti sloupců pro tisk?

9 15 18 27 45 33 15 Ne Ano

Počet sloupců: 7

E-mail: [xsvobo58\(zavinac\)stud.fit.cz](mailto:xsvobo58(zavinac)stud.fit.cz)
Web: <http://www.stud.fit.vutbr.cz/~xsvobo58/>

Obrázek C.1: Ukázka informačního systému - tabulky

Na obrázku C.1 jsou zobrazeni zaměstnanci s ukázkovými údaji.

Příloha D

Obsah přiloženého CD

Přiložené CD obsahuje zdrojové texty, programy, nástroje a výslednou aplikaci. Obsah CD je členěn do těchto adresářů:

- **Aplikace** - výsledný informační systém
- **Programy** - programy potřebné k instalaci, nástroje pro tvorbu písma nebo databázi PSČ ze dne 11.5. 2007
- **Text** - zdrojový text bakalářské práce a výsledná elektronická verze