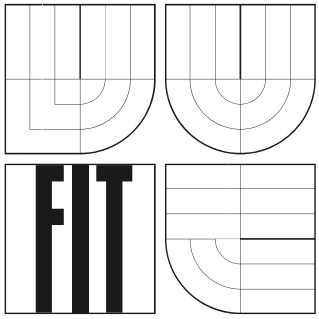
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Systém pro vedení daňové evidence

Semestrální projekt

2006 Bc. Jakub Dostal

Systém pro vedení daňové evidence

Prohlášení

Prohlašuji, že jsem tuto semestrální práci vypracoval samostatně pod vedením
Ing. Radka Burgeta, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jméno Příjmení
Datum

Abstrakt

Webový informační systém pro vedení daňové evidence s ohledem na českou legislativu, seznámení s problematikou daňové evidence, podmínkami pro korektní vedení příjmů a výdajů důležitého pro výpočet základu daně z příjmu. Součástí práce je předběžný návrh informačního systému s ohledem na možnosti a potřeby uživatele. Práce obsahuje navíc srovnání existujících produktů v dané oblasti právě z pohledu uživatele.

Klíčová slova

Java, J2EE, JSP, Servlet, Apache, Struts, Hibernate, MVC, Tomcat, MySQL, daně, evidence, účetnictví, legislativa, informační systém, objednávka, faktura, doklad, pokladna, banka, adresář

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Radkovi Burgetovi za jeho nápady a podněty.

Abstract

Keywords

Obsah

1 Úvod

Dnešní doba širokého rozmachu možností využití internetu je zdá se vhodnou příležitostí k vytváření informačních systémů různých zaměření a funkčností. Právě proto jsem se přiklonil k úvaze o realizaci informačního systému zaměřeného na pro mě ne zcela známou oblast. Pokusil jsem se propojit výhody internetového prostoru s nezbytnou součástí všech právních subjektů ve sféře podnikání na řekněme základnější úrovni. Jedná se o systém pro vedení daňové evidence, který si klade za cíl v co nejvetší míře zjednodušit a usnadnit zpracování veškerých náležitostí, které všem podnikajícím elementům ukládá zákon a to i s ohledem na aktuální českou legislativu. Je třeba si mimo jiné uvědomit, že oblast finanční sféry se neustále vyvíjí a pozměnuje, a proto je třeba již při návrhu zohledňit i možné budoucí legislativní úpravy. Na tuto problematiku je ve spoustě případů často zapomínáno a pak bývá aplikování daných problémů v již existujícím systému mnohem náročnější. Co ovšem je daňová evidence a co všechno musí daný systém pro její korektní plnění obsahovat, bude tématem následující kapitoly.

Nyní jen stručně, proč vlastně využívat pro danou problematiku možností internetových aplikací. Většina dostupných systémů pro vedení daňové evidence, tedy zjednodušenou formu účetnictví, předpokládá využití pro daný účel samostatných aplikací. Z nich nekteré využívají dostupné databázové technologie různých licencí či firem a jiné zase sází na vývoj a uplatnění technologií vlastních. Některé aplikace považují distribuovanou správu evidence za nadstandardní funkčnost a další ji, díky své technologii, vůbec neumožňují. Pokud ano, jedná se o složité klient-server aplikace, které svojí robustností převyšují nejen potřeby, ale často i finanční možnosti potenciálního uživatele. Proto čistě serverová internetová aplikace, která ke svému chodu na straně klienta potřebuje pouze standardní internetový prohlížeč, je alternativou jistě výhodnější po všech stránkách jak pro klienta, tak i samotného vývojáře. Je jisté, že daná technologie nabízí více možností, než využije systém pro vedení jednoduchého účetnictví, je však vynikajicím základem pro další možná rozšíření, které jsou nutností pro mnohem náročnější formy vedení finančních záležitostí. Na první pohled by mohla vyvstat otázka bezpečnosti, avšak například nemálo bank využívá danou technologii pro službu internetového bankovnictví svých klietů. Můžeme tedy s klidem potvrdit, že pokud nastane nějaká bezpečnostní chyba, je jisté, že nepochází z použité technologie, nýbrž selhání lidského faktoru s ní pracujícího. O použitých technologiích a jejich možnostech bude pojednáváno v některé z následujících kapitol.

V rámci semestrální práce se budeme zabývat nejen prvotním návrhem a plánem projektu výstavby konkrétní aplikace ale i analýzou již existujících systémů a jejich porovnání s ohledem na možnosti, jež nabízejí. Tato analýza budiž základem pro stanovení požadavků na vlastní systém, jež bude míti snahu rozvinout a využít kladných vlastností testovaných produktů a vyvarovat se zjevných důležitých nedostatků.

2 Daňová evidence

2.1 Úvodem

Jak již bylo v úvodu naznačeno, v této kapitole zmíním základní a obecné informace o daňové evidenci, co vůbec evidence daní obnáší a co z toho vyplývá pro samotný návrh. Součástí poslední podkapitoly bude i ukázka vedení evidence příjmů a výdajů v tabulkové podobě. Obrázek byl použit ze serveru euroekonom.cz, ze kterého byly ostatně částečně čerpány i doprovodné informace.

Jednotlivé konkrétní části evidence (kniha pohledávek a závazků, karty dlouhodobého majetku atp.) budou podrobněji popsány až v samotném návrhu podle vhodnosti uživatelského rozhraní. Touto částí by bylo zbytečné se zabývat znovu v této kapitole, neboť obecně teorie daňové evidence není hlavním tématem.

2.2 Definice

Pro bližší pochopení dané problematiky je nutné objasnit si základní pojmy. Daňová evidence (DE) obecně slouží ke stanovení základu daně z příjmu. Obsahuje veškeré údaje o příjmech a výdajích, dále pak údaje o majetku a závazcích. DE je ve své podstatě velice zjednodušenou a zredukovanou formou účetnictví. Svým způsobem navazuje na jednoduché účetnictví, které bylo v České republice zrušeno v roce 2004. Nyní zůstává pouze účetnictví, kterým se v Zákoně o účetnictví rozumí pouze podvojné účetnictví. Jednoduché účetníctví tedy zaniklo, ale fyzické osoby (FO), které chtějí vést evidenci na podobných principech jako bylo jednoduché účetnictví, mohou vést právě DE, a to podle § 7b Zákona o daních z příjmu.

2.3 Pro koho je určena

DE je určena pro ty subjekty, které nejsou účetní jednotkou (ÚJ) ve smyslu Zákona o účetnictví č. 563/91 Sb. Obecně je tedy určena pro všechny podnikající FO, které nepovedou podvojné účetníctví a své výdaje nebudou prokazovat „paušálně“, jinými slovy procentem z příjmů.

Daňovou evidenci v žádném případě nepovedou ty osoby, které jsou ÚJ. To se týká následujících osob:

- Právnické osoby (PO), které mají sídlo na území České republiky
- FO, které jsou jako podníkatelé zapsání v obchodním rejstříku
- FO, jejichž obrat přesáhl v předchozím kalendářním roce částku 15 mil. Kč

- FO, které vedou podvojné účetnictví dobrovolně

Daňovou evidenci tedy mohou bez problémů vést všichni drobní podnikatelé (FO), kteří nejsou zapsaní v obchodním rejstříku, s ročním obratem pod 15 mil. korun.

2.4 Jak by měla vypadat

O tom, jak má daňová evidence přesně vypadat, nehovoří žádný z paragrafů zákona o daních z příjmu. Potenciálnímu daňovému poplatníkovi dává jen zcela základní informace, o čem daňová evidence příjmů a výdajů vypovídá, jakým způsobem je oceněn majetek a odpovídající závazky. Dále se zákon zmiňuje o tom, že je potřeba provádět pokudmožno pravidelnou inventarizaci a archivaci. Záleží tedy na samotném poplatníkovi, jakou formu evidence zvolí. Podle zákona nejsou předepsány knihy závazků, ani knihy a kartymajetku, dále pak peněžní deník a pokladní kniha. Celou daňovou evidenci je, podle úrovně zjednodušení, možno vést třeba i v libovolném tabulkovém editoru. Je však třeba si velice dobře uvědomit, že se na základě vedené daňové evidence bude finančnímu úřadu dokazovat, že byly v následném daňovém přiznání uvedeny daňové příjmy a výdaje ve skutečné výši.

Nemálo důležité je taktéž si uvědomit, že DE musí splňovat náležitosti, které vyplývají ze zákona o daních z příjmu. Jinými slovy musí obsahovat informace o příjmech a výdajích, členěné pokudmožno způsobem vhodným pro snadné zjištění základu samotné daně. Dále pak informace o majetku a závazcích. Podle serveru euroekonom.cz [1] musí být evidence příjmů a výdajů (peněžní evidence) oddělena od evidence majetku a závazků. Příjmy daňové je taktéž třeba oddělovat od příjmů nedaňových a výdaje na zjištění a udržení příjmů („daňové výdaje“) je třeba oddělovat od výdajů neovlivňujících základ daně („nedaňové výdaje“).

2.5 Cíl

Hlavním cílem vedení daňové evidence je tedy (jak již bylo zmíněno dříve) žjištění základu daně. To je důležité pro vyplnění daňového přiznání a následnému odvedení daní finančnímu úřadu. Pokud je daňová evidence vedena přehledně a se značnou pečlivostí, poskytuje danému podnikateli přehled o stavu a pohybu majetku a dluhů (pohledávky a závazky), nemluvě pochopitelně o samotném výsledku jeho podnikání.

Výhodou určité volnosti oproti například podvojnému účetnictví je to, že si každý může evidenci jakkoli rozšířit, např. o rozdělení jednotlivých příjmů a výdajů podle různých kritérií dle libovůle poplatníka, neboť důležitý je až samotný výsledek.

Obr. 1 znázorňuje přehledné vedení příjmů a výdajů včetně dělení na daňové a nedaňové, o kterém jsem se zmiňoval v předchozí kapitole.

Datum	Doklad	Popis	Příjmy nedaňové	Výdaje nedaňové	Příjmy daňové	Výdaje daňové	Příjmy § 8
1.5.2005	PN1	Vklad na běžný účet	5 000,00				
10.5.2005	VD1	Nákup techniky				8 000,00	
15.5.2005	VD2	Reklamní materiály				4 000,00	
27.5.2005	PD1	Poradenská činnost			15 000,00		
2.6.2005	VD3	Zdravotní pojištění				1 000,00	
2.6.2005	VD4	Sociální pojištění				1 100,00	
5.6.2005	PD2	Instalace software			2 000,00		
21.6.2005	PD3	Tvorba prezentace			5 000,00		
2.7.2005	VD5	Zdravotní pojištění				1 000,00	
2.7.2005	VD6	Sociální pojištění				1 100,00	
13.7.2005	PD4	Instalace software			2 500,00		
31.7.2005	Výpis1 BÚ	Úroky na běžném účtu					50,00
31.7.2005	Výpis1 BÚ	Poplatky za běžný účet				200,00	
1.8.2005	Výpis2 BÚ	Osobní spotřeba (výběz z BÚ)		7 000,00			
Součty			5 000,00	7 000,00	24 500,00	16 400,00	**50,00**
			Základ daně z příjmu			**8 100,00**	

Obr. 1 Evidence příjmů a výdajů

3 Analýza existujících produktů

3.1 Úvodem

Produktů, které se zabývají více či méně podobnou tématikou, je celá řada. Vzhledem k dlouhodobým zkušenostem, které provázejí vývoj jednotlivých aplikací, je zřejmé, že snaha o modelování zcela nové aplikace by bylo pouze opakované „objevování Ameriky“. Zřejmě i proto zadání práce hovoří o prostudování již vytvořených produktů, neboť požadavky a jejich následná analýza bude z větší čáti ovlivněna právě výsledkem analýzy (porovnání a zhodnocení) existujících produktů. Pro tento účel jsem zvolil ty produkty, u kterých existuje možnost získání zkušební verze bez nutnosti jejich nákupu. Zároveň byla snaha vybrat produkty, jež jsou podle dostupných informací jedny z nejpoužívanějších ve spektru drobných podnikatelů a živnostníků.

U každého ze čtyř testovaných produktů bude zkoušeno vytvoření faktury (v jednoduchém účetnictví též daňový doklad) a její zaúčtování na základě pokladního dokladu. Případně i testování evidence ostaních dokladů. Prakticky všechny produkty mají společné základní vlastnosti, neboť je ukládá zákon a legislativa. Zmíním tedy pouze případné rozdíly v uživatelské části (ovládání, notifikace chyb a jejich oprava, atp.). Použitými technologiemi jednotlivých produktů není nutné se zabývat, na samotný návrh vliv nemají. Pochopitelně veškeré zmíněné klady a zápory jsou čistě subjektivního rázu a slouží pro usnadnění volby požadavků na samotný navrhovaný systém. Všechny aplikace jsou na tak vysoké úrovni, že konrola uživatele a jeho případných zásahů do evidence je samozřejmostí, není nutné se o ni dále zmiňovat. Množství a druh implementovaných modulů se příliž neliší. Jsou to:

- *Adresář* - není nutné pro samotné účetnictví, nezbytné však pro usnadnění práce
- *Daňová evidence (účetnictví)* – základ aplikace (peněžní deník)
- *Fakturace* – správa faktur přijatých a vydaných
- *Majetek* - správa majektu, hmotný, nehmotný a jejich odpisy
- *Sklady* – správa zásob, příjmových a výdejových dokladů
- *Mzdy a personalistika* – správa zaměstnanců

3.2 Produkty

3.2.1 Stormware Pohoda 2006

Hlavní jednotkou evidence je podle očekávání Peněžní deník. Zápisy v peněžním deníku vytváří Pohoda automaticky při zápisu a opravách dokladů v agendách Banka a Pokladna. Zaúčtování do

peněžního deníku se provádí dle vyplněné předkontace v záhlaví a položkách prvotního dokladu. Přímé zápisy do peněžního deníku se provádějí pouze u dokladů typu Převod a při opravách a rozúčtování jednotlivých řádků deníku.

Klady – zajímavé zpracování filtrace vypsaných položek, pěkně provedená daňová kalkulačka, přehled finančních prostředků, aplikace obecně velice rychlá a na ovládání přívětivá, právě používané agendy jsou přehledně k dispozici, v úvodu výpis ekonomických informací

Zápory – filtrace není příliž intuitivní, poněkud krkolomný výběr např. dodavatele/odběratele do různých dokladů, nepřehledný výpis a editace položek dokladů, zaúčtované faktury v peněžním deníku (uhrazovaný doklad) nejsou přehledě zobrazené, z toho vyplývá horší možnost jejich zětného dohledání, neukládájí se poslední otevřené agendy při ukončení práce s programem

3.2.2 Kastner software Stereo 2006

Záznamy o hotovostních platbách, výpisech z bankovních účtů i ostatní záznamy, které se mají objevit v peněžním deníku, se do peněžního deníku nezapisují přímo, ale pomocí agend prvotních dokladů (Pokladna, Bankovní výpisy, Interní doklady). Do samotného peněžního deníku se tyto doklady zaúčtovávají na základě nastavení údaje *Způsob zaúčtování*, v každé z jednotlivých agend.

V peněžním deníku je pak možno zaúčtované záznamy ze všech agend prohlížet nebo některý záznam rozúčtovat do více záznamů nebo jej změnit. V případě ručního zásahu do zaúčtování v peněžním deníku se nastaví v odpovídajícím dokladu prvotní evidence příznak, že byl záznam zaúčtování ručně změněn (údaj *účtováno* bude obsahovat hodnotu R).

Klady – volby možností jednotlivých atributů dokladů jsou k dispozici intuitivně a přehledně, včetně aplikování daného výběru zpět do dokladu, velkou výhodou je *Mapa programu* (velice intuitivní přehled možných operací přehledně uspořádán v chronologickém sledu dle použití)

Zápory – zbytečné zobrazení duplicitních informací v seznamu dokladů a současně zobrazovaném detailu zvoleného dokladu, v peněžním deníku není přehledně znázorněno, zda-li se jedná o příjem či výdej

3.2.3 ABRA G1

Velice komplexní systém pro vedení účetnictví pro dorbné podnikatele a živnostníky. Obsahuje nepřeberné množství různých doplňkových modulů pro vedení evidence všemožných dokladů. Peněžní deník (základ DE) se vykazuje přímo ze zdrojových dokladů za pomoci typů příjmů a výdajů. Systém jako takový je velice robustní, mnohonásobně převyšuje běžné požadavky na podobný systém, což je s jistotou jeho výhoda, ovšem svojí komplexností může odradit zákazníky, kteří vyžadují jednoduchost a přímočarost.

Klady – samotný peněžní deník je veden velice přehledně, neobsahuje nadbytečné informace o dílčích dokladech, modul evidence klasické pošty

Zápory – nemožnost snadného přepínání mezi jednotlivými agendami, díky rozsáhlosti celého systému je samotný jeho běh relativně pomalý.

3.2.4 MRP Daňová evidence

Oproti výše zmíněným je tento systém skutečně velice zjednodušený, ovšem v základní verzi. Je možné jej komponovat z množství doplňků, které mohou společně tvořit komplexní celek, který je možno přizpůsobit právě pro konkrétní potřeby potenciálních zákazníků. Zaúčtování se provádí, tak jako v předchozích případech, na základě přijaté či vydané faktury a přijatého resp. vydaného dokladu o pohybu peněz (pokladna, banka).

Klady – jednoduchost a variabilita systému, celkově přehledná a intuitivní správa potřebných dokladů.

Zápory – v základní verzi chybí zabezpečení a víceuživatelská úroveň přihlášení

4 Analýza problému

4.1 Neformální specifikace požadavků

Pro neformální specifikaci požadavků budeme vycházet ze samotného zadání projektu. To hovoří jednoduše o systému pro vedení daňové evidence. Což je samo o sobě relativně rozsáhlé téma předpokládáme-li, že systém nebude splňovat pouze základ daňové evidence (peněžní deník), ale evidenci i většiny dílčích dokladů, čímž se aplikace velice rozšíří. Jedním z obecných požadavků je s určitostí snaha navrhnout systém pro uživatele co nejpřívětivější. Měl by být komplexní ale současně si musí zachovat přehlednost. Proto by bylo nejvhodnější umožnit každému potenciálnímu uživateli zvolit si potřebné části dle svého uvážení a zaměření. Tomu odpovída nutnost navrhnout systém modulární. Konkrétní požadavky na celý informační systém by mohly být rozděleny do dvou skupin:

a) požadavky na daňovou evidencí

- Systém tedy musí obsahovat **evidenci příjmů a výdajů**, který se skládá z dokumentů dokazující fyzické příjmy a výdaje v podobě pokladních a bankovních dokladů.
- Pokladní a bankovní doklady jsou vystavovány na základě pohledávek a závazků. Ty jsou reprezentovány přijatými a vydanými fakturami. Jejich zpracování není sice v principu daňové evidence vyžadována, ovšem v případě kontroly vyslané státním institutem zvaným Finanční úřad, budou tyto doklady vyžadovány k předložení. Předpokládá se tedy jejich fyzické uskladnění. Evidence v systému následné vyhledání v mnohém usnadní. **Správa faktur** by měla být v systému obsažena.
- Faktury mohou být vydávány či přijímány dle objednávek. Jenomže **evidence objednávek** by pro relevantní informovanost vyžadovala mimo jiné i **správu skladových zásob**. Tyto moduly můžeme zařadít do seznamu potenciálních rozšíření daného systému.
- **Správu dlouhodobého majetku**, který má při výpočtu základu daně také svoji úlohu, některé systémy neumožňují. Ovšem zjišťování a vedení odpisů majetku je v pravdě velice složité a podléhá legislativním předpisům, proto by pro uživatele bylo vhodné, kdyby tyto operace systém mohl umožnit.
- **Správa bankovních účtů a pokladen** budiž taktéž přidána do seznamu budoucích rozšířeni.

- Nastavení **předkontace**, seznam druhů možných zaúčtovaných dokladů.
- **Personalistika a mzdová evidence** bude pro návrh zohledněna formou faktur přijatých od zaměstnanců.

b) požadavky na systém

- **Adresář kontaktů** pro vedení dodavatelů a odběratelů je sice doplňkovým (provozním) modulem, ale taktéž nezbytnou součástí sytémů podobného zaměření. Bylo by vhodné, ba takřka nutné, jej zahrnout do návrhu.
- **Správa firem.** Můžeme předpokládat, že si uživatel bude chtít vést daňovou evidenci pro všechny své firmy v jednom sytému. To samé platí pro v případě více živnostníků. K tomu poslouží modul správy firem (živností**)**
- **Přístupová práva** pro víceuživatelský režim jedné firmy.
- **Reporty, tiskové sestavy** bude možno tisknout na základě evidovaných dokladů. (případné rozšíření)

Je zřejmé, že zde nejsou uvedeny všechny možnoti, které by daný systém mohl obsahovat, prozatím pouze spíše ty základnější. Případné doplnění dalších by bylo předmětem navazujícího projektu.

4.2 Prvotní analýza požadavků

Jelikož by se dalo říct, že doklady vedené v peněžním deníku jsou posledním a nejdůležitějším článkem řetězce či dokonce stromu dokladů, musí být vedení samotných příjmů a výdajů navrženo s největší pečlivostí. V této kapitole se budeme zabývat jednotlivými oblastmi systému a uvedeme i atributy, které musejí být zpracovány. Nebudou zde však uvedeny všechny moduly zmíněné ve specifikaci požadavků, neboť některé přesahují rámec plánované realizace semestrálního projektu. Konkrétní a detailní analýza bude provedena až v jednotlivých iteracích návrhu a realizace projektu.

Evidence příjmů a výdajů

- Číslo dokladu a druh zaúčtování, případně variabilní symbol
- údaje o firmě od které resp. pro kterou je platba určena
- uhrazovaný doklad (faktura atp.)
- okamžik zaúčtování (datum, čas)
- rozepsaná částka (částka + dph) a částka celková
- definování doplňujícího textu
- kontrola korektnosti zadaných informací
- v závislosti na použitých modulech možnost přidávat a upravovat záznamy

Evidence pohledávek a závazků

- oddělené zpracování přijatých a vydaných faktur
- číslo faktury, variabilni symbol
- informace o dodavateli / odběrateli
- okamžik přijetí / vydání faktury
- druh faktruy v závislosti na předkontaci
- částka celkem (rozepsaná), výše zálohy, výše úhrady (i pro vícero měn)
- správa položek faktury
- typ obchodování (tuzemský / zahraniční)
- číslo souvisejícího dokladu (objednávka, příjemka, výdejka)
- kontrola korektnosti zadaných informací
- v závislosti na použitých modulech možnost přidávat a upravovat záznamy

Evidence dlouhodobého majetku

- kmenové údaje (název, inventární číslo, druh, členění, výrobní číslo, evid. Centrála, datum pořízení, druh původu, poznámka, pořizovací cena, zůstatková cena, atp.)
- údaje pro odpisy (způsob odpisu, odpisová skupina, uplatněný odpis)
- údaje - minulost (zařazení do evidence, atp.)
- cekový přehled cen
- kontrola korektnosti zadaných informací
- v závislosti na použitých modulech možnost přidávat a upravovat záznamy

Předkontace, typ účtování

- druh, název
- směr platby (příjem, výdej)
- typ DPH
- kontrola korektnosti zadaných informací

Evidence pokladních a bankovních dokladů

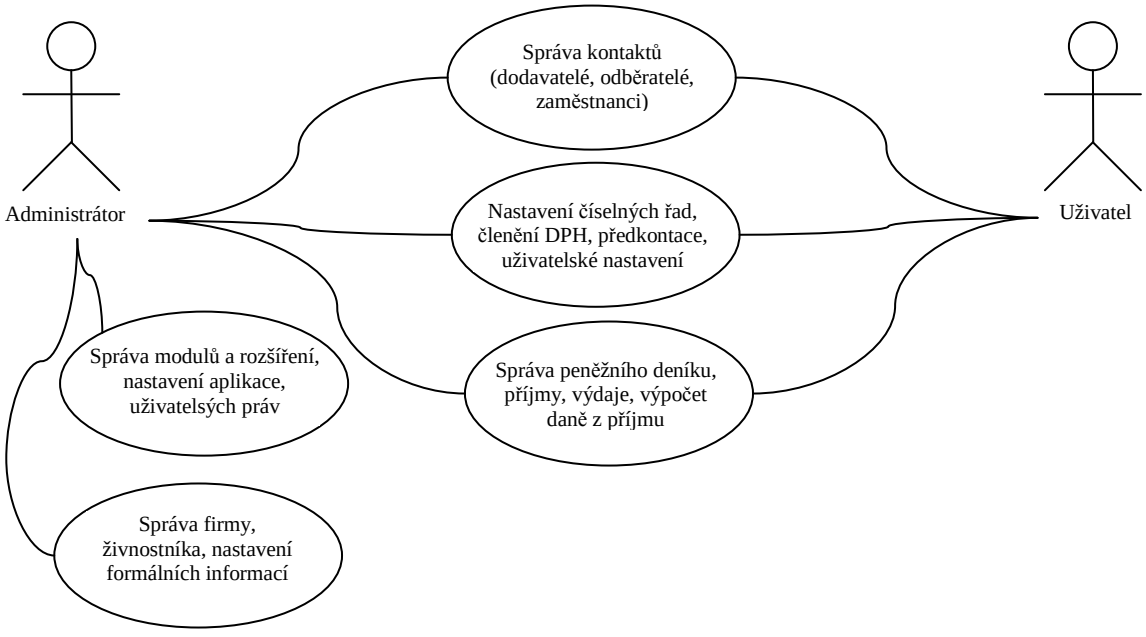
- typ dokladu (příjem, výdej, odchozí či příchozí platba)
- datum platby, vystavení
- druh účtového dokladu, předkontace
- doplňující informace
- nastavení dodavatele retp. Odběratele
- rozepsaná částka, včetně možností zobrazení v cizí měně

5 Návrh systému

5.1 I. iterace

Pojem samotné iterace zahrnuje několik fází, od specifikace požadavků, přes analýzu a návrh až po implementaci navrženého systému. Výsledkem každé z iterací je tedy funkční systém, který je v konečném důsledku možno nasadit u zákazníka. V kapitole Návrh systému se budeme zabývat návrhem jednotlivých iterací a to velice předběžným, neboť vstupem každého dalšího cyklu jsou požadavky ovlivněné výsledkem cyklů předchozích. A jelikož je implementace jednotlivých iterací naplánována na pokračování semestrálního projektu, není aktuálně k dispozici dostatek informací.

První cyklus, řekněme prvotní zpracování systému, bude věnován hlavní kostře aplikace. Kostrou je míněno rozhraní pro funkční komunikaci a běh všech potřebných modulů. Jak již bylo řečeno ve specifikaci požadavků, bude této funkčnosti věnována většina pozornosti, alespoň pro první ciklus vývoje systému, neboť se nám tímto usnadní jakékoli zpracování dalších rozšíření. Základem tedy budiž diagram případů použití pro první iteraci. Znázorněn na obrázku Obr. 3.



Obr. 3 Diagram případů použití, I. iterace

Pro názornou ukázku a vysvětlení bude vhodné rozdělit manipulaci se systémem do dvou různých pohledů, neboť v rámci první iterace se možnosti v mnohém rozcházejí. Jedná se o práci se systémem jako takovým.

Uživatel – dle jednotlivých oprávnění má uživatel možnost manipulovat s daty uloženými v peněžním deníku. Vkládá a upravuje doklady reprezentující příjmy a výdaje v podobě pokladních či bankovních dokladů. V průběhu celého zúčtovacího období si může nechat zobrazit výpočet předběžné daně z příjmu rozepsané na jednotlivé položky, jako je samotný základ, odčitatelné položky, slevy na daních atp.

Dále uživatel spravuje nastavení číselných řad určených pro jednotlivé dokumenty (zaměstnanci, přijaté a vydané doklady atp.), nastavení různých možností DPH podle typů spojených s aktuální legislativou. Dále může sparvovat předkontaci, což je definování různých druhů zaúčtování příjmů a výdajů. Uživatelským nastavením je míněna možnost předdefinovat si některé počáteční hodnoty různých atributů pro zrychlení práce s peněžním deníkem.

Může provádět určité úpravy související se správou kontaktů na úrovni odběratelů a dodavatelů.

Administrátor – má stejné možnosti jako obyčejný uživatel, rozšířšné např. o správu zaměstnanců v adresáři kontaktů a některá rozsáhlejší nastavení, ke kterým by obyčejný uživatel neměl mít přístup.

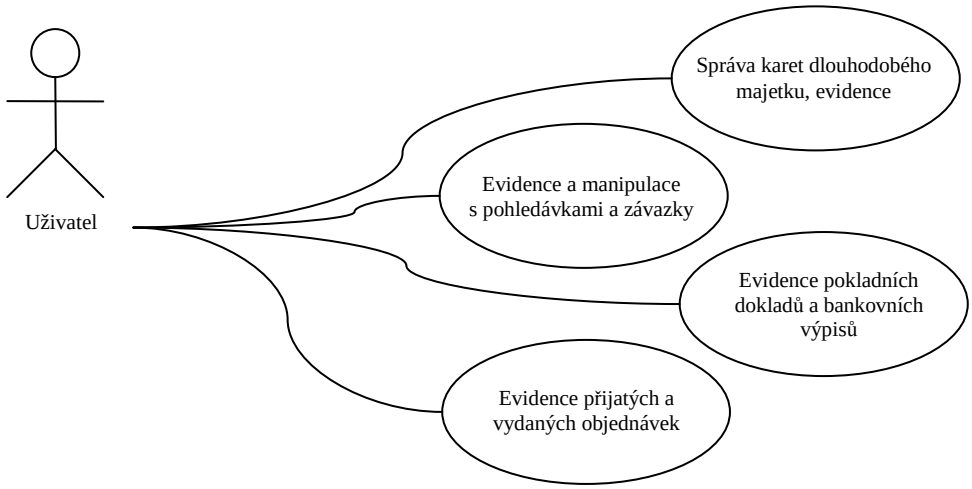
Mimo tyto běžné operace se správce systému zabýva samotným jeho nastavením, jako jsou například moduly, různá případná rozšíření. Hlavně také správou uživatelů včetně nastavování jejich uživatelských práv pro přístup do jednotlivých oblastí systému.

Upravuje informace o své firmy, vkládá údaje spojené se svojí živností. Provádí zálohování dat v podobě exportů a importů.

5.2 II. Iterace

V druhé iteraci budeme vycházet z již hotového systému, jež byl výsledkem předchozího cyklu. Můžeme tedy předpokládat, že byl systém realizován v podobě, v jaké byl definován a navržen v rámci iteraci první. Nyní se tedy otevírá prostor pro implementaci rozšíření o doplňující moduly a agendy určené pro správu dílčích dokladů, z nichž pak může vycházet samotný peněžní deník. Vycházíme z požadavků, které zatím nebyly realizovány Nyní se již budeme zabývat návrhem pouze z pohledu uživatele. Administrátor je jedním z uživatelů s absolutními přistupovými právy, má tedy v agendách možnost neomezených úprav. A jelikož se předpokládá jen minimální zásah do rozšíření kostry systému, není nutné explicitně definovat možnosti pouze samotného správce. Pro ukázku

možností uživatele tedy poslouží opět diagram případů použití, tentokrát bez rozdělení typů uživatelů. V následujícím popisu diagramu bude řečeno, co bude omezením v manipulaci pro uživatele v daném modulu. Diagram je znázorněn na obrázku Obr. 4.



Obr. 4 Diagram případů použití, II. iterace

Dlouhodobý majetek – evidence dlouhodobého majetku je relativně složitou záležitostí. Tuto možnost by měl mít jen uživatel s vyšší úrovní práv a administrátor. Součástí vedení evidence je i její detaliní rozpis včetne položek daného souboru.

Pohledávky a závazky – evidence pohledávek a závazků obnáší detailní vedení všech potřebných informací. Pro přehlednost budou faktury rozděleny do dvou agend, přijaté faktury a vydané faktury. Pro jejich manipulaci budou zvlášť definovatelná přístupová práva.

Pokladní doklady, bankovní výpisy – správa pokladen a bankovních účtů je součástí agend. Taktéž rozdělení přistupových oprávnění s ohledem na citlivost informací.

Objednávky – správa objednávek je doplňkovým modulem, realizován bude pouze okrajově. Ve své podstatě bude řešen podobně jako agendy faktur. Tedy pro přehlednost rozdělen do agend přijatých a vydaných objednávek

5.3 Možnosti rozšíření

Na tomto místě bych rád zmínil další možnosti rozšíření, které by mohly být realizovány v rámci celkové implementace systému. Jelikož z pochopitelných důvodu není možné s naprostou přesností předpovědět náročnost realizace zmíněných modulů, dá se předpokládat, že se mlže oběvit prostor pro některé další doplňující části.

Jednou z nich může být i detailní presonalistika, jež zahrnuje formální evidenci zaměstnanců, jejich pracovních poměrů, pojištění a nákladů na mzdy a pod.

Dalším by mohla být evidence firemních vozidel a kniha jízd pro evidenci dílčích spotřebních nákladů a detailů příslušného dlouhodobého hmotného majetku.

Analýza a návrh výše zmíněných modulů by tedy byl součástí realizace jedné z následujících iterací v rámci pokračování semestrálního projektu.

6 Implementační prostředky

6.1 Úvodem

Kapitola o implementačních prostředcích nechť pojednává o teoretickém základu technologií, které budou použity v rámci realizace systému. Celková implementace v obecném měřítku bude založena na platformě Java J2EE 5.0. Dle zadání má být použito webové architektury Java Servlet a skriptovacího jazyka JSP.

Proč použít právě tuto technologii pro realizaci internetové aplikace? Odpověď je zcela na snadě. Správná kombinace Servletů a JSP nám umožní vytvářet stránky jednoduchostí PHP a s možnostmi CGI. Přitom aplikace bude přenositelná a bezpečná. Ovšem hlavním důvodem je možnost použití celé webové aplikace jako vrstvy prezentační libovolné jiné aplikace. Je tím myšleno jednoduché propojení systému na platformě např. J2SE, které může bez problémů komunikovat právě se Servlety s využitím tzv. EJB komponent. To například při použití technologie PHP není ani zdaleka možné. Dále budou v této kapitole popsány Servlety a obecně i JSP, Apache Struts, což je framework pro vývoj webových aplikací, technologie hibernate pro převod objektů na strukturu relační databáze a konečně systém řízení báze dat MySQL.

6.2 Použité technologie

6.2.1 Apache Tomcat

Tomcat je oficiální referenční implementace technologií Java Servlet a Java Server Pages (JSP), obojí vyvíjené pod záštitou společnosti SUN. Ve své stabilní řadě 4.1 Tomcat nabízí podporu pro Java Servlet verze 2.3 a JSP verze 1.2. Řada 5, která se nyní dostává na úroveň stabilní verze, nabízí podporu Java Servlet verze 2.4 a JSP verze 2.0. Mocným nástrojem se však server Tomcat stává až ve spojení s klasickým http serverem. Z důvodu kompatibility a praktických zkušeností byl zvolen webový server Apache 2.0. Hybridní konfigurace serverů Apache a Tomcat se nasazuje tehdy, je-li na aplikaci kladeno veliké zatížení jak na statické, tak i na dynamické dokumenty. Apache 2.0 je velmi rychlý a flexibilní webový server a v kombinaci s kontejnerem Tomcat nabízí stabilní platformu pro webovou Javu. Pokud však aplikace neobsahuje příliš mnoho dotazů na neměnné HTML stránky a obrázky, je dobré přemýšlet o čistě javovské konfiguraci (tzn. pouze server Tomcat s konektorem Coyote).

6.2.2 Java Servlet a JSP

6.2.2.1 Servlet

Servlety jsou, zjednodušeně řečeno, programy napsané v jazyce Java. Jejich využití je hlavně v generování dynamických webových stránek, provádění oprací a akcí na straně serveru a v lepší a hlavně pohotovější interakci se samotným uživatelem. Instance servletu je aplikace běžící na serveru, jež má za úkol zprostředkovávat komunikaci mezi webovým prohlížečem resp. libovolným klientem zpracovávající protokol HTTP a webovým serverem. Aplikace zpravidla obsluhuje požadavky klienta na práci s databází. Servlety provádějí tyto operace:

- čtou data odeslaná uživatelem např. pomocí formuláře nebo Java apletu
- zpracovávají podrobnější informace o požadavku, např. cookies, informace o prohlížeči atp.
- zpracovávají požadavky, komunikují s databází a vytvářejí výsledky
- formátují výsledky, např. do XML či obecně HTML
- odesílají výsledná data v podobě dokumentu zpět uživateli

Co se samotného provozu servletů týče, na serveru stále běží JVM (Java Virtual Machine), který interpretuje kompilované servlety (binární kód instrukcí pro JVM). Každý servlet běží stále, dokonce i po dokončení odezvy, tím je zaručena schopnost uchovat hodnoty a data pro případ, kdy by bylo vhodné je mít dostupné bez opakovaného výpočtu. Servlety mají široké spektrum datových struktur a metod pro práci s http požadavky a odezvami, strukturami cookies, monitorování session a html formuláři.

Servlety moho mezi sebou sdílet data. To je v praxi používáno například pro sdílení připojení k databázi, předávání informací z požadavku na požadavek (vhodné např. pro sledování session specifické pro každého přihlášeného uživatele). Jsou prakticky přenosné. Díky jejich implementaci v Jave mohou bez větších problémů a úprav kódu fungovat na libovolné platformě či webovém serveru jež servlety podporuje. Jejich podpora může být zaručena přímo, nebo pomocí zásuvných modulů.

6.2.2.2 JSP

Servlety obsahují taktéž metody pro vytváření HTML do výstupního dokumentu. Pokud bychom ovšem ukládali statická data, kterých je v obecém dokumentu většina, stal by se zdrojový kód servletu krajně nepřehledným. Ne jen proto, ale také pro, alespoň částečné oddělení aplikační logiky od prezentační, bylo vyvinuto JSP (Java Server Pages). Ty nám umožňují skládat statické HTML s dynamicky generovaným obsahem. JSP se integruje do dokumentu v podobě specializovaných značek podobných HTML kódu. Ty vytvářejí a definují dynamickou část stránky. Fungují tak, že se při prvním zavolání dokumentu vytvoří servlet, ten se zkompiluje a zpustí. Dále se volá už jen ona

zkompilovná verze. Prvotní kompilace může o něco zpomalit odezvu serveru, což je ovšem zanedbatelné např. oproti PHP, kde se skripty kompilují a interpretují pro každé volání. Dynamická sekce stránek je psána v jazyce Java, jsou tedy programy v ní napsané použitelné jako komponenty programů jiných.

6.2.3 Apache Struts

Struts se snaží být co nejjednodušším webovým rozhraním. Nechává většinu práce a voleb na programátorovi. Pokud je tedy nutné vystavět webovou aplikaci „na zelené louce“, je Struts ideálním řešením. Skládá se minimálně ze tří částí - modelové (model), prezentační (view) a logické (controller). Prezentační vrstvu tvoří JSP dokumenty, které obsahujé speciální Struts značky. Další možností jsou případně servlety. Logickou vrstvu tvoří tzv. Action třídy a modely jsou komponenty JavaBeans. Všechny vrstvy dohromady spojí konfigurační soubor struts-config.xml, který dále přidává některé drobnosti navíc, jako je připojení do databáze s JDBC2. Zde je tedy patrná zkratka MVC (Model-View-Controller). Celkově se tedy jedná o rámec, který řídí tok aplikace, usnadňuje vícejazyčnou podporu, zpřehledňuje kód JSP dokumentů nebo provádí kontrolu vstupních údajů z webových formulářů.

Klíčovým prvkem frameworku Struts je výše zmíněné MVC (jinak také nazýván Model2), jež řídí kontrolní tok celé aplikace. Majoritní rozdíl mezi Modelem1 a Modelem2 je v tom, že každý z modelů zpracovává požadavky různými komponentami. V případě Modelu1 odpovídá za zpracování požadavků dokument JSP, který má taktéž na starosti zobrazení výstupu na straně klienta. Kdežto Model2 je tvořen třemi komponentami. Za prvotní zpracování požadavků zde odpovídá servlet (Controller). Ten po zpracování požadavků určí, který z JSP dokumentů bude dále volání obsluhovat a případně se zobrazí jako možná odezva. Komponenta Model odpovídá za správu vnitřní struktury. Tu reprezentuje komponenta View.

Nyní jen stručně o architektůře Struts. Úlohu Controlleru zvládá několik různých komponent. Definujme strom tříd *struts* pro zjednodušeni zápisu - *org.apache.struts.action*. Třída struts.ActionServlet umožňuje směrovat http požadavky k jednotlivým částem frameworku. Mezičlánkem spojujícím požadavek s aplikační logikou je třída struts.Action. V potomkovi je možné převzít a upravit metodu execute, jež je volána komponentou Controller na základě nějaké definované akce. Akce se defunují v konfiguračním souboru pro každou aplikaci zvlášť. Dále by bylo možné zmínit už jen třídu struts.RequestProcessor, která je řekněme odpovědná za analýzu a zpracování samotného požadavku. Další množství komponent nemá smysl v rámci této kapitoly popisovat, avšak dá se předpokládat, že se objeví v některé z kapitol popisující samotnou implementaci.

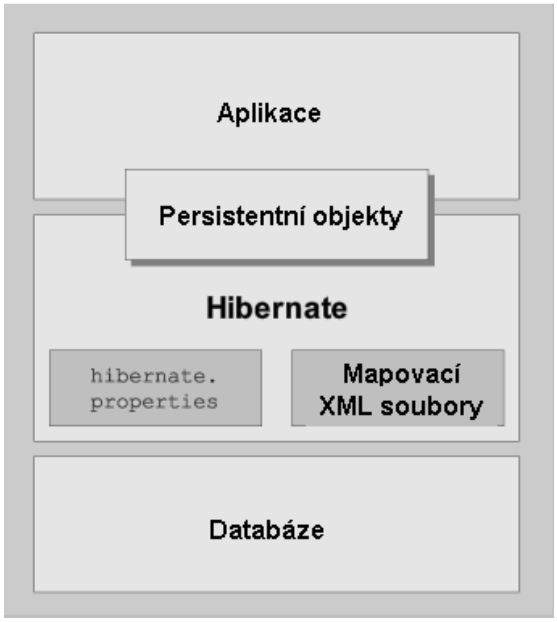
6.2.4 Hibernate

Zajímavý nástroj Hibernate se používá k zajištění perzistence dat. To ve skutečnosti znamená, že jsou data uchována i po ukončení běhu aplikace. To má několik vlastností. Jednou z nich je přístup současně několika uživatelů.k datům, tzn.že se jednotliví uživatelé vzájemně neblokují při čtení a manipulaci s daty. Další vlastností by bylo dobré zmínit transakční přístup. Právě ta umožňuje vícero uživatelům pracovat současně nad stejnými daty bez obaz z jejich porušení. Stará se o konzistenci dat, jež by mohla být narušena. Poslední z vlastností by mohla být přenositelnot a relativní snadnost použití. Relačním databázím přichází vhod podpora oběktově-relačního mapování.Ve své podstatě je to proces převodu (překladu) objektů na tabulkovou reprezentaci a převod vazeb mezi těmito objekty do tabulek dodatečných.Jak lze předpokládat, jedná se o nemnoho jednoduchý proces.

Vývoj perzistentních objektů s využitím hibernate v jazyce Java splňuje vlastností jako jsou kompozice, asociace, polymorfismus, dědičnost. Dále práce s kolekcemi objektů. Hibernate umožňuje získávání databázových dat na základě vlastního plně objektově orientovaného jazyka nazvaného Hibernate Query Language (HQL), který je syntakticky velmi podobný SQL.

Hibernate, jakožto objektově relační nástroj, je řízen konfiguračními soubory ve formátu XML. V těchto souborech jsou uvedena mapování aplikovaná na jednotlivé tabulky daného databázového schématu na základní třídy. Na základe těchto mapovacích souborů je Hibernate schopen zajistit persistenci objektu. Mapovací soubory lze ale využít i při opačném postupu. Jsou-li nejdříve vytvořeny základní objekty a k nim odpovídající mapovací XML soubory, je možné z těchto souborů vygenerovat odpovídající databázová schémata.

Samotnou architekturu hibernate nejlépe znázorní obr. 2 [2].



Obr. 2 Architektura Hibernate

6.2.5 MySQL

MySQL je databázový systém, vytvořený švédskou firmou MYSQL AB. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licencí GPL tak pod komerční placenou licencí. 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. Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích.. MySQL byla od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, triggery, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet.

Plnohodnotným databázovým systémem se všemi náležícími funkčnostmi se systém MySQL stal až od verze 5.x. S těmito vylepšeními ovšem poněkud ztratil svoji hlavní výhodu, a tou byla rychlost. Nová verze je proklamována jako plnohodnotná konkurence různým robustním komerčním systémům, avšak názory z praxe se poněkud rozcházejí. Záleží na účelnosti systémů, které MySQL využívají. Ovšem pro naše, v pravdě demonstrační, účely a podle nenáročností na operace, bude tento systém více než postačující.

Závěr

Cílem semestrálního projektu bylo seznámit se s teoretickým zázemím nutným pro úspěšnou realizaci webového informačního systému pro vedení daňové evidece. Součástí práce jsou kapitoly zabývající se obecnou osvětou problematiky daňové evidence v podobě vhodné snad i pro čtenáře, jemuž se nedostalo alespoň základního ekonomického vzdělání v dané oblasti. Další část projektu se zabývala analýzou již existujících komerčních produktů určených pro danou oblast. Přínosem bylo seznámení se s nutnými podmínkami, jež musí splňovat každá aplikace s ohledem na její zaměření a různými metodami řešení pro zpracování souvisejících dokumentů. Tato analýza pak sloužila jako výchozí bod pro stanovení požadavků a předběžný návrh. Samotný návrh systému je pojat z pohledu soustředěného na možnosti uživatele a jeho snadnou orientaci ve funkčnosti aplikace. Dalším krokem v návrhu bude rozšíření jednotlivých iterací o objektového rozhraní pro jejich imlementaci.. Závěrem práce je pak seznámení čtenáře s prostředky, jež budou dle předpokladů využity při implementaci celého informačního systému.

Literatura

[1] www.euroekonom.cz - Ekonomický portál

[2] Ing. Marek Běhálek: Hibernate framework.
Dokument dostupný na http://www.cs.vsb.cz/behalek/frvs/2005/java/hibernate/hibernate.html

[3] Lukáš Zapletal: Úvod do problematiky Apache Struts. (Květen 2003)
Dokument dostupný na URL http://www.root.cz/clanky/apache-struts-1-1

[4] MySQL AB: MySQL Manual. (Únor 2004)
Dokument dostupný na URL http://www.mysql.com/documentation.

[5] Bollinger, Gary, Natarajan, Bharathi: JSP – Java Server Pages.
Grada Publishing, Praha, 2003.

[6] James Turner: MySQL™ and JSP™ Web Applications: Data-Driven Programming Using Tomcat and MySQL, Sams Publishing, 2002.