

UNIVERZITA PARDUBICE

Fakulta elektroniky a informatiky

Ekonomicko správní systém pro malé  
a střední společnosti

Josef Brožek

Bakalářská práce

2010

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 12. 5. 2010

Josef Brožek

## **Anotace**

Tato práce se zabývá porovnáním ekonomických a ekonomicko správních software-ů, posouzení jejich obecné využitelnosti a vhodnosti pro různé typy společností.

Praktická část je zaměřena na návrh a tvorbu vlastního ekonomického systému pro malé a střední společnosti.

## **Klíčová slova**

Ekonomika, účetní software, účetní program, správní systém, porovnání, PHP, SQL.

## **Title**

Economicly-management system for small and medium-sized companies

## **Anotation**

Main target of this project is compare economy and economyc-management softwares and judge these uselles for companies.

Practical part is based on developt own economical system for small and middle companies.

## **Keywords**

Economy, bookkeeper software, management software, helpdesk, compare, PHP, SQL.

## Obsah

1 Úvod.....	10
2 Teoretická část.....	11
2.1 Cíle.....	11
2.2 Přehled výrazů.....	11
2.2.1 ARES.....	11
2.2.2 Programovací jazyk.....	11
2.2.3 Nadstavba programovacího jazyka.....	12
2.2.4 Skriptovací programovací jazyk.....	12
2.2.5 Databázový systém.....	12
2.2.6 Instalace SW.....	12
2.2.7 Lokální instalace.....	12
2.2.8 Multilokální instalace.....	13
2.2.9 Vzdálená instalace.....	13
2.2.10 Účetní SW.....	13
2.2.11 Systém pro správu skladu.....	13
2.2.12 Ekonomický SW.....	13
2.2.13 Administrativní systém.....	13
2.2.14 Správní systém.....	13
2.2.15 Administrativně správní systém.....	14
2.2.16 Licence.....	14
2.3 Rozdělení ESS.....	14
2.3.1 Dle specializace.....	14
2.3.2 Dle zákazníka.....	15
2.3.3 SAP systémy.....	15
2.3.4 Dle instalace.....	15
2.4 Využití ESS.....	15
2.4.1 Živnostníci.....	16
2.4.2 Malé společnosti.....	16
2.4.3 Výrobní společnosti.....	16
2.4.4 Společnosti poskytující služby.....	16
2.4.5 Společnosti obchodní.....	16
2.5 Nevýhody užívání ESS.....	17
2.5.1 Vnitropodniková bezpečnost.....	17
2.5.2 Zabezpečení dat.....	18
2.5.3 Metodika ochrany.....	18
2.6 Sběr dat pro porovnání ESS.....	18
2.6.1 Metoda výběru.....	19
2.6.2 Výběr SW.....	19
2.7 Porovnání vybraných ESS.....	20
2.7.1 Získání vzorků.....	20
2.7.2 Hodnocená kritéria a způsob hodnocení.....	21
2.7.3 Informace o produktech.....	21

2.7.4	Struktura hodnocení.....	22
2.7.5	Pohoda.....	22
2.7.6	Money S3.....	23
2.7.7	ADMWin.....	25
2.7.8	Altus Vario.....	26
2.7.9	Manas 2010.....	28
2.7.10	Pořadí hodnocených ESS.....	29
2.7.11	Závěry z testování jednotlivých ESS.....	30
3	Praktická část.....	31
3.1	Cíle.....	31
3.1.1	Systém vzdálené správy skladu.....	31
3.1.2	Pojmenování systému.....	31
3.2	Zadání pro tvorbu vlastního ESS.....	31
3.2.1	Způsob tvorby zadání.....	31
3.2.2	Zadání dle průzkumu.....	32
3.2.3	Zadání dle požadavků společnosti.....	32
3.2.4	Zadání od účetní společnosti.....	32
3.3	Příprava prací.....	33
3.3.1	Kontextová analýza.....	33
3.3.2	Analýza vztahu entit.....	34
3.3.3	Základní rozvržení prostředků.....	35
3.3.4	Zvolení databázového systému.....	35
3.3.5	Základní zabezpečení.....	35
3.3.6	Zabezpečení databáze na uživatelské úrovni.....	35
3.3.7	Uživatelské role.....	36
3.3.8	Ritch picture.....	36
3.3.9	Normalizace databáze.....	37
3.3.10	Uživatelské rozhraní.....	37
3.3.11	Vzhled rozhraní.....	38
3.4	Databáze.....	38
3.4.1	Návrhový SW.....	38
3.4.2	Normalizovaná databáze.....	38
3.4.3	Uživatelské role v databázi.....	39
3.4.4	Značení prvků v tabulkách.....	40
3.4.5	Popis jednotlivých tabulek.....	40
3.4.6	Vazby mezi tabulkami.....	44
3.5	Realizace uživatelského rozhraní.....	48
3.5.1	Požadavky na uživatelské rozhraní.....	48
3.5.2	Požadavky z datového hlediska.....	48
3.6	Metodika realizace.....	49
3.6.1	Připojení k databázi.....	49
3.6.2	Deoptimalizace pro vyhledávače.....	50
3.6.3	Soubory.....	50
3.7	Rozbor zabezpečení.....	50
3.7.1	Přístup uživatelů.....	50
3.7.2	SQL injection.....	50
3.7.3	Struktura databáze.....	51

3.7.4 Mazání prvků v databázi.....	51
3.7.5 Sledování manipulace s databází.....	51
3.7.6 Transakční režim.....	51
3.8 Ostatní funkce.....	51
3.8.1 Zadání počátečních stavů.....	51
3.8.2 Sledování stavů financí.....	52
3.9 Testování.....	52
3.9.1 Test zatížením.....	52
3.9.2 Test škozením.....	52
3.9.3 Finanční testy.....	52
3.9.4 Složité finanční testy.....	52
3.10 Zhodnocení projektu.....	53
3.10.1 Možnosti budoucího rozvoje.....	53
4 Závěr.....	54
5 Soupis bibliografických citací.....	55

## **Seznam zkratk**

ARES            Administrativní registr ekonomických subjektů

DPH            Daň z přidané hodnoty

ESS            Ekonomicko správní systém

MFČR           Ministerstvo financí České republiky

OS            Operační systém

SW            Software

## Seznam ilustrací

Obrázek 1: Struktura databázového systému.....	12
Obrázek 2: Nenormalizovaná databáze.....	33
Obrázek 3: Kontextový diagram.....	34
Obrázek 4: Entity relationship model.....	34
Obrázek 5: User case diagram.....	36
Obrázek 6: Ritch picture.....	37
Obrázek 7: Entity relationship diagram.....	39
Obrázek 8: Data flow diagram.....	49



## Seznam tabulek

Tabulka 1: Zastoupení ESS.....	20
Tabulka 2: Užití ESS ve firmách.....	20
Tabulka 3: Pohoda, hodnocení SW.....	23
Tabulka 4: Pohoda, hodnocení firmy.....	23
Tabulka 5: Pohoda, specifikace.....	23
Tabulka 6: Money, hodnocení SW.....	24
Tabulka 7: Money, hodnocení firmy.....	25
Tabulka 8: Money, specifikace.....	25
Tabulka 9: ADM, hodnocení SW.....	26
Tabulka 10: ADM, hodnocení firmy.....	26
Tabulka 11: ADM, specifikace.....	26
Tabulka 12: Altus, hodnocení SW.....	27
Tabulka 13: Altus, hodnocení firmy.....	27
Tabulka 14: Altus, specifikace.....	28
Tabulka 15: Manas, hodnocení SW.....	28
Tabulka 16: Manas, hodnocení firmy.....	29
Tabulka 17: Manas, specifikace.....	29
Tabulka 18: Celkove hodnoceni ESS.....	30
Tabulka 19: Ucty.....	40
Tabulka 20: Hlavni kniha.....	41
Tabulka 21: PomocnaKniha.....	41
Tabulka 22: DPH.....	42
Tabulka 23: PrvekZaverky.....	42
Tabulka 24: Zaverka.....	42
Tabulka 25: TypZaverky.....	43
Tabulka 26: Role.....	43
Tabulka 27: Users.....	43
Tabulka 28: Clovek.....	44
Tabulka 29: Plat.....	44
Tabulka 30: Vazba 1.....	45
Tabulka 31: Vazba 2.....	45
Tabulka 32: Vazba 3.....	45
Tabulka 33: Vazba 4.....	46
Tabulka 34: Vazba 5.....	46
Tabulka 35: Vazba 6.....	46
Tabulka 36: Vazba 7.....	47
Tabulka 37: Vazba 8.....	47
Tabulka 38: Vazba 9.....	47
Tabulka 39: Vazba 10.....	47
Tabulka 40: Vazba 11.....	48

# 1 Úvod

Cílem této práce je vytvoření efektivního ekonomicko-správního systému pro malé a střední společnosti. Jde o rozšíření původního účelu, kterým bylo vybudování systému na klíč pro společnost Pozitron technology k.s..

Pro zvládnutí celé problematiky, jak ze stránky programové, tak ze stránky účetní, bude třeba vybudovat kvalitní teoretický a později i analytický základ.

Zjištění, kde, kým a jak jsou ESS využívány, by mělo být prvním krokem k vytvoření úspěšného systému. Dalším faktem, který je vhodné zjistit, je riziko při využívání těchto systémů.

Kvalitnímu programovému řešení předchází analytické bádání, které by mělo určit směr budoucímu programu a umožnit neopakovat chyby, kterých se ostatní při tvorbě takovýchto druhů softwarů dopouští. Adekvátní by proto bylo otestovat několik již zavedených produktů z oblasti ekonomicko-správní a zjištěná pozitiva i negativa jednotlivých produktů pak zohlednit při tvorbě vlastního systému. Zejména v případě, že bude nalezeno společné negativum, bylo by jeho potlačením možné vystihnout mezeru na trhu, která by produktu umožnila expandovat mezi uživatele.

Stěžejní částí práce bude návrh vlastní databáze reprezentující datovou vrstvu aplikace. Vzhledem ke složitosti účetnictví a účetních operací je nezbytné věnovat výstavbě databázi systému maximální pozornost.

Pro zpřístupnění datové vrstvy pro uživatele bude vybudováno uživatelské rozhraní, které by mělo být intuitivní a mělo by poskytovat dostatečný uživatelský komfort pro snadné a příjemné zvládnutí veškerých operací, které by uživatel mohl chtít vykonat.

Nezbytnou součástí před dokončením samotné práce musí být testování na skutečných datech, aby byly odhaleny případné nesrovnalosti ve výstupních hodnotách programu.

Výsledkem práce by tedy mělo být shrnutí obecných informací o ESS, analýza jednotlivých produktů z uživatelského pohledu a vlastní systém uspokojující potřeby malých i středních společností.

## 2 Teoretická část

### 2.1 Cíle

Cílem této práce je definovat základní problematiku užívání ekonomicko správního softwaru ve společnostech, rozdělení ESS a analýza možného nasazení jednotlivých typů programů v různých prostředích.

V druhé části bude, dle jasných kritérií, vybráno několik systémů. Prostřednictvím určitých testů budou porovnány nejen tyto systémy, ale i jejich podpora a po rozhovoru s tvůrci i užití systémy při vývoji aplikace. Okrajově bude testováno i to, jak firmy svůj produkt prezentují, zda podávají přesné informace a samozřejmě cena produktu.

Na základech této teoretické práce bych rád, v případě, že bude nalezeno více nedostatků, vystavěl vlastní funkční systém pro malé a střední společnosti, avšak šířen pod licencí open-source.

### 2.2 Přehled výrazů

#### 2.2.1 ARES

"Administrativní registr ekonomických subjektů je informační systém, který umožňuje vyhledávání nad ekonomickými subjekty registrovanými v České republice. Zprostředkovává zobrazení údajů vedených v jednotlivých registrech státní správy, ze kterých čerpá data (tzv. zdrojové registry)."<sup>1</sup>

ARES poskytuje možnost získat na jednom místě všechny veřejné, státem vedené informace o všech ekonomických subjektech. Jako své zdrojové registry využívá např. databáze živnostenského úřadu, finančního úřadu, obchodního rejstříku atd. Systém je provozován ministrem financí České republiky a je dostupný na internetové adrese [www.info.mfcr.cz](http://www.info.mfcr.cz).

#### 2.2.2 Programovací jazyk

"Programovací jazyky jsou jazyky sloužící k tvorbě počítačových programů (programování). Programování je proces algoritmizace dané úlohy, tj. vytváření postupu, jenž vede k řešení dané úlohy."<sup>2</sup> V rámci práce budou uvažovány výhradně tzv. vyšší programovací jazyky, např. C/C++, Pascal, Java apod.

---

1 <http://www.info.mfcr.cz/ares/ares.html>

2 <http://k-prog.wz.cz/progjaz/index.php>

### 2.2.3 Nadstavba programovacího jazyka

Nadstavbou programovacího jazyka se rozumí jeho rozšíření. Rozšíření je zpravidla naprogramováno přímo v programovacím jazyce, kterému nadstavbu dělá.

Nadstavby jsou užívány pro snazší práci programátorů. Nejznámější nadstavbou je zřejmě prostředí Delphi pro Object Pascal, nebo modul Qt pro C++.

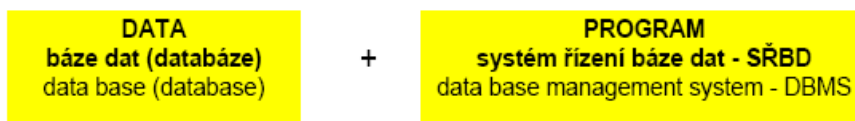
### 2.2.4 Skriptovací programovací jazyk

Skriptovací jazyk používá, stejně jako programovací jazyk, přesně stanovenou syntax příkazů. Na rozdíl od běžného programovacího jazyka jsou však skripty (zdrojové části programu) interpretovány tzv. interpretem, z čehož plyne, že skripty samotné nemusí být před spuštěním (interpretací) zkompileovány do binární podoby.

### 2.2.5 Databázový systém

**vnitřní struktura databázového systému**

každá databáze se skládá ze dvou částí:



**Obrázek 1: Struktura databázového systému**

"Soubor programových prostředků určených pro práci s daty, včetně organizace a realizace přístupu klientů k těmto datům." (Žák, 2009)

Jednotlivé části databázového systému jsou znázorněny na obrázku (Žák, 2009).

### 2.2.6 Instalace SW

Je takový proces, při kterém je SW upraven do stavu, ve kterém může být spuštěn se všemi funkcemi, které SW poskytuje.

### 2.2.7 Lokální instalace

Je taková instalace, kde SW stačí pro jeho práci lokální počítač. V případě, že se užívá databáze, je užíván standard převzatý z MS Access.

### **2.2.8 Multilokální instalace**

Je instalace na více fyzických počítačích v rámci jedné firmy či lokální sítě (mimo síť se lze připojit např. přes VPN). Data jsou ukládána dle myšlenky klient/server. Tj. veškerá data jsou uložena na jednom místě a ostatní k nim přistupují.

### **2.2.9 Vzdálená instalace**

Využívá čisté metody klient-server. Na serveru vždy musí být nainstalována aplikace včetně výpočetní části. Klient může přistupovat buď prostřednictvím speciálního klientkého SW, nebo dokonce přes internetový prohlížeč.

### **2.2.10 Účetní SW**

Je jednoduchý systém pro vydávání faktur. Výstupem jsou, kromě tištěných faktur, data srozumitelná pro účetní, která na jejich základě dokáže zpracovat účetnictví. Velmi často je účetní SW, v rámci reklamy, vydáván za ekonomický SW.

### **2.2.11 Systém pro správu skladu**

Systém umožňuje provádění základních evidenčních operací se skladem. Velmi snadno lze evidovat pohyby zboží, jeho množství, cenu, dobu, po kterou je zboží na skladě atp. Zpravidla bývá součástí ekonomického systému.

### **2.2.12 Ekonomický SW**

Je komplexní systém pro tvorbu podnikového účetnictví. Obsahuje možnosti tvorby faktur, účetních operací, operací se skladem, či práci se mzdovým účetnictvím. Výstupem jsou data pro příslušné státní instituce (daňová přiznání aj.).

### **2.2.13 Administrativní systém**

Administrativní systém je pojmenování pro celou škálu prostředků pro správu podniku. Administrativní systémy zahrnují nejen ESS, helpdesky, ale hlavně soustavu databází s uživatelským rozhraním, které dovolí přesně evidovat další aspekty činnosti firmy (ve firmě), jako jsou výkonnosti zaměstnanců, jejich nemocnost, atp.

### **2.2.14 Správní systém**

Systém pro řízení lidských zdrojů a koordinaci práce. Zjednodušení systému je velmi často využíváno pod názvem Helpdesk (řízení jen určitých zaměstnanců).

### **2.2.15 Administrativně správní systém**

Je spojení ekonomického a správního systému. Většinou kombinuje více typů instalace, např. klientské aplikace ekonomického systému v multilokální instalaci s objednávkovým a správním systémem v instalaci vzdálené.

### **2.2.16 Licence**

Licencí se rozumí právo na užití duševního vlastnictví (softwaru) druhé osoby. Jedna licence zpravidla vlastníka opravňuje k užití jedné kopie programu, přičemž může vlastníka licence i časově omezovat.

Licence upravuje zákon 121/2000 Sb., zkráceně Autorský zákon, konkrétní typy licencí pak vyhláška Ministerstva průmyslu a obchodu, která vychází z mezinárodních doporučení.

## **2.3 Rozdělení ESS**

Ekonomicko správní systémy je obecné pojmenování pro skupinu programů používaných ke snazšímu řízení a správě společnosti a k přípravě podkladů pro komunikaci se státními úřady.

### **2.3.1 Dle specializace**

Dnes se bez ESS neobejde téměř žádná instituce. Většina systémů pracuje provázaně, a proto je velmi těžké odlišit, kdy už jde o samostatný systém a kdy jde ještě o součást systému jiného. Pro účely této práce byl zvolen praktický způsob dělení, tj. pokud je možné nainstalovat systém samostatně tak, aby plnil svou funkci, jde o samostatný systém, pokud toto udělat nelze, jde o část jiného systému.

Dle specializace dělíme ESS takto:

- účetní software,
- ekonomický software,
- administrativní software,
- pokladní software,
- správní systém,
- systém správy skladu,
- helpdesk pro komunikaci se zákazníkem,

- administrační systém.

### **2.3.2 Dle zákazníka**

ESS mohou být jednoduché programy i komplexní řešení s nepřeberným množstvím funkcí. Základní rozdělení (které je výrazné zejména při pohledu na cenu produktů) tedy přibližně kopíruje možnost zákazníka do systému investovat.

Dle cílového zákazníka dělíme ESS takto:

- malé podniky které vedou daňovou evidenci,
- malé podniky, které vedou účetnictví,
- střední podniky,
- velké podniky,
- SAP systémy.

Tato práce je zaměřena jen na programy pro malé a střední společnosti.

### **2.3.3 SAP systémy**

SAP systémy jsou špatným pojmenováním ekonomicko správních softwarů pro velké ekonomické celky. Toto špatné pojmenování je způsobeno tím, že pro společnost SAP neexistuje na poli ADS korporátního rázu žádná přímá konkurence (společnost SAP sídlí v Německu, je čtvrtá největší SW firma na světě).

### **2.3.4 Dle instalace**

ESS je samozřejmě možné instalovat na jediný počítač. Avšak pro samotný účel ESS je tato instalace velmi často nevhodná. Přístup k systému z více míst je mnohem efektivnější a pro některé činnosti zcela nezbytný. Existují proto systémy s následujícími možnostmi instalace:

- lokální - převážně pro jednoduché účetní, či skladové systémy,
- multilokální - nejčastější instalace, nejčastěji vyžaduje nákup multilicence
- vzdálená - vzdáleně většinou jen části systému, řízení osob, objednávky, helpdesk atp.

## **2.4 Využití ESS**

Praktická využitelnost těchto systémů závisí nejen na tom, co všechno dovede samotný program, ale velmi významně také na znalostech člověka, který se systémem

pracuje. Funkce SW velmi často nejsou přístupné veřejnosti a také pracovník školený v oboru na jiném typu SW mívá s ovládnutím SW problémy.

#### **2.4.1 Živnostníci**

Přestože potenciál systému je opravdu výrazný, většina živnostníků využívá z celé škály ESS pouze pokladní systém, tedy systém, pomocí kterého vytvoří a vytisknou fakturu. Tyto vytvořené a vytištěné faktury na konci roku předají svému daňovému poradci, který má vlastní ESS na zcela jiném stupni složitosti, údaje do něj zadá a vše tak dělá za živnostníka.

#### **2.4.2 Malé společnosti**

V malých společnostech zpravidla nebývá zaměstnána účetní, která by se starala o veškeré účetnictví. Kvůli malému počtu zaměstnanců k nim má management velmi blízko, a tak není třeba ani nákup dalších softwarů.

Přesto je zde výrazně viditelný přínos ESS. Pomocí jedné či několika licencí lze velmi snadno evidovat a řídit veškeré finanční (občas jsou řízeny i materiální) toky. Systém zvládne vše od vyplácení mezd zaměstnancům po vypracování podkladů pro daňová přiznání a přehledu zisku a ztrát.

#### **2.4.3 Výrobní společnosti**

Výrobní společnosti se potýkají se složitější administrativou než ostatní společnosti. Pro jejich potřeby je nutné přesně evidovat materiál, výrobky a zdroje.

V tomto prostředí je často užíván složitější systém než v prostředích jiných. Už malé společnosti mají elektronicky vedený sklad, docházku zaměstnanců a toto vše je propojeno pomocí ekonomického SW.

#### **2.4.4 Společnosti poskytující služby**

Společnosti poskytující služby zpravidla využívají velmi jednoduchých forem ESS. Platí to nejen pro malé, ale velmi často také pro střední společnosti.

V tomto prostředí je jako postačující systém pro vydávání dokladů. Ostatní evidence jsou zpravidla zbytečné. Až střední společnosti užívají také ekonomický systém, který veškerá data zpracovává.

#### **2.4.5 Společnosti obchodní**

Obchodní společnosti jsou společnosti, které se zabývají nákupem a prodejem zboží, přičemž rozdíl v cenách nákupu a prodeje jim tvoří zisk. Toto je jednoduchá operace



nenáročná na administrativu či správu. Při větších obratech (at' finančních, či movitých) je však velmi výhodné užití ekonomického systému, který bude zajišťovat dohled nad správou skladu i finálními toky.

Společnosti tohoto typu velmi často požadují, aby jimi používaný ESS byl připojitelný na rozhraní elektronického bankovníctví a nezbytně také na vlastní elektronické obchody.

## **2.5 Nevýhody užívání ESS**

Protože do těchto systémů je vloženo mnoho informací o chodu firmy, jejím majetkovém a personálním stavu, jsou zde uložené informace velmi citlivé a jejich zneužití či poškození by mělo pro firmu trisťní následky.

### **2.5.1 Vnitropodniková bezpečnost**

Ekonomicko správní systémy musí být zabezpečny pro přístup uživatelským jménem a heslem. Je velmi žádoucí, aby každý uživatel měl nastavena uživatelská práva přesně odpovídající popisu práce. Naopak je velmi nežádoucí, aby měl kdokoli (třeba i administrátor) práva na všechno (tj. na úpravu databáze i manipulaci s daty).

Uvnitř podniku však nehrozí jen krádež dat, či jejich poškození, ale také velmi častý nešvar, který lze pojmenovat jako žárlivost na ostatní zaměstnance. Je způsoben neomezeným přístupem zaměstnanců i k tzv. citlivým informacím, přestože tato data ke své práci nepotřebují. Pak může nastat situace, kdy např. zaměstnanec zjistí, že spolupracovník na podobné pracovní pozici má vyšší plat, což vede k tomu, že pracovník s nižším platem odmítá vykonávat tolik práce jako dosud a klesá tak pracovní morálka nejen u něj, ale i u dalších zaměstnanců.

Za nejcitlivější je pochopitelně považováno nastavení uživatelských účtů pro mzdové účetní, které mají mít jako jediné ve firmě přehled o výši mzdy jednotlivých zaměstnanců. U středních a větších podniků je navíc doporučeno, aby mzdy pro nejvyšší management firmy zpracovávala externí společnost. Toto opatření zabrání ve společnosti mnohým sporům.

Nebezpečím pro tyto systémy jsou také zaměstnanci, kteří byli propuštěni. Velmi často jimi dochází k znehodnocení práce, kterou zde udělali. A pokud by například skladník smazal pohyby na skladu za poslední rok, podnik by tuto chybu mohl napravovat po řadu měsíců.

Pro ochranu je vhodné použít kombinace metod přístupu přes login, vhodného nastavení uživatelských rolí a systému evidence, který bude zaznamenávat každou operaci daných přihlášených osob. V případě nebezpečí vyvedení zboží ze skladu atp. je vhodné

přidat metodu párování příkazů; tj. pro vyvedení zboží ze skladu je nutné mít prodejní příkaz, či žádost od technika atp.

### **2.5.2 Zabezpečení dat**

Protože HW prostředky jsou napájeny elektřinou, a protože jsou poruchové, může dojít k situaci, kdy nebudou data dostupná. Řešení pomocí zálohy serverů není pro malé a střední podnikatele finančně únosné, proto se musí na nějaký čas bez dat obejít.

Problém však nastává s udržení jejich konzistence. Při výpadku proudu může dojít k jejich potrhání atp. Může také nastat problém s udržení dat v případě chyby pevných disků (většinou se řeší spojením do polí RAID). Kromě zmíněných problémů může dojít k úmyslnému i neúmyslnému poškození člověkem, virové nákaze, či k fyzické krádeži počítače. Proto je vhodné správně aplikovat zálohování dat, a to fyzicky mimo počítač, ze kterého jsou data zálohována.

### **2.5.3 Metodika ochrany**

Zde jsou shrnuty vhodné postupy pro uchránění dat nejen před ztrátou z důvodů poruchy HW, ale i před ztrátou údajů způsobené tzv. lidským faktorem:

- zálohování,
- párování příkazů,
- přístup přes uživatelské jméno a heslo,
- uživatelské role,
- evidence činnosti.

## **2.6 Sběr dat pro porovnání ESS**

Pro zjištění, zda je trh nasycen kvalitními systémy, byla zvolena metoda srovnání několika produktů.

Porovnání může pomoci určit slabé i silné stránky jednotlivých SW, čímž může určit cestu pro implementaci vlastního systému.

V obecné obchodní strategii není vhodné konkurovat zavedeným značkám jejich kopírováním, ale nabídnutím větší přidané hodnoty. V tomto případě (open source licence) bez nutnosti finančních investic.

### **2.6.1 Metoda výběru**

Protože určení, kdy se jednotlivé systémy liší, je velmi obtížné, byla přijata metoda sronávání dle výrobce, nikoli dle typu SW. Jedna společnost často vyvíjí několik typových řad svého produktu, kde každá řada je určena jinému segmentu trhu. Situace je o to nepřehlednější, že každá samostatná řada je téměř vždy tvořena jednotlivými balíčky, ze kterých se ESS skládá.

V České republice je dostupno přibližně 16 ESS od rozličných výrobců. Porovnání takového množství produktů není vhodné, proto byl výběr zúžen.

Výběr byl proveden prostřednictvím emailového dotazování živnostníků a malých společností v Pardubickém kraji. Otázky jsou uvedeny níže a byly, kromě pozdravu, žádosti a poděkování, jedinou součástí emailu.

- Otázka č.1: Používáte ve Vaší firmě ekonomický či správní systém? Popřípadě jaký?
- Otázka č.2: Jsou veškeré výsledky zpracovávány unitř firmy, nebo je výstup z programů zpracováván druhou osobou?

### **2.6.2 Výběr SW**

Výběr porovnávaných SW byl proveden na základě emailového dotazníku. Tento email byl rozeslán ve 120 kopiích, celkem se vrátilo 74 odpovědí. Výsledky uvedené v tabulce jasně znázorňují orientaci místních podnikatelů na ekonomický systém Pohoda s tím, že ostatní balíčky nejsou téměř používány. Protože z výzkumu vyšlo využívání práce pěti systémů, budou tyto systémy následně porovnávány. Bylo několikrát zmíněno, že firma užívá ESS, kterým je vybaven jejich eshop. Protože jde jen o funkci eshopu „vydávání faktur“, nejsou tyto hlasy brány v potaz.

**Tabulka 1: Zastoupení ESS**

Systém	Četnost	
	Absolutní	Relativní
Pohoda	37	56,1%
Money S3	16	24,2%
Altus	10	15,2%
ADMWin	2	3,0%
Manas 2010	1	1,5%
Celkem	66	100,0%

Z šetření dále vyplynulo, že podniky využívají převážně správní systémy – v čele s ekonomickým systémem a skladovým systémem. Správní systémy jsou využívány až ve středních podnicích, z nichž byla návratnost emailů nejnižší. Je možné prohlásit, že živnostníci a malé společnosti většinou využívají jen možnosti elektronické fakturace, případně správy skladu a výsledky a zpracování si nechávají vytvořit od daňových poradců. Až střední podniky mají vlastního zaměstnance specializovaného na ESS. Tyto podniky také využívají širší škálu systémů a to od systémů účetních a skladových, až po systémy pro řízení lidských a materiálních zdrojů.

**Tabulka 2: Užití ESS ve firmách**

Možnost	Četnost	
	Absolutní	Relativní
Účetnictví si pracujeme sami	21	31,8%
Vytváříme doklady, zpracovává daňová poradkyně	43	65,2%
Zpracovává naše účetní a pak daňová poradkyně	2	3,0%
Celkem	66	100,0%

## 2.7 Porovnání vybraných ESS

### 2.7.1 Získání vzorků

Veškeré ESS vybrané pro testování nabízí možnost stáhnutí omezené verze zdarma (pozn. u jednoho SW není umožněno stáhnutí, ale je možné si objednat omezenou verzi zdarma na CD). Veškeré informace, kde byl SW získán jsou vždy uvedeny u konkrétního porovnávaného vzorku.

## 2.7.2 Hodnocená kritéria a způsob hodnocení

Hodnocení je prováděno stupnicí "jako ve škole", tedy bodovým hodnocením od jedné do pěti, přičemž jednička znamená nejlepší výsledek a pětka výsledek nejhroší. Pro přesnější srovnání je možno dávat mezistupně, tj 1,5; 2,5 atp.

Hodnocení probíhá ve dvou vlnách. Zprv je hodnocen samostatný SW, zadruhé je hodnocen SW jako obchodní produkt. U hodnocení samostatného SW jsou tyto kategorie:

- funkce,
- vzhled SW,
- přehlednost a umístění ovládacích prvků,
- jak se SW ovládá znalému člověku,
- jak se SW ovládá laikovi.

Při hodnocení SW jako obchodního produktu bylo využito toho, že byly společnosti několikrát kontaktovány. Jednou telefonicky, čistě obchodně. A dvakrát emailem, jednou obchodně a jednou s dotazem na daný SW (viz další kapitola).

Hodnoceny tedy byly:

- www stránky,
- telefonická komunikace,
- elektronická komunikace,
- a neobchodní komunikace.

## 2.7.3 Informace o produktech

Pro potřeby této práce je nezbytné znát nejen popředí jednotlivých systémů, ale také podpůrné systémy, které jsou jednotlivými systémy používány, programovací jazyky, které byly použity pro vývoj a další data o systémech. Proto bylo využito elektronické pošty a společnostem vyvíjejícím dané produkty bylo zasláno několik dotazů, konkrétně:

- Od kdy je systém na trhu?
- Kolik lidí se podílí na vývoji a jak dlouho je systém vyvíjen?
- Jde o společnost lokální, nebo nadnárodní?
- Kolik (řádově) licencí je užíváno na našem trhu?

- V jakém programovacím jazyce je SW vytvořen?
- Používá systém databáze? Pokud ano, jaké?

Odpovědi na jednotlivé otázky jsou uvedeny vždy u každého konkrétního produktu.

#### **2.7.4 Struktura hodnocení**

Každý SW byl hodnocen samostatně, až po otestování všech softwarů. V článku je vždy popis společnosti i SW a následně jeho hodnocení v tabulce, včetně slovního komentáře.

#### **2.7.5 Pohoda**

Systém Pohoda je dílem české společnosti Stormware, která vznikla v roce 2006. V téže roce uvedla ekonomický systém Pohoda, který se nyní může chlubit několika desítkami tisíc prodaných licencí.

Společnost vlastní webové stránky, které nijak neurazí. Jde o jednoduché stránky, které plní svůj účel. Zaměstnanci společnosti jsou ochotní a zdvořilí. Kromě obchodních záležitostí jsou ochotni hovořit o svém produktu a také poskytnout zákazníkům velmi kvalitní podporu.

Nová verze softwaru je uvolňována několikrát ročně. Společnost se navíc chlubí tím, že na programu neustále pracuje přibližně 30 programátorů a mnoho odborníků z dalších oborů.

Systém je v omezené verzi ke stažení přímo ze stránek společnosti Stormware na <http://www.stormware.cz/ke-stazeni/>. Program je plně funkční, avšak pouze po dobu 3 měsíců.

Základní verze systému, která umožňuje jen vydávání faktur je dostupná již za 2.000 Kč. Tato verze je velmi přehledná i pro laiky, a to je zřejmě důvod, proč jde o nejpoužívanější software na českém trhu. Dražší mutace pak stojí až 8.000 Kč pro daňovou evidenci a od 8.000 Kč do 14.000 Kč lze pořídit software umožňující zpracovávat účetnictví.

Složitější programy jsou dobře funkčně vybaveny, ale pro laiky se jeví mírně nepřehledné. Celkové hodnocení tak tomuto programu snižuje nejvíce vizuální stránka - software není zajímavý a občas přehlčuje uživatele informacemi.

**Tabulka 3: Pohoda, hodnocení SW**

Hodnocená oblast	Hodnocení
Funkce	2
Vzhled	3,5
Přehlednost a umístění ovládacích prvků	3
Jak se SW ovládá znalému člověku	2
Jak se SW ovládá laikovi	2,5
Průměrná známka	2,6

**Tabulka 4: Pohoda, hodnocení firmy**

Hodnocená oblast	Hodnocení
WWW stránky	2,5
Telefonická komunikace	1
Elektronická komunikace	1,5
Neobchodní komunikace	2
Průměr	1,74

**Tabulka 5: Pohoda, specifikace**

Výrobce	Stormware
Doba vývoje	4 roky (tým 60 IT odborníků)
Programovací jazyk	C++ (fila-server verze)
Použité databáze	MS SQL
Hodnocení SW	2
Hodnocení prezentace	1,74
Průměrná cena licence	8 000 Kč

### 2.7.6 Money S3

Money S3 je produktem společnosti Cígler Software a.s., která působí v České a Slovenské republice od roku 1990. S 80.000 licencemi se řadí mezi nejpoužívanější softwary tohoto typu na trhu. Malý výzkum provedený v rámci této práce to nepřímo potvrzuje, protože ze středních firem, které odpověděly, se jiná odpověď než Money S3 nevyskytla. To, že návratovost odpovědí ze středních firem byla mizivá způsobuje zkreslení, při kterém Money S3 nemá dostatečné zastoupení.

Společnost Cígler Software zaměstnává asi 100 kmenových IT odborníků, má 4 podobočky v ČR a dvě na Slovensku. Obrat za poslední fiskální období překročil 160 milionů Kč.

Velmi zajímavé je, že zpočátku (v roce 1990) software vyvíjely pouze tři osoby. Díky kvalitě si však program získal místo na trhu. Společnost nyní nabízí několik různých systémů, Money S3 (který je v této práci předmětem zájmu), Money S4 (informační systém) a Money S5 určený převážně pro korporátní klientelu.

Produkt s omezeným počtem provedených záznamů je možné stáhnout na adrese <http://www.money.cz/ke-stazeni/>.

Samostatný systém je pak velmi intuitivní. Už instalace překvapila – během ní stačilo zadat identifikační číslo společnosti, která bude systém používat a veškeré další informace byly staženy ze systému ARES. Propojení s tímto systémem přináší programu velké plus.

Ovládání softwaru je jednoduché a příjemné. Občas se bohužel není možné vyhnout hledání některé z funkcí. Přesto jde o velmi povedený SW, který nabízí všechny funkce, včetně příprav podkladů na daňové přiznání, uzávěrky apod. Hůře se však pracovalo s pokladnou, na které by problém sledovat pohyby hotovosti.

Společnost má příjemné webové stránky, které patří k průměru toho, co lze na trhu vidět. Naopak přístup zaměstnanců je nadprůměrný. Obchodníci jsou velmi ochotní, a to nejen při vidinách prodeje, ale také při podpoře zákazníka. Vynikající byla i komunikace prostřednictvím elektronické pošty, která byla velmi rychlá a věcná. V případě obecných dotazů na SW také velmi příjemná. Tato společnost opravdu vzbuzuje dojem toho, že jí jde o zákazníka, nejen o velké zisky.

Je také vhodné zmínit, že začínající podnikatelé mohou tento systém používat zdarma, protože omezení free verze je opravdu benevolentní a bylo by třeba vydat velké množství dokladů, aby začínajícímu podnikateli tato verze nestačila. Pokud by se tak stalo, tak základní verzi programu lze zakoupit za 2.000 Kč. Verze nejdražší pak stojí 13.000 Kč.

**Tabulka 6: Money, hodnocení SW**

Hodnocená oblast	Hodnocení
Funkce	1,5
Vzhled	1,5
Přehlednost a umístění ovládacích prvků	2
Jak se SW ovládá znalému člověku	1
Jak se SW ovládá laikovi	2
Průměrná známka	1,4



**Tabulka 7: Money, hodnocení firmy**

Hodnocená oblast	Hodnocení
WWW stránky	1,5
Telefonická komunikace	1
Elektronická komunikace	1
Neobchodní komunikace	1
Průměr	1,13

**Tabulka 8: Money, specifikace**

Výrobce	Cígler Software
Doba vývoje	od roku 1990
Programovací jazyk	Delphi (Money S4, S5 C#)
Použité databáze	MS SQL 2005/2008
Hodnocení SW	1,4
Hodnocení prezentace	1,13
Průměrná cena licence	6 000 Kč

### **2.7.7 ADMWin**

Tento software vyvíjí Ing. Jehlička od roku 1991 pod jménem ADM. Později byl program přejmenován na ADMWin. U tohoto SW lze velmi těžko vysledovat počet prodaných licencí, protože jednu licenci je možno použít na jeden počítač, stejně jako pro celou firmu.

System samotný se velmi liší od veškeré konkurence. Jako jediný nespolupracuje s žádným databázovým systémem. Pracuje přímo se soubory DBF, ve kterých jsou veškeré informace zaznamenány.

ADMWin je možné zakoupit ve dvou verzích, a to ve verzi pro daňovou evidenci a pro podvojně účetnictví a to za 2.900, resp. 7.900 Kč.

Software nepatří k nejpřehlednějším či nejlépe použitelným. Relativně příjemné ovládání a jednoduchost z něj dělají dobrého soupeře pro ostatní programy. Svou cestu k zákazníkům si však najde zejména díky vynikající podpoře a ochotě Ing. Jehličky.

Tento systém je distribuován zejména v Praze, kde je sídlo výrobce, proto se při výzkumu v Pardubickém kraji téměř nevyskytoval. Svým podílem patří tento program k menším hráčům na trhu.

Produkt je možné k vyzkoušení stáhnout z webových stránek: <http://www.admwin.cz/download/>. Tato ukázková verze má omezení na 200 záznamů v hlavní knize, pro větší počet záznamů je nutné si produkt zakoupit.

**Tabulka 9: ADM, hodnocení SW**

Hodnocená oblast	Hodnocení
Funkce	3
Vzhled	4
Přehlednost a umístění ovládacích prvků	2
Jak se SW ovládá znalému člověku	2,5
Jak se SW ovládá laikovi	3
Průměrná známka	2,9

**Tabulka 10: ADM, hodnocení firmy**

Hodnocená oblast	Hodnocení
WWW stránky	2
Telefonická komunikace	1
Elektronická komunikace	1
Neobchodní komunikace	1
Průměr	1,25

**Tabulka 11: ADM, specifikace**

Výrobce	Ing. Jehlička
Doba vývoje	od roku 1991
Programovací jazyk	CA-VO 2.5 (objektová nástavba nad C++)
Použité databáze	žádné
Hodnocení SW	2,9
Hodnocení prezentace	1,25
Průměrná cena licence	5 000 Kč

### 2.7.8 Altus Vario

Název programu se skládá z názvu společnosti Altus a typového označení programu, tedy Vario. Jde o další z řady produktů určeného pro malé a střední podniky. Přestože je tento SW prezentován jako SW určený primárně na řízení podniku, jeho vybavení tomu neodpovídá.

O společnosti, programovém vybavení ani působnosti společnosti na trhu jsou jen minimální údaje, protože společnost, přestože se tváří jako firma s moderní a luxusní image, nehodlá komunikovat jinak, než čistě obchodně. Její zaměstnanci odpovídají na emaily velmi pomalu a při telefonním rozhovoru nejsou příjemní. Jediné získané informace jsou od jednoho z dealerů, který tvrdí, že program je napsán v C++ a využívá databáze MS SQL.

Internetové stránky společnosti jsou zastaralé, nepřehledné a nelze na nich najít požadované informace. Primárním účelem těchto webových stránek je podání základních informací a navedení na objednání testovací verze programu, který lze najít na <http://www.ekonomicky-software-system.cz/objednavka-ekonomickeho-software>. Při tomto objednání je požadováno mnoho kontaktních údajů na společnost. Objednané CD je prostřednictvím pošty doručeno po 6 dnech a je k němu přiložen krátký dopis. Narozdíl od internetových stránek obsahuje CD ceník tohoto softwaru.

Samotný produkt není vizuálně moc příjemný. Jeho obsluha je těžkopádná a nepřehledná. Přesto SW obsahuje velkou spoustu funkcí, pokročilý způsob zálohování a některá usnadnění, která u konkurence nejsou k nalezení. Cena software-u se pohybuje od základní verze za 4.000 Kč, až po verzi s většinou funkcí za 14.000 Kč.

**Tabulka 12: Altus, hodnocení SW**

Hodnocená oblast	Hodnocení
Funkce	2
Vzhled	4
Přehlednost a umístění ovládacích prvků	3,5
Jak se SW ovládá znalému člověku	3
Jak se SW ovládá laikovi	3
Průměrná známka	3,1

**Tabulka 13: Altus, hodnocení firmy**

Hodnocená oblast	Hodnocení
WWW stránky	3,5
Telefonická komunikace	3,5
Elektronická komunikace	4
Neobchodní komunikace	5
Průměr	4

**Tabulka 14: Altus, specifikace**

Výrobce	Altus Software
Doba vývoje	neznámo
Programovací jazyk	C++
Použité databáze	MS SQL
Hodnocení SW	3,1
Hodnocení prezentace	4
Průměrná cena licence	9 000 Kč

### 2.7.9 Manas 2010

Společnost Manas.info s.r.o je společností, která odmítá komunikovat, a to jak s potenciálním zákazníkem, tak s člověkem, který se o jejich produkt zajímá jinak.

Nutno říci, že webové stránky této společnosti jsou zastaralé a nevhledné, ale přesto poskytují veškeré informace, které by uživatel mohl potřebovat. Produkt společnosti je možné nalézt na <http://manas.info/ekonomicky-system-download/demoverze-es-manas>.

Samotný systém je na tom vzhledově mnohem lépe. Je velmi přehledný a téměř nikdy nezahltí uživatele informacemi. Přesto se s ním pracuje velmi těžkopádně a některé základní funkce nejsou podporovány.

Mezi velké výhody tohoto systému patří to, že bez problému funguje na starých počítačích s 333 MHz a operačním systémem MS Windows 95. Na druhou stranu to přináší otázku, zda program během své desetileté historie prošel nějakou celkovou inovací.

Jako u jediného z uvedených SW je možné mít licenci v pronájmu. Pronájem licence stojí 300 Kč/měs.

**Tabulka 15: Manas, hodnocení SW**

Hodnocená oblast	Hodnocení
Funkce	4
Vzhled	3
Přehlednost a umístění ovládacích prvků	1,5
Jak se SW ovládá znalému člověku	4
Jak se SW ovládá laikovi	4
Průměrná známka	3,3

**Tabulka 16: Manas, hodnocení firmy**

Hodnocená oblast	Hodnocení
WWW stránky	4,5
Telefonická komunikace	5
Elektronická komunikace	5
Neobchodní komunikace	5
Průměr	4,9

**Tabulka 17: Manas, specifikace**

Výrobce	Manas.info s.r.o
Doba vývoje	cca 10 let
Programovací jazyk	Delphi, Nexus DB3
Použité databáze	neznámo
Hodnocení SW	3,3
Hodnocení prezentace	4,9
Průměrná cena licence	5 000 Kč, 300 Kč/měs. pronájem

### **2.7.10 Pořadí hodnocených ESS**

Vítězem testu se jednoznačně stává ESS Money S3, který velmi zaujal po všech hodnocených stránkách, jeho velkou přidanou hodnotou jsou možnosti technické podpory a konzultací s ochotnými zaměstnanci společnosti Cigler Software.

Boj o druhé místo byl velmi těsný. Nakonec bylo nutné přihlédnout k tomu, že největší devizou systému Pohoda je jeho jednoduchost a absence funkcí a že výsledky poskytované programem nejsou v příjemném formátu. Proto druhé místo připadlo softwaru ADMWin, který si drží mírný náskok právě díky většímu množství funkcí a lepší technické podpoře.

Třetí místo tedy získává systém Pohoda, momentálně zřejmě nejrozšířenější ESS pro živnostníky a malé podniky v Pardubickém kraji.

Souboj o čtvrté a páté místo byl také velmi těžký. Oba testované SW lze, na základě testu, považovat za nekvalitní a jejich prodejní i technická podpora je přinejmenším slabá. Čtvrté místo nakonec získal Altus Vario, přes jeho nevybíravé obchodní praktiky a neochotu komunikovat, jde o kvalitnější ESS, než konkurenční Manas 2010.

Systému Manas 2010 připadá místo páté. Nelze říci, že nezaslouženě. Dá se přepokládat, že pokud by se testu zúčastnilo více systémů, tento by opět skončil

na posledním místě. Na trhu se jistě najdou jeho zastánci, ale během testu nevyplýnuly na povrch žádné z jeho výhod.

**Tabulka 18: Celkove hodnoceni ESS**

Název	Money S3	ADMWin	Pohoda	Altus Vario	Manas 2010
Výrobce	Cígler Software	Ing. Jehlička	Stormware	Altus Software	Manas.info s.r.o
Hodnocení SW:	1,4	2,9	2	3,1	3,3
Prezentace SW:	1,13	1,25	1,74	4	4,9
Cena	6 000 Kč	5 000 Kč	8 000 Kč	9 000 Kč	5 000 Kč, 300 Kč/měs. pronájem

### 2.7.11 Závěry z testování jednotlivých ESS

Na našem trhu je dostupné velké množství ESS, které se velmi liší vlastní kvalitou i kvalitou technické podpory a prodeje.

Stěžejní programová kvalita byla uspokojivá u tří z testovaných programů. Třetí v pořadí, program Pohoda, obstál jen díky velké jednoduchosti, bez možnosti aplikace složitých funkcí. Jako komplikaci při užívání toho SW je navíc třeba uvést, že výstupy jsou srozumitelné převážně jen pro člověka s ekonomickým vzděláním. Navíc program patří mezi cenově hůře dostupné. Přesto základní myšlenka – tedy jednoduchost je vynikajícím předpokladem pro masové rozšíření programu.

Vítěz testu, Money S3, je ukázkou toho, jak by ESS měl fungovat. Avšak velké množství funkcí může být pro laika matoucí, zřejmě proto je SW používán zejména ve středních a větších společnostech. Jednoduché pracovní principy a možnost přiřazování dokladů bez jejich fyzického vytvoření v systému je velmi efektivním nástrojem.

## **3 Praktická část**

### **3.1 Cíle**

Hlavním cílem je vytvoření funkčního ESS, který bude splňovat níže uvedené požadavky, bude funkční a prakticky využitelný v prostředí malých a středních společností.

Vzhledem k rozsáhlosti řešení ESS bude předmětem práce jen samotné srdce těchto systémů, a to je ekonomický systém.

#### **3.1.1 Systém vzdálené správy skladu**

Systém vzdálené správy skladu (SVSS), který byl vytvořen jako semestrální projekt na předměty Databázové systémy 2 a Návrh a tvorba www stránek, je použitelný jako pripojitelný modul k tomuto projektu, přičemž některé tabulky jsou užity společně a některé jsou pro každý systém vlastní.

#### **3.1.2 Pojmenování systému**

Pro jednoduchost a názornost získá systém hned v úvodu název. A to proto, aby v dalším textu bylo jasně označeno, kdy jde o tento konkrétní systém a kdy jde o rozpravu nad obecnými ESS.

Název ESS vytvořeného v této práci bude Prefekt. Název původně pochází z latiny, výraz „prefekta“ znamená „správa“. Vzhledem k užití systému jde tedy o vhodný název.

Název byl vybrán jako vhodný, protože splňuje základní lingvistické předpoklady pro název (je krátký, jednoduchý, snadno zapamatovatelný a zvučný) a díky snadné záměně se slovem „perfekt“ či „perfektní“ je název velmi snadno zpracovatelný v oblasti marketingu. Třetí výhodou je to, že slovo není často užíváno a proto se při správné SEO optimalizaci nebude problém na produkt upozornit v rámci sítě internet.

### **3.2 Zadání pro tvorbu vlastního ESS**

Pro tvorbu úspěšného systému je nutno už od počátku vědět, co má systém umět a co bude obsahovat.

#### **3.2.1 Způsob tvorby zadání**

Na tvorbě zadání se podílely zejména dva faktory, a to znalosti a zkušenosti převzaté od společnosti Pozitron technology k.s. Celý tento projekt je namířen na praktické

použití ESS v této společnosti (jakožto ukázkové malé společnosti); pro testování budou využita faktická data této společnosti.

Druhým vlivem je provedený průzkum v předchozí části práce. Zjištěné skutečnosti budou mít vliv na tvorbu vlastního systému. Zkušenosti získané s ovládáním pak umožní vytvořit jednodušší a intuitivnější systém.

### **3.2.2 Zadání dle průzkumu**

Protože je žádoucí, aby systém vyplňoval mezeru na trhu, je vhodné soustředit se na vytvoření silných stránek systému v místech, kde ostatní systémy mají místa slabá.

Prioritou systému Prefekt tak bude jednoduchost a přehlednost. Z uživatelského pohledu by mělo mít velmi jednoduché ovládání a měl by ho zvládnout i laik. Přesto by systém měl mít všechny základní funkce (o funkce ostatní se mohou starat později vytvořené moduly).

Výstupy z programu by měly být použitelné jak pro daňového poradce či jinou osobu zpracovávající daň, tak pro vytvoření samotného daňového přiznání. Databáze by měla být snadno modifikovatelná pro případ legislativních změn.

### **3.2.3 Zadání dle požadavků společnosti**

Systém musí být zabezpečený s možností zálohování dat. Vhodné by bylo sledování, kdo provedl příslušnou účetní operaci. Systém by samozřejmě měl být funkční tak, že vstupní i výstupní hodnoty budou vždy odpovídat realitě a nebude docházet k matematickým chybám.

Pro potřeby společnosti musí systém obsahovat operativní a evidenční funkce. Není tedy třeba, aby systém sám o sobě vydával doklady, neboť to by ho činilo zbytečně komplikovaným. Mnohem vhodnější řešení je možnost zadat doklad přímo k příslušné operaci.

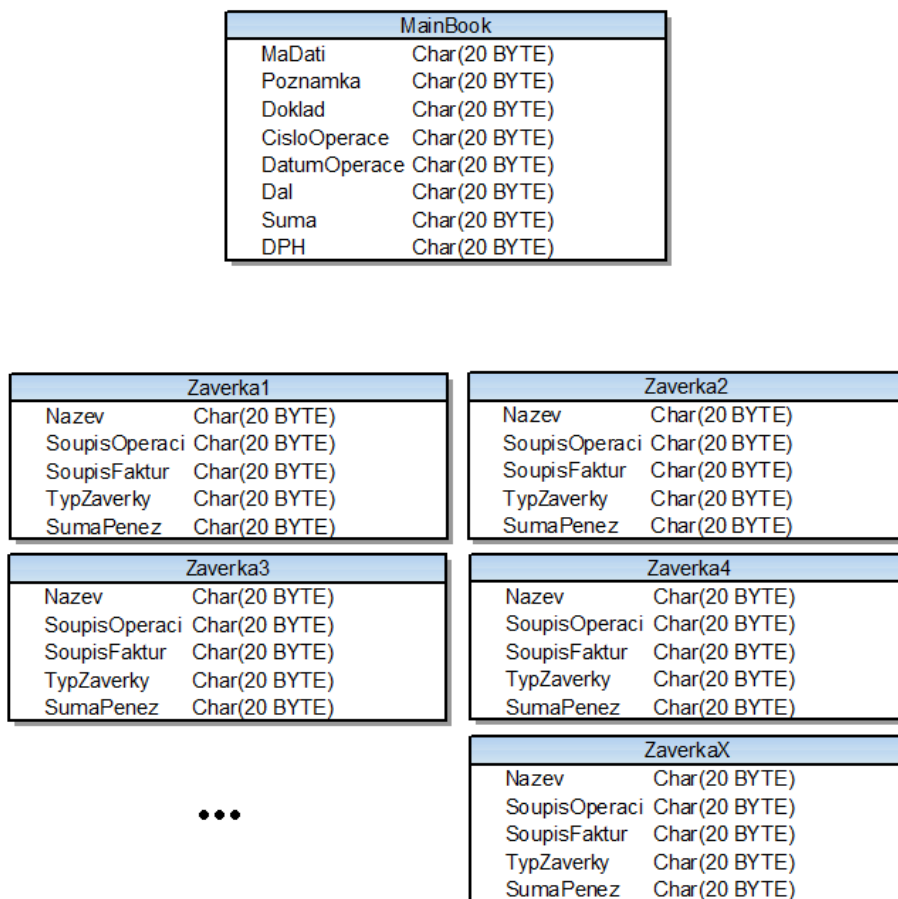
Zásadní je možnost přístupu podle několika uživatelských rolí. Např. role účetní by měla mít veškeré možnosti nad systémem, narozdíl od vedoucích, kteří by měli mít možnost pouze prohlížet. Možnosti modifikace struktury databáze by neměla být přístupná nikomu. Modifikace uživatelů, jejich tvorbě či rušení zase přísluší pouze správci systému, který by další možnosti mít neměl.

### **3.2.4 Zadání od účetní společnosti**

Požadavky na účetní systém jsou takové, že musí existovat hlavní kniha, která bude evidovat dva účty z účetní osnovy, datum vytvoření, číslo dokladu, hodnotu plnění, případně DPH plnění, slovní komentář, % DPH a číslo operace.



Pod touto hlavní knihou by měly existovat soupisy faktur, které jsou určeny pro jednotlivé uzávěrky.



Obrázek 2: Nenormalizovaná databáze

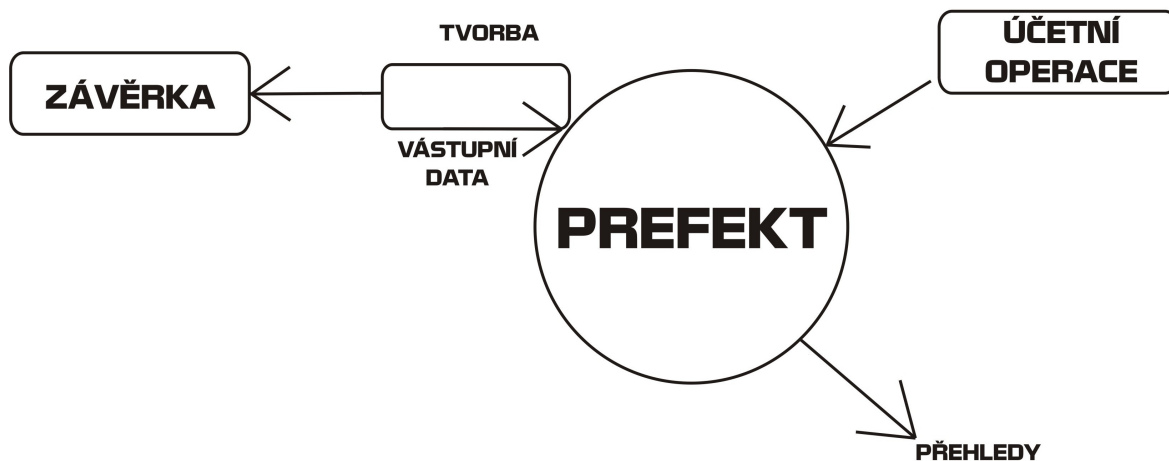
### 3.3 Příprava prací

Po obdržení zadání je nezbytné toto zadání zpracovat. Veškerá získaná data jsou zanalyzována a je vybrán nejvhodnější způsob jejich zpracování, včetně výběru prostředků zpracování.

#### 3.3.1 Kontextová analýza

Tvorba kontextového diagramu pomůže určit základní požadavky na chování systému z vnější strany. Pokud víme, co má systém provádět, je důležité určit, co bude do systému vstupovat a co bude jeho výstupem.

# KONTEXTOVÝ DIAGRAM



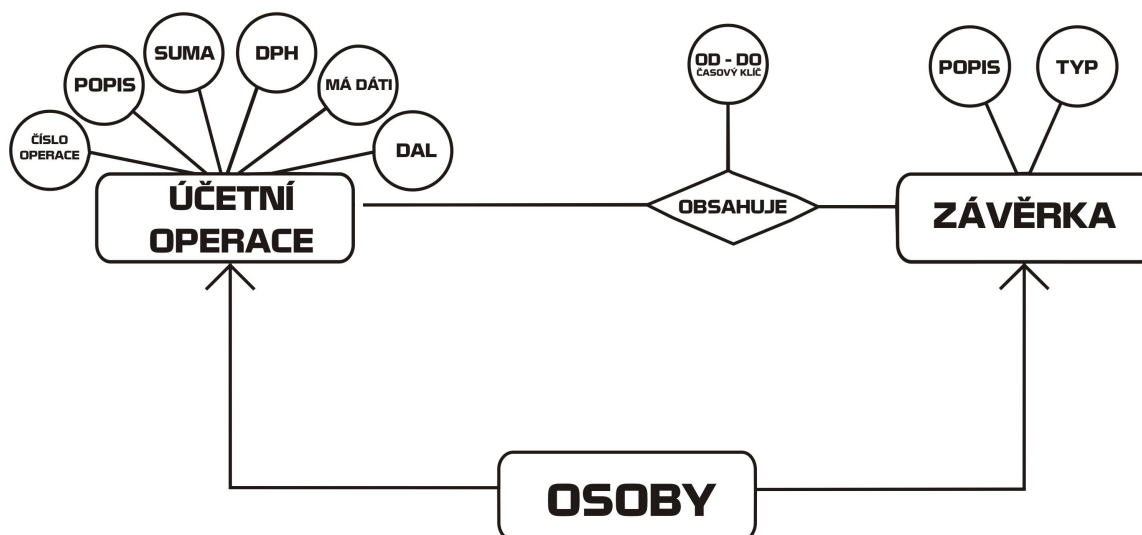
Obrázek 3: Kontextový diagram

V tomto konkrétním případě je stěžejním vstupem účetní operace. Vnitřně tvořeny jsou pak závěrky, které jsou i jedním z výstupů. Druhým požadovaným výstupem jsou veškeré přehledy o účetním dění v systému.

## 3.3.2 Analýza vztahu entit

Prostřednictvím této analýzy je možné získat na celou problematiku pohled z nejnižší, datové vrstvy.

# ENTITY RELATIONSHIP MODEL



Obrázek 4: Entity relationship model

Model zobrazuje datové požadavky na systém. Vzhledem k velkému množství záznamů a nutnosti udržování dat v konzistentní podobě by bylo nejvhodnější použít k realizaci těchto vztahů některou z databází.

### **3.3.3 Základní rozvržení prostředků**

Z předešlé analýzy vyplývá, že nejlepším řešením datové úrovně systému bude databázový systém.

Je zřejmé, že samotný databázový systém, s případnou nadstavbou, na vše stačit nebude. Zejména zabezpečení by bylo velmi slabé. Stejně jako přístup, který by byl často jen lokální. Největší nevýhodou by pak bylo velmi nekvalitní uživatelské prostředí. Uživatel by musel znát strukturu databáze, příslušné příkazy pro ovládání databáze a vzhledem k řešené problematice také pravidla účetnictví.

Právě kvůli přístupu, který by měl odpovídat vzdálené instalaci, popřípadě multilokální instalaci, by bylo vhodné zvolit takový prostředek, který umožňuje spolupracovat s databází a pracovat na všech platformách. Jako nejperspektivnější se tak jeví PHP – které bylo nakonec také zvoleno.

### **3.3.4 Zvolení databázového systému**

Vzhledem k tomu, že má jít o open-source aplikaci s vysokou úrovní zabezpečení, byly jako možné prostředky vybrány tři databáze, a to MySQL - pro svoji rozšířenost, Microsoft SQL a Oracle SQL 10g Express - pro širší podporu, lepší zabezpečení a větší robustnost samotného databázového řešení. Protože jsou všechny systémy zdarma, splňují podmínku možnosti nasazení v open source produktu.

Vítězem výběru se stalo řešení od společnosti Oracle. Hlavní devizou jsou zkušenosti, které autor se systémem má, robustnost a možnosti systému.

### **3.3.5 Základní zabezpečení**

Základním zabezpečením rozumíme přístup k systému jen pro ty osoby, kterým to bude umožněno, tedy uživatelům registrovaným přímo v systému. Toto opatření sice může být řešeno pomocí .htaccess přímo v zabezpečení serveru, ale vzhledem k tomu, že ne vždy je .htaccess podporován, je zvolen jednoduchý login v PHP aplikaci (tj. přihlášení na straně serveru), přičemž jednotlivé uživatelské účty budou uloženy přímo v MySQL databázi.

### **3.3.6 Zabezpečení databáze na uživatelské úrovni**

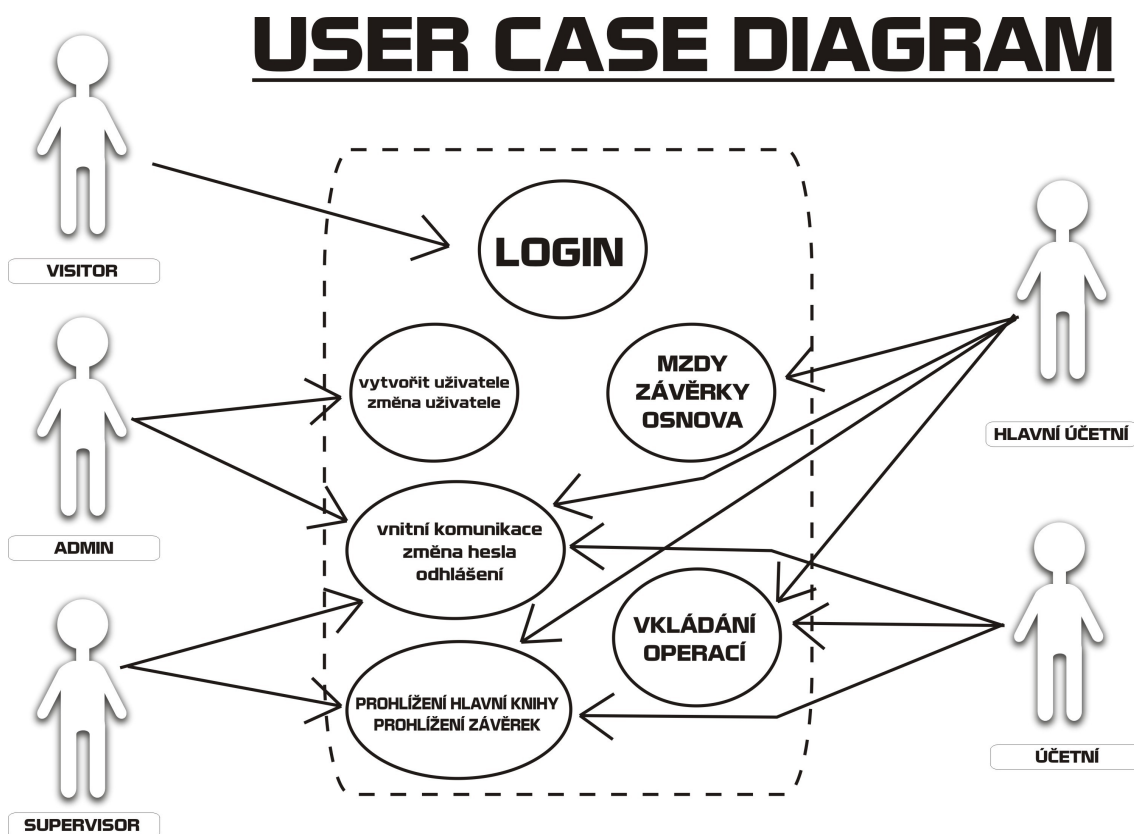
Systém bude nutno vybavit vlastním instalačním rozhraním. Toto instalační rozhraní vytvoří veškeré tabulky v databázi a prvního uživatele. Až poté, co bude instalační

rozhraní odstraněno, bude možné spustit systém. Tímto bude zajištěno, že již nebude možná žádná další manipulace se strukturou databáze.

### 3.3.7 Uživatelské role

Uživatelské role jsou přesně dle požadavků specifikovány do tří základních rolí. A to na:

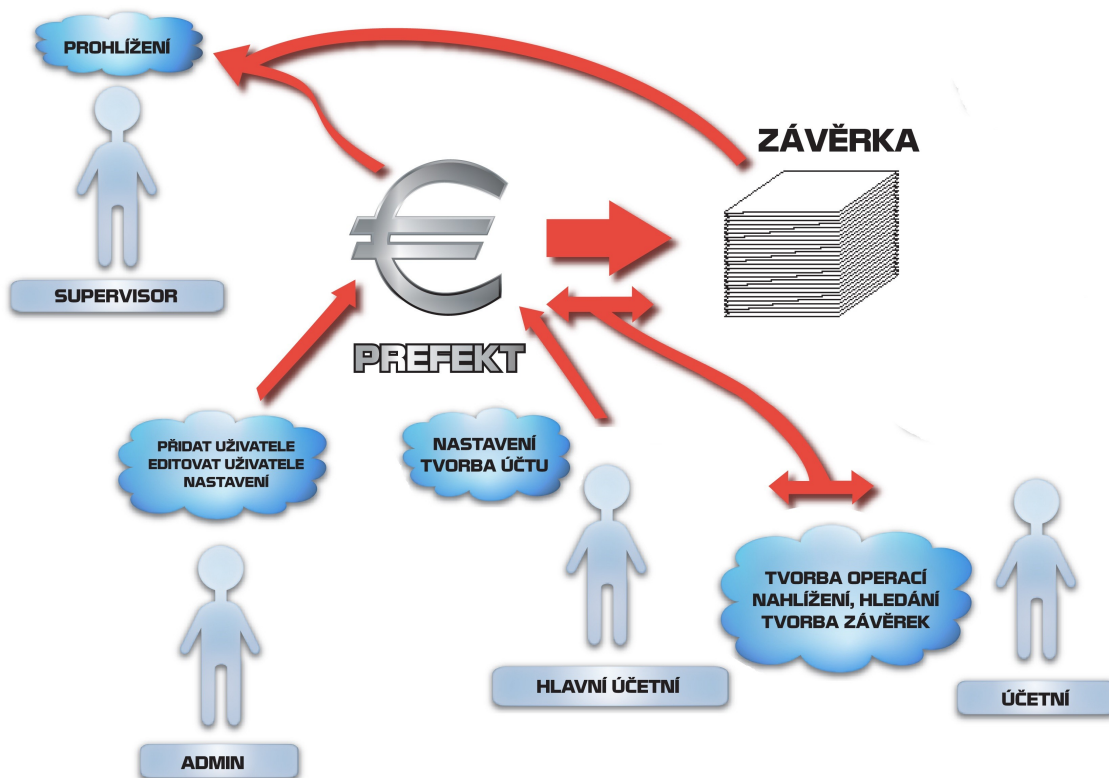
- administrátora, s možností úpravy a přidávání uživatelů,
- účetní, která má možnostmi práce s účetními operacemi, vytvářením daňových přiznání a dalších přehledů,
- hlavní účetní, s právy účetní rozšířené o možnosti úpravy účetní osnovy a dalších účetních systémů
- a kontrolní roli, která je vhodná pro vedoucí podniku či pro vzdálené kontroly auditorů (bez možnosti manipulace s daty).



Obrázek 5: User case diagram

### 3.3.8 Ritch picture

Ze všech uvedených skutečností je možné vytvořit ritch picture, jednoduchý obrázek, který zobrazuje možnosti uživatelských i systémových akcí.



Obrázek 6: Ritch picture

### 3.3.9 Normalizace databáze

Vzhledem k nevhodnému rozvržení v původním návrhu bylo nutné databázi normalizovat. Nově vzniklé tabulky i jejich vztahy jsou popsány níže, v kapitole 3.4. Databáze.

### 3.3.10 Uživatelské rozhraní

Pro maximální jednoduchost uživatelského rozhraní bylo zvoleno jedno hlavní menu (může být i vícesložkové – tzn. obsahovat více kategorií) pro většinu ovládacích funkcí. Doplnkové horní menu by mělo uživatelům poskytovat základní podporu pro komunikaci a správu vlastního účtu – tedy ne pro ovládání samotného SW.

Struktura menu je tedy jednoduše větvená do jednoho řádu a je znázorněna na obrázku v příloze.

### 3.3.11 Vzhled rozhraní

Jako základ pro uživatelské rozhraní byl použit template, který je zdarma ke stažení (šířený pod opensource licenci) na stránkách <http://www.templatebasics.com/web-templates/isometric/>. Template byl barevně upraven, bylo přidáno CSS nastavení pro tisk.

Celý vzhled je validní dle CSS 2.0 a obsahuje pouze prvky zobrazitelné ve všech nejběžnějších prohlížečích.

## 3.4 Databáze

Užitá databáze, vytvořená v databázovém systému Oracle 10g express, slouží pro správu datové vrstvy tak, aby nezáleželo na jazyce, který bude tvořit uživatelské prostředí. Veškeré vstupně/výstupní, evidenční a početní operace jsou realizovány pomocí funkcí, či jednoduchých příkazů v jazyce SQL.

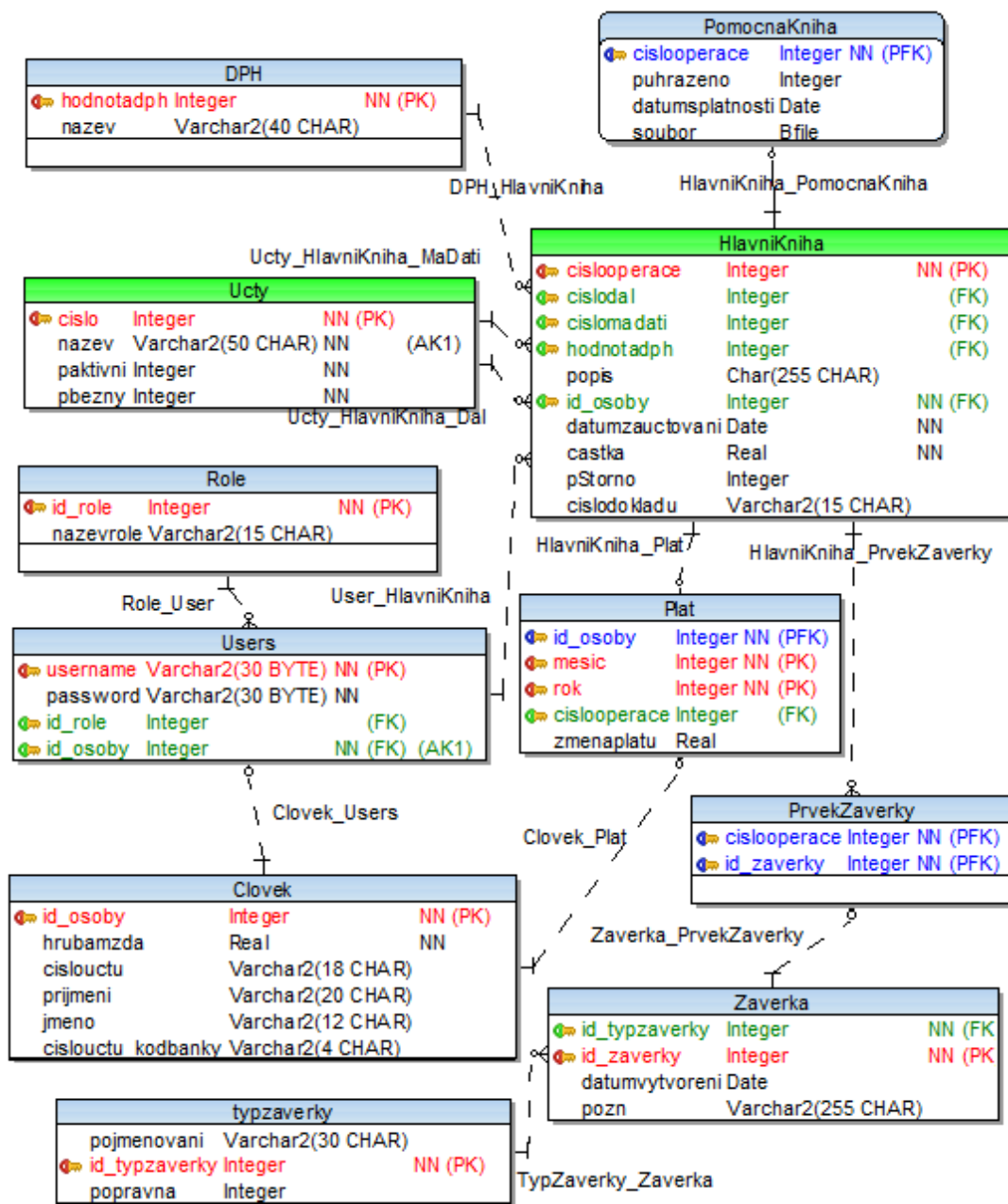
### 3.4.1 Návrhový SW

Pro snazší tvorbu databáze byl užit program Toad Data Modeler 3 ve studentské licenci. V tomto programu proběhla normalizace databáze. Výsledným výstupem programu je, kromě schématu, sada SQL příkazů pro tvorbu tabulek, klíčů, restriktivních omezení, procedur a funkcí (model v Toad Data Modeleru je součástí CD s přílohou).

### 3.4.2 Normalizovaná databáze

Na obrázku níže je vidět vzhled databáze po normalizaci do 3. normální formy. Barevně jsou označeny dvě stěžejní části databáze. Jde o samotné jádro účtovacího systému vycházející z principů účetnictví.

Většina součástí datové vrstvy (triggery, funkce a procedury v databázi) pracuje s tímto jádrem v současné podobě. Je proto velmi nevhodné zasahovat do této části systému v případě, že bude nutné provést úpravy struktury databáze.



Obrázek 7: Entity relationship diagram

### 3.4.3 Uživatelské role v databázi

Databáze je vybavena dvěma uživateli s rozličnými uživatelskými rolemi. První je uživatel, který má pouze přístup čtení ke čtení tabulek Role a Users. Komunikace prostřednictvím tohoto uživatele je proto ideální pro přihlašování do databáze. Z vnější strany nejsou ohroženy žádné další tabulky.

Druhá uživatelská role je již standardní se všemi právy k tabulkám. Tato role má však být dostupná až po přihlášení uživatele. Další práva každého uživatele pak mohou být omezena na uživatelské úrovni.

### 3.4.4 Značení prvků v tabulkách

Prvky v tabulkách jsou značeny českými či anglickými samovysvětlujícími názvy. Ve snaze dosáhnout maximální jednoduchosti a výkonu databáze bylo nutno zakomponovat několik sloupců, které budou označovat příznaky. V této databázi jsou vždy pojmenovány samovysvětlujícím názvem, avšak s prefixem „p“, např.: puzrazeno.

### 3.4.5 Popis jednotlivých tabulek

#### Ucty

Tabulka reprezentuje účet ve smyslu (podvojného) účetnictví. Základní předplnění podle účetní osnovy doporučené ministrestvem financí ČR bude provedeno v již výchozím stavu ihned po instalaci SW. Vlastní účty je možné kdykoli přidat.

Příznaky pAktivni a pBezny znázorňují, kde má daný účet strany má dáti a dal, resp. určení, zda jde o účet pro běžné účtování, nebo účet pro konečné účtování, závěrky atp.

Tabulka 19: Ucty

Název	Typ	Omezení	Poznámka
cislo	Integer	NN (PK)	
nazev	Varchar	NN	
paktivni	Integer	NN	
pbezny	Integer	NN	

#### HlavniKniha

Stěžejní tabulka celé databáze je právě Hlavní kniha. V této tabulce dochází ke všem účetním operacím (tj. operace provádí funkce či procedury pracující s touto tabulkou).

Většina sloupců tabulky je tvořena klíči ostatních tabulek. Vlastními sloupci jsou pouze popis, datum zaúčtování a částka.

Jediný sloupec, který přímo nesouvisí s účetními operacemi je cizí klíč id\_osoby. Toto je evidenční řádek, aby bylo vždy možné dohledat, která osoba s účetním systémem pracovala.



**Tabulka 20: Hlavni kniha**

Název	Typ	Omezení	Poznámka
cislooperace	Integer	NN (PK)	
cislodal	Integer	(FK)	Ucty:cislo
cislomadati	Integer	(FK)	Ucty:cislo
hodnotadph	Integer	(FK)	
id_osoby	Integer	NN (FK)	Kdo provedl
popis	Varchar		
datumzauctovani	Date	NN	
castka	Real	NN	

### PomocnaKniha

Pomocná kniha je tabulka, která pouze rozšiřuje hlavní knihu. Zavedena je proto, že se v ní udržuje elektronický formát daňového dokladu a tabulka tak dosahuje větší datové velikosti. Použitím oddělené tabulky jsou sníženy datová a výpočetní zátěž, které jsou vytvářeny vlastní správou databáze.

Tabulka je vhodná pro přidávání dalších údajů do hlavní účetní části, bez nutnosti manipulace s hlavní účetní částí.

Tabulka v tomto stavu nese informace o faktuře (pojem faktura není zcela správný, může jít i o dodejku, prodejku, paragon atp., avšak běžně je používáno zjednodušení, kde jdou daňové doklady označovány za faktury) patřící k operaci, datu splatnosti, číslu dokladu a příznaku, zda byla daná faktura již uhrazena.

**Tabulka 21: PomocnaKniha**

Název	Typ	Omezení	Poznámka
cislooperace	Integer	NN (PFK)	
puhrazeno	Integer		
datumsplatnosti	Date		
soubor	Bfile		

### DPH

Tabulka DPH je velmi jednoduchou tabulkou, která nese údaj o DPH v % hodnotě. Druhým řádkem je pojmenování této hodnoty.

Vzhledem k tomu, že při legislativní změně DPH nedojde ke zpětné změně, je možné jako primární klíč užít samotnou hodnotu DPH.

**Tabulka 22: DPH**

hodnotadph	Integer	NN (PK)	Uváděna v %
nazev	Varchar	NN	

### PrvekZaverky

Jde o pomocnou tabulku, která zastupuje vazbu n:n mezi tabulkou Zaverky a tabulkou HlavniKniha. Primární klíče z obou tabulek zde spolu tvoří primární klíč tabulky.

**Tabulka 23: PrvekZaverky**

Název	Typ	Omezení	Poznámka
cislooperace	Integer	NN (PFK)	
id_zaverky	Integer	NN (PFK)	

### Zaverka

Tato tabulka tvoří přehled veškerých provedených závěrek. Díky tabulce Prvků závěrky je pak možné sledovat veškeré částky v závěrečce vždy novým propočtem. Pro udržení konzistence dat je to nejvhodnější řešení.

**Tabulka 24: Zaverka**

Název	Typ	Omezení	Poznámka
id_typzaverky	Integer	NN (FK)	
id_zaverky	Integer	NN (PK)	
datumvytvoreni	Date		
pozn	Varchar		

### TypZaverky

Tabulka nese informaci o typech závěrky. Opět jde o systém (organizační), který zvyšuje přehlednost a umožňuje vést více typů závěrek (měsíční, čtvrtletní, roční atp.). Příznak popravna znázorňuje, zda je závěrka standardní, nebo jde o opravnou závěrku.

**Tabulka 25: TypZaverky**

Název	Typ	Omezení	Poznámka
id_typzaverky	Integer	NN (PK)	
pojmenování	Varchar		
popravna	Varchar		

### Role

Tabulka uživatelských rolí slouží jazyku, implementujícímu uživatelské rozhraní, pro předání informací o tom, jaké role existují.

**Tabulka 26: Role**

Název	Typ	Omezení	Poznámka
id_role	Integer	NN (PK)	
nazevrole	Varchar		

### Users

Tato tabulka udržuje informace o uživatelských účtech oprávněných k přístupu do systému. Hlavními údaji jsou uživatelské jméno a heslo. Údaj o roli pak označuje, jaká má uživatel práva. Pokud je role null, uživatel nemá právo přístupu do systému a tento fakt musí být ošetřen v uživatelském rozhraní.

Primární klíč je totožný s primárním klíčem tabulky Clovek, tedy id\_osoby.

**Tabulka 27: Users**

Název	Typ	Omezení	Poznámka
id_role	Integer	FK	
id_osoby	Integer	NN (PFK)	
username	Varchar	NN	
password	Varchar	NN	

### Clovek

Tabulka nese informace o všech lidech spravovaných ekonomickým systémem, přičemž tito lidé nemusí mít vlastní uživatelské účty. Kromě jejich základních iniciálů je uvedeno též bankovní spojení (pro výplatu mezd) a vlastní hodnota hrubé mzdy ze které

jsou dopočítávány zákonné položky (odvody za zaměstnance, zdravotní a sociální pojištění) a čistá mzda.

**Tabulka 28: Clovek**

Název	Typ	Omezení	Poznámka
id_osoby	Integer	NN (PK)	
hrubamzda	Real	NN	
cislouctu	Varchar		
prijmeni	Varchar		
jmeno	Varchar		
cislouctu_kodbanky	Varchar		

### Plat

Tabulka plat nese informace o konkrétních mzdách zaměstnanců. Kromě uvedeného měsíce a roku, za který je mzda vyplácena, existuje i vazba na konkrétní účetní případ, díky kterému je možné získat všechny ostatní potřebné údaje. Primární klíč v této tabulce je složený ze tří sloupců, a to identifikace zaměstnance a měsíce a roku mzdy, která je zaměstnanci připisována.

**Tabulka 29: Plat**

Název	Typ	Omezení	Poznámka
id_osoby	Integer	NN (PFK)	
mesic	Integer	NN (PK)	
rok	Integer	NN (PK)	
cislooperace	Integer	(FK)	
zmenaplatsu	Real		

### **3.4.6 Vazby mezi tabulkami**

#### Ucty\_HlavniKniha\_MaDati

Tato vazba spojuje sloupek madati v tabulce HlavniKniha se sloupcem cislo v tabulce Ucty. Spojení reprezentuje jednoznačnou vazbu mezi účtem účetní osnovy a operací (podvojného) účetnictví na straně "Má dáti".

**Tabulka 30: Vazba 1**

Kardinalita	1:n
Mandatorní rodič	ano
Mandatorní potomek	ne

#### Ucty\_HlavniKniha\_Dal

Tato vazba spojuje sloupek dal v tabulce HlaviKniha se sloupcem cislo v tabulce Ucty. Spojení reprezentuje jednoznačnou vazbu mezi účtem účetní osnovy a operací účetnictví na straně "Dal".

**Tabulka 31: Vazba 2**

Kardinalita	1:n
Mandatorní rodič	ano
Mandatorní potomek	ne

#### DPH\_HlavniKniha

Spojení přidává do hlavní knihy možnost počítat s DPH.

**Tabulka 32: Vazba 3**

Kardinalita	1:n
Mandatorní rodič	ano
Mandatorní potomek	ne

#### HlavniKniha\_PomocnaKniha

Toto spojení je zajímavé tím, že by nemuselo existovat za předpokladu, že by byla data spojena do jedné tabulky. Pokud by se tak stalo, záznam jednoho řádku by se snížil (konkrétně o velikost 1x int) a došlo by k podstatnému zjednodušení v práci s tabulkou. Na druhou stranu by došlo ke zvýšení výpočetní náročnosti při vyhledávání a práci s tabulkou z toho důvodu, že součástí tabulky PomocnaKniha je binární soubor, který může dosahovat velikost až 2GB (v praxi maximálně desítky MB).

Druhým důvodem, proč zachovat vazbu a nespojovat tabulky je zajištění přehlednosti a systematičnosti v databázi. Základní údaje jsou vedeny v samotném jádře databáze a je nepravděpodobné, že se s jejich strukturou bude manipulovat (základní

system je odolný i změnám legislativy, protože vychází z mezinárodního účtového standardu), oproti tomu možnost přidávat další doplňující údaje do PomocnaKniha je snadné a nenarušuje procedury a funkce, které pracují ve stěžejní části databáze.

**Tabulka 33: Vazba 4**

Kardinalita	1:1
Mandatorní rodič	ano
Mandatorní potomek	ne

#### HlavniKniha\_PrvekZaverky

Označuje řádky hlavní knihy, které se stanou součástí příslušné závěrky.

**Tabulka 34: Vazba 5**

Kardinalita	1:n
Mandatorní rodič	ano
Mandatorní potomek	ne

#### TypZaverky\_Zaverka

Přiřazuje různým závěrkám obecný závěrkový typ.

**Tabulka 35: Vazba 6**

Kardinalita	1:n
Mandatorní rodič	ano
Mandatorní potomek	ne

#### User\_HlavniKniha

Tato vazba zajišťuje v hlavní knize informaci o tom, který uživatel provedl danou účetní operaci.

**Tabulka 36: Vazba 7**

Kardinalita	1:n
Mandatorní rodič	ano
Mandatorní potomek	ne

### Role\_User

Prostřednictvím této vazby je každému uživateli přidána uživatelská role. Pokud uživateli žádná role přidána není, znamená to, že nemá povolen přístup k systému.

**Tabulka 37: Vazba 8**

Kardinalita	1:n
Mandatorní rodič	ne
Mandatorní potomek	ano

### Clovek\_User

Člověk evidovaný v systému může mít vlastní uživatelský účet s vlastním přístupem. Aby bylo propojení uživatele a zaměstnance možné, zajišťuje to tato vazba.

**Tabulka 38: Vazba 9**

Kardinalita	1:1
Mandatorní rodič	ne
Mandatorní potomek	ne

### Clovek\_Plat

Přiřazuje zaměstnanci možnost nárokovat si plat.

**Tabulka 39: Vazba 10**

Kardinalita	1:n
Mandatorní rodič	ano
Mandatorní potomek	ne

## HlavniKniha\_Plat

Provádí potřebná spojení tak, aby bylo možné zaměstnanci plat uhradit a provést ostatní potřebné operace spojené se zaúčtováním platu.

**Tabulka 40: Vazba 11**

Kardinalita	1:n
Mandatorní rodič	ano
Mandatorní potomek	ne

### **3.5 Realizace uživatelského rozhraní**

#### **3.5.1 Požadavky na uživatelské rozhraní**

Pro realizaci uživatelského rozhraní bylo přihlédnuto k tomu, že spousta dnes fungujících malých a středních společností nemá žádné, nebo jen malé vlastní zázemí. Tento produkt by měl být koncipován tak, aby byl užitečným doplňkem pro společnosti spravující elektronické obchody, poskytující služby u zákazníka atp.

Vytvoření uživatelského prostředí ve skriptovacím jazyce je ideální řešení. Pracovníci mohou být umístěni v několika městech, nebo pracovat z domova a přesto mají všichni možnost přístupu do systému.

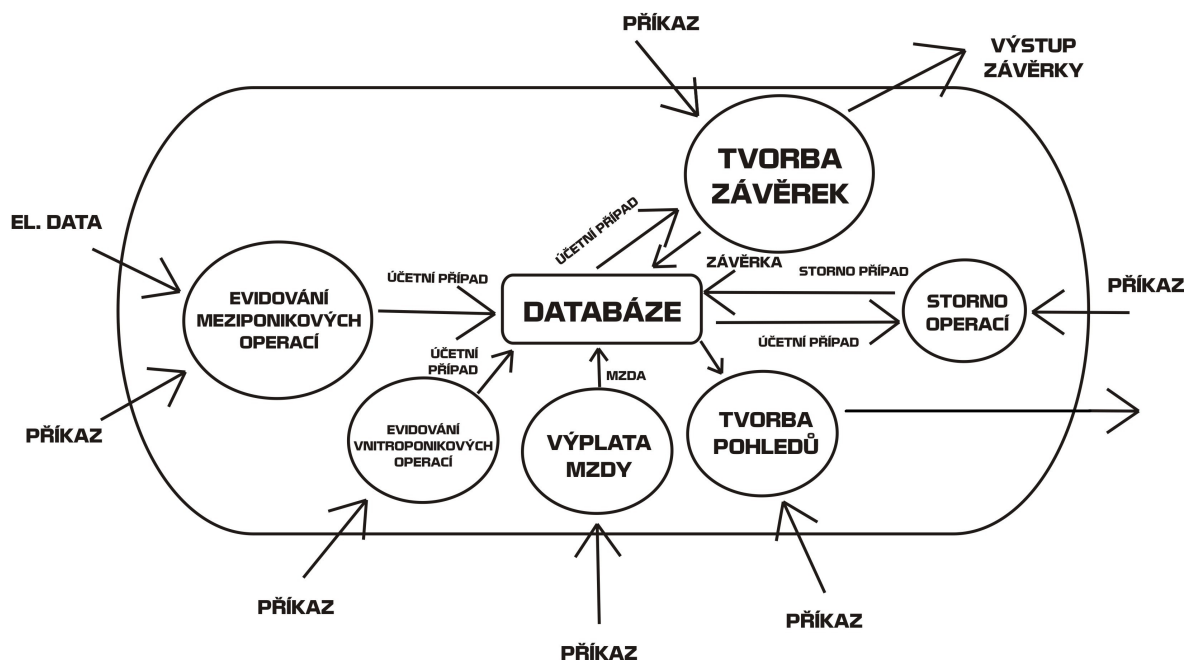
Vzhledem k tomu, že veškerá činnost s daty je prováděna na úrovni databáze, není třeba složitých funkcí, nebo složitých datových struktur, což je jedna z podmínek pro použití skriptovacího jazyka.

#### **3.5.2 Požadavky z datového hlediska**

Uživatelské rozhraní musí dokázat obsloužit všechny nezbytné funkce a podat nezbytná data k tomu, aby byl SW plně funkční.



# DFD - DATA FLOW DIAGRAM



Obrázek 8: Data flow diagram

Na obrázku jsou znázorněny všechny vrstvy programu. Vnitřek obalu patří funkcím aplikace, šipky dovnitř a ven značí komunikaci s uživatelem.

## 3.6 Metodika realizace

Řešení systému Prefekt využívá skriptovacího jazyka PHP. Pro jednotlivé části aplikace bude využita metoda vkládání hlavičky a patičky. Tato metoda má, oproti metodě vkládání stránek podle příznaku, několik výhod. Jde zejména o to, že při neprovádění operací s hlavičkou a patičkou tato zůstává vždy stejná. Druhá a podstatnější výhoda se skrývá v tom, že tato metoda poskytuje lepší možnosti řízení přístupu k jednotlivým částem programu.

### 3.6.1 Připojení k databázi

Databáze je připojena k aplikaci pomocí ovladače OCI8. Tento ovladač byl uvolněn společností Oracle pro volné použití v roce 2007. Je tedy možné ho použít i do tohoto open source produktu. Příkazy pro připojení lze nalézt v příloze.

### 3.6.2 Deoptimalizace pro vyhledávače

Soubor robots.txt je uložen v kořenovém adresáři aplikace a zakazuje robotům procházení stránek.

V hlavičce je pak další upozornění, které znemožňuje indexaci stránek.

Pomocí těchto opatření má být dosaženo toho, že se stránky s ekonomicko správním systémem nikdy nedostanou do prohlížečů a nestanou se tak místem, jež navštěvují nepovolaní uživatelé.

### 3.6.3 Soubory

Pro větší přehlednost je celý systém logicky rozdělen do několika adresářů, které zaštitují určitou část webové aplikace. Kořenový adresář obsahuje pouze soubory index.php, php.php, robots.txt a .htaccess, dále obsahuje tyto adresáře:

- template – obsahující soubory css a obrázky pro vzhled aplikace,
- core – obsahující soubory s jádrem aplikace, připojení k databázi atp,
- install – obsahuje soubory, které umí vytvořit tabulky a ostatní objekty v databázi a naplnit je daty nezbytnými pro funkci celého programu,
- pages – obsahuje přímo jednotlivé stránky aplikace.

## 3.7 Rozbor zabezpečení

### 3.7.1 Přístup uživatelů

Přístup k vnitřním funkcím je pro uživatele možný až po úspěšném přihlášení. V případě úspěšného přihlášení jsou veškeré údaje o uživateli uloženy v session proměnných (komunikace se serverem probíhá pomocí identifikace záznamu v cookies), a tak je zajištěno, že uživatel získá jen práva, která mu náleží (možnosti menu jsou mu přiřazovány na straně serveru).

### 3.7.2 SQL injection

Veškerá komunikace s databází je řešena s pomocí ovladače OCI8 a vnořeného injection scanneru bind. Použití této technologie zabraňuje využití techniky SQL injection k získání dat, která uživateli nenáleží.

### **3.7.3 Struktura databáze**

Struktura tabulek, jejich pojmenování, pojmenování řádků a sloupců a jejich proměnné jsou měněny jen při instalaci samotného systému. Po instalaci je možné smazat složku install – na funkci programu to nebude mít žádný vliv. Avšak zajistíme tím, že nebudou existovat příkazy měnící struktury databáze.

### **3.7.4 Mazání prvků v databázi**

Většina částí databáze je pomocí omezení (zejména FK) odolná proti mazání ze strany uživatelského prostředí. Samotné účetní operace a další údaje, které je třeba smazat, jsou mazány pouhým přidáním příznaku do tabulky. Tabulek, u kterých je mazání možné, je minimum a jejich smazání by nemělo mít vliv na účetní část programu.

### **3.7.5 Sledování manipulace s databází**

Zabezpečení proti nevhodnému zacházení s databází, proti finanční kriminalitě a zpronevěrám zařizuje kontrola nad tím, který uživatel provádí dané účetní operace. Tento údaj není možné měnit ani mazat a je povinnou položkou v tabulce.

V případě problémů je tedy vždy snadno možné dohledat, kdo se kterého prohřešku dopustil.

Metoda není odolná pouze proti získání a zneužití přihlašovacího jména a hesla.

### **3.7.6 Transakční režim**

Veškerá komunikace s databází probíhá pomocí transakčního režimu. Tímto je zajištěno, že v případě přerušení spojení nedojde k poškození integrity dat v databázi.

## **3.8 Ostatní funkce**

Webová aplikace obsahuje několik stěžejních metod, které nebyly popsány výše. Jde o funkce, bez kterých by program nebyl plně funkční.

### **3.8.1 Zadání počátečních stavů**

Po přihlášení do aplikace je možné nastavit počáteční stavy jednotlivých účtů. Tato metoda je nezbytná pro společnosti, které přechází z jiného ESS na systém Prefekt. Stejně tak je metoda použitelná při uzavření jednoho účetního roku a vytvoření účetního roku nového (avšak je to zbytečné, protože SW toto dokáže řešit sám).

### **3.8.2 Sledování stavů financí**

Dle zjištění je velmi častým problémem složitých ESS zobrazení aktuálních stavů hotovosti, nebo předpokládané sumy na účtech. Prefekt obsahuje velmi jednoduché postupy, kterými je schopný spočítat tyto sumy.

### **3.9 Testování**

Po úspěšném vytvoření aplikace proběhlo testování na několika cvičných datech. Pro testování posloužil program Easy PHP a lokální instalace serveru SQL Express.

#### **3.9.1 Test zatížením**

Na systém se v jeden okamžik připojilo 6 uživatelů a ti, nezávisle na sobě, prováděli účetní operace. Test skončil úspěšně. Žádný z uživatelů neměl žádné problémy a veškeré prováděné operace byly v systému zaevidovány.

#### **3.9.2 Test škozením**

Tento test se seskládá ze dvou částí. Snaha ohrožit či poškodit systém bez možnosti přihlášení (hádání hesel slovníkovou metodou, pokus o průraz do systému atp.) a pak snaha způsobit škody po přihlášení (mazání, získání cizích práv, získání nepříslušných informací atp.)

Test byl úspěšný s jednou výjimkou. Pokud se uživatel přihlásí s právy administrátora (tj. pokud je administrátorem), může způsobit velké škody smazáním všech uživatelských účtů a tím faktickou nedostupnost systému. I v tomto případě však veškeré účetní údaje zůstávají v bezpečí.

#### **3.9.3 Finanční testy**

Finanční testy byly prováděny tak, že byly vzaty účetní knihy společnosti Pozitron technology k.s. a jejich obsah byl přepsán do systému Prefekt. Sledováním, zda odpovídají hodnoty na skutečných závěrkách a závěrkách vytvořených systémem prefekt bylo testováno, zda si odpovídají veškeré hodnoty. Tento test byl úspěšný.

#### **3.9.4 Složité finanční testy**

Po poradě s účetní společností byla vytvořena sada účetních operací, které je pro běžné účetní systémy velmi obtížné zpracovávat. Šlo zejména o:

- splátky leasingu,
- nákup stroje, jeho přestavba a uvedení do provozu,

- výplata prvních dní nemocenské
- odvody na stavební spoření a důchodové pojištění při výplatě mzdy

Vzhledem k možnosti zvolit si účtový postup krok za krokem je možné zaúčtovat i tyto operace. Je však třeba základní znalosti účetnictví.

### **3.10 Zhodnocení projektu**

Prefekt je jednoduchý ESS systém, který obsahuje všechny funkce potřebné pro vedení podnikového účetnictví.

Velmi silný základ celému systému dává databáze, která je navržena kvalitně, s maximálním důrazem na jednoduchost a efektivitu.

Nad databází pracují PHP skripty, které spolu s HTML a CSS tvoří uživatelské rozhraní. Toto dělá systém ideální pro malé eshopy, živnostníky poskytující služby a další osoby, které nemají podnikání pevně vázáno na provozovnu.

Naopak systém není vhodný pro podniky, které jsou pevně vázány na provozovnu a mají časté malé obraty (např. prodejna potravin).

Díky svému jednoduchému a intuitivnímu ovládnutí si Prefekt jistě najde své uživatele, kteří nechtějí platit za ekonomické systémy, ani za služby daňových poradců či účetních.

#### **3.10.1 Možnosti budoucího rozvoje**

Pro úspěšné nasazení, nejen jako open source produktu, ale jako komerčního produktu, by bylo nezbytné vytvořit druhé uživatelské rozhraní v některém ze standardních programovacích jazyků (Delphi, Java, C++). Toto druhé prostředí by softwaru dalo velkou konkurenční výhodu. K systému by bylo možné přistupovat vzdáleně, z libovolného počítače, přes jednoduché rozhraní ve webovém prohlížeči. Stejně tak by však bylo možné mít několik robustních klientských instalací pro kanceláře a podobná místa, kde je vyžadován vyšší komfort práce. Navíc standardní aplikace mají větší důvěru než aplikace webové (přestože jde vždy o tytéž procesy, které obstarává databáze a liší se jen uživatelské prostředí).

## 4 Závěr

V této práci byl definován ekonomicko-správní systém a popsány jeho základní funkce. Systémy byly rozděleny dle několika kritérií. Bylo popsáno, v jakých případech je vhodné užití některých typů ESS. Byl popsán přínos systémů do společností, stejně jako rizika používání těchto systémů.

Pomocí kvantitativní metodiky zkoumání bylo vybráno několik produktů, které byly předmětem kvalitativního zkoumání. Během tohoto zkoumání byly zjištěny silné i slabé stránky konkrétních implementací ESS a tyto informace sloužily jako základ pro tvorbu vlastního systému.

Vlastní kvalitativní srovnání bylo zaměřeno nejen na samotný SW z uživatelského hlediska, ale také na prostředky, pomocí kterých byl tento SW vytvořen (zejména šlo o zkoumání programovacích jazyků a způsobu ukládání dat). Druhou částí pak byl průzkum vlastní marketingové prezentace těchto produktů. Zejména zjištění, že marketingová stránka je velmi slabá a umožňuje proniknutí na trh, změnilo cíl projektu. Z účetního systému psaného na míru jedné společnosti se tak stal základ open source projektu podnikového účetnictví.

Samotné tvorbě systému Prefekt předcházelo analytické bádání, které se skládalo z již zmíněného porovnání konkurenčních produktů a pohovoru s osobami, které mají vzdělání v oboru ekonomiky či účetnictví a zkušenosti s prací s těmito programy.

Po získání dostatečně specifického zadání proběhl výběr prostředků pro realizaci vlastních návrhů. Zveřejněné databáze zahraničních ESS byly pro použití nevhodné kvůli své složitosti (každá operace byla tvořena několika záznamy v tabulkách databáze), proto bylo nutno navrhnout databázi celou. K tomuto účelu posloužilo několik modelů, které celou práci velmi zjednodušily.

Výsledná implementace uživatelského rozhraní byla posledním krokem k vytvoření vlastního ESS systému.

Nový systém byl vystaven několika nezávislým testovacím provozům na reálných číslech (pro ověření funkce závěrek a podkladů pro podání daňových přiznání) a ve stávajícím stavu je plně funkční.

Budoucí rozvoj systému by bylo vhodné zaměřit na další zpříjemňování uživatelského rozhraní, popřípadě na tvorbu druhého uživatelského rozhraní určeného pro lokální provoz aplikace bez nutnosti použití interpretů.

## 5 Soupis bibliografických citací

Ministerstvo financí ČR. *Ares* [online]. 2009 [cit. 2010-04-10]. Dostupné z WWW: <<http://www.info.mfcr.cz/ares/ares.html>>.

*Programování* [online]. 2001 [cit. 2010-04-10]. Programovací jazyky. Dostupné z WWW: <<http://k-prog.wz.cz/progjaz/index.php>>.

ŽÁK, David. *Databazové systémy I - přednáška 1*. Pardubice, 2009. 42 s. Přednáška. Univerzita Pardubice.

ŽÁK, David. *Databazové systémy I - přednáška 1*. Pardubice, 2009. 42 s. Obrázek z přednášky. Univerzita Pardubice.

## **Příloha 1: Tvorba struktury tabulku**

Tato příloha obsahuje kód, který vygeneroval program Toad Data Modeler. Z generovaného výstupu je vložena pouze část, která generuje vlastní tabulky a klíče.

```
-- Create tables section
-----

-- Table Ucty
CREATE TABLE "Ucty"(
  "cislo" Integer NOT NULL,
  "nazev" Varchar2(50 CHAR) NOT NULL,
  "paktivni" Integer NOT NULL,
  "pbezny" Integer NOT NULL
)
/
-- Add keys for table Ucty
ALTER TABLE "Ucty" ADD CONSTRAINT "key_ucty" PRIMARY KEY ("cislo")
/
ALTER TABLE "Ucty" ADD CONSTRAINT "nazev" UNIQUE ("nazev")
/
-- Table Role
CREATE TABLE "Role"(
  "id_role" Integer NOT NULL,
  "nazevrole" Varchar2(15 CHAR)
)
/
-- Add keys for table Role
ALTER TABLE "Role" ADD CONSTRAINT "Key_role" PRIMARY KEY
("id_role")
/
-- Table Users
CREATE TABLE "Users"(
  "username" Varchar2(30 BYTE) NOT NULL,
  "password" Varchar2(30 BYTE) NOT NULL,
  "id_role" Integer,
  "id_osoby" Integer NOT NULL
)
/
-- Add keys for table Users
ALTER TABLE "Users" ADD CONSTRAINT "usernamePK" PRIMARY KEY
("username")
/
ALTER TABLE "Users" ADD CONSTRAINT "Key_Users" UNIQUE ("id_osoby")
/
```



```

-- Table Clovek
CREATE TABLE "Clovek"(
  "id_osoby" Integer NOT NULL,
  "hrubamzda" Real NOT NULL,
  "cislouctu" Varchar2(18 CHAR),
  "prijmeni" Varchar2(20 CHAR),
  "jmeno" Varchar2(12 CHAR),
  "cislouctu_kodbanky" Varchar2(4 CHAR)
)
/
-- Add keys for table Clovek
ALTER TABLE "Clovek" ADD CONSTRAINT "Key_clovek" PRIMARY KEY
("id_osoby")
/
-- Table HlavniKniha
CREATE TABLE "HlavniKniha"(
  "cislooperace" Integer NOT NULL,
  "cislodal" Integer,
  "cislomadati" Integer,
  "hodnotadph" Integer,
  "popis" Char(255 CHAR),
  "id_osoby" Integer NOT NULL,
  "datumzauctovani" Date NOT NULL,
  "castka" Real NOT NULL,
  "pStorno" Integer,
  "cislodokladu" Varchar2(15 CHAR)
)
/
-- Add keys for table HlavniKniha
ALTER TABLE "HlavniKniha" ADD CONSTRAINT "Key_HlavniKniha" PRIMARY
KEY ("cislooperace")
/
-- Table DPH
CREATE TABLE "DPH"(
  "hodnotadph" Integer NOT NULL,
  "nazev" Varchar2(40 CHAR)
)
/
-- Add keys for table DPH
ALTER TABLE "DPH" ADD CONSTRAINT "Key_DPH" PRIMARY KEY
("hodnotadph")
/
-- Table PomocnaKniha

```

```

CREATE TABLE "PomocnaKniha"(
  "cislooperace" Integer NOT NULL,
  "puhrazeno" Integer,
  "datumsplatnosti" Date,
  "soubor" Bfile
)
/
-- Add keys for table PomocnaKniha
ALTER TABLE "PomocnaKniha" ADD CONSTRAINT "Key2" PRIMARY KEY
("cislooperace")
/
-- Table Plat
CREATE TABLE "Plat"(
  "id_osoby" Integer NOT NULL,
  "mesic" Integer NOT NULL,
  "rok" Integer NOT NULL,
  "cislooperace" Integer,
  "zmenaplatu" Real
)
/
-- Add keys for table Plat
ALTER TABLE "Plat" ADD CONSTRAINT "Key_plat" PRIMARY KEY
("rok","id_osoby","mesic")
/
-- Table PrvekZaverky

CREATE TABLE "PrvekZaverky"(
  "cislooperace" Integer NOT NULL,
  "id_zaverky" Integer NOT NULL
)
/
-- Add keys for table PrvekZaverky
ALTER TABLE "PrvekZaverky" ADD CONSTRAINT "Key_PrvekZaverky"
PRIMARY KEY ("cislooperace","id_zaverky")
/
-- Table Zaverka
CREATE TABLE "Zaverka"(
  "id_typzaverky" Integer NOT NULL,
  "id_zaverky" Integer NOT NULL,
  "datumvytvoreni" Date,
  "pozn" Varchar2(255 CHAR)
)
/

```

```

-- Add keys for table Zaverka

ALTER TABLE "Zaverka" ADD CONSTRAINT "Key_zaverky" PRIMARY KEY
("id_zaverky")
/

-- Table typzaverky

CREATE TABLE "typzaverky"(
  "pojmenovani" Varchar2(30 CHAR),
  "id_typzaverky" Integer NOT NULL,
  "popravna" Integer
)
/

-- Add keys for table typzaverky

ALTER TABLE "typzaverky" ADD CONSTRAINT "Key_TypZaverky" PRIMARY
KEY ("id_typzaverky")
/

-- Create relationships section
-----
ALTER TABLE "Users" ADD CONSTRAINT "Role_User" FOREIGN KEY
("id_role") REFERENCES "Role" ("id_role")
/
ALTER TABLE "Users" ADD CONSTRAINT "Clovek_Users" FOREIGN KEY
("id_osoby") REFERENCES "Clovek" ("id_osoby")
/
ALTER TABLE "HlavniKniha" ADD CONSTRAINT "Ucty_HlavniKniha_MaDati"
FOREIGN KEY ("cislodal") REFERENCES "Ucty" ("cislo")
/
ALTER TABLE "HlavniKniha" ADD CONSTRAINT "Ucty_HlavniKniha_Dal"
FOREIGN KEY ("cislomadati") REFERENCES "Ucty" ("cislo")
/

ALTER TABLE "HlavniKniha" ADD CONSTRAINT "DPH_HlavniKniha" FOREIGN
KEY ("hodnotadph") REFERENCES "DPH" ("hodnotadph")
/
ALTER TABLE "HlavniKniha" ADD CONSTRAINT "User_HlavniKniha"
FOREIGN KEY ("id_osoby") REFERENCES "Users" ("id_osoby")
/
ALTER TABLE "PomocnaKniha" ADD CONSTRAINT
"HlavniKniha_PomocnaKniha" FOREIGN KEY ("cislooperace") REFERENCES

```

```
"HlavniKniha" ("cislooperace")
/
ALTER TABLE "Plat" ADD CONSTRAINT "Clovek_Plat" FOREIGN KEY
("id_osoby") REFERENCES "Clovek" ("id_osoby")
/
ALTER TABLE "Plat" ADD CONSTRAINT "HlavniKniha_Plat" FOREIGN KEY
("cislooperace") REFERENCES "HlavniKniha" ("cislooperace")
/
ALTER TABLE "Zaverka" ADD CONSTRAINT "TypZaverky_Zaverka" FOREIGN
KEY ("id_typzaverky") REFERENCES "typzaverky" ("id_typzaverky")
/
ALTER TABLE "PrvekZaverky" ADD CONSTRAINT
"HlavniKniha_PrvekZaverky" FOREIGN KEY ("cislooperace") REFERENCES
"HlavniKniha" ("cislooperace")
/
ALTER TABLE "PrvekZaverky" ADD CONSTRAINT "Zaverka_PrvekZaverky"
FOREIGN KEY ("id_zaverky") REFERENCES "Zaverka" ("id_zaverky")
```

## ***Příloha 2: Předplnění dat***

Kromě samotného vzhledu tabulek je pro případné tvůrce uživatelské přílohy nutné znát také základ předplněných dat. Data předplněná v této databázi reprezentují prvního vytvořeného uživatele s právy administrátora (je zřejmé, že jsou vytvořeny i jednotlivé uživatelské role) a nezbytně také základní strukturu účetní osnovy. Tuto osnovu lze libovolně doplňovat a měnit, pro potřeby malé společnosti však stačí neúplná účetní osnova vložená do systému v tomto případě.

```
-- Vyplnění prav
INSERT INTO Role VALUES ('1', 'Admin')
INSERT INTO Role VALUES ('2', 'Supervisor')
INSERT INTO Role VALUES ('3', 'Účetní')
INSERT INTO Role VALUES ('4', 'Hlavní účetní')
-- Vyplnění uzivatele
INSERT INTO Users VALUES ('admin', 'admin', '1', 'null')
-- Vyplnění uctu
INSERT INTO Ucty VALUES ('011', 'Zřizovací výdaje', '1', '1')
INSERT INTO Ucty VALUES ('013', 'Software', '1', '1')
INSERT INTO Ucty VALUES ('019', 'Ostatní dlouhodobý hmotný
majetek', '1', '1')

INSERT INTO Ucty VALUES ('111', 'Pořízení materiálu', '1', '1')
INSERT INTO Ucty VALUES ('112', 'Materiál na skladě', '1', '1')
INSERT INTO Ucty VALUES ('123', 'Výrobky', '1', '1')
INSERT INTO Ucty VALUES ('131', 'Pořízení zboží', '1', '1')
INSERT INTO Ucty VALUES ('132', 'Zboží na skladě a v prodejnách',
'1', '1')
INSERT INTO Ucty VALUES ('151', 'Poskytnuté zálohy na materiál',
'1', '1')
INSERT INTO Ucty VALUES ('153', 'Poskytnuté zálohy na zboží', '1',
'1')

INSERT INTO Ucty VALUES ('211', 'Peníze', '1', '1')
INSERT INTO Ucty VALUES ('221', 'Bankovní účty', '1', '1')
INSERT INTO Ucty VALUES ('261', 'Peníze na cestě', '1', '1')

INSERT INTO Ucty VALUES ('311', 'Odběratelé', '1', '1')
INSERT INTO Ucty VALUES ('321', 'Dodavatelé', '0', '1')
INSERT INTO Ucty VALUES ('324', 'Přijaté zálohy', '0', '1')
INSERT INTO Ucty VALUES ('343', 'Daň z přidané hodnoty', '0', '1')
INSERT INTO Ucty VALUES ('353', 'Pohledávky za upsaný základní
kapitál', '1', '1')

INSERT INTO Ucty VALUES ('411', 'Základní kapitál', '0', '1')
```

```

INSERT INTO Ucty VALUES ('419', 'Změny základního kapitálu', '0',
'1')
INSERT INTO Ucty VALUES ('428', 'Peníze v bance', '1', '1')

INSERT INTO Ucty VALUES ('501', 'Spotřeba materiálu', '1', '1')
INSERT INTO Ucty VALUES ('502', 'Spotřeba energie', '1', '1')
INSERT INTO Ucty VALUES ('504', 'Prodané zboží', '1', '1')
INSERT INTO Ucty VALUES ('513', 'Náklady na reprezentaci', '1',
'1')
INSERT INTO Ucty VALUES ('518', 'Ostatní služby', '1', '1')
INSERT INTO Ucty VALUES ('542', 'Prodaný materiál', '1', '1')

INSERT INTO Ucty VALUES ('601', 'Tržby za vlastní výrobky', '0',
'1')
INSERT INTO Ucty VALUES ('602', 'Tržby z prodeje služeb', '0',
'1')
INSERT INTO Ucty VALUES ('604', 'Tržby za zboží', '0', '1')
INSERT INTO Ucty VALUES ('642', 'Tržby z prodeje materiálu', '0',
'1')
INSERT INTO Ucty VALUES ('662', 'Úroky', '0', '1')

INSERT INTO Ucty VALUES ('701', 'Počáteční účet rozvažný', '2',
'0')
INSERT INTO Ucty VALUES ('702', 'Konečný účet rozvažný', '2', '0')
INSERT INTO Ucty VALUES ('710', 'Účet zisků a ztrát', '2', '0')

```

### ***Příloha 3: Příklady některých užítých databázových objektů***

Operace v hlavní knize jsou označeny svým vlastním číslem postupně za sebou tak, že první operace v roce má číslo 1 a každá bezprostředně následující operace číslo o 1 vyšší. O popsanou funkci se stará níže uvedený trigger ve spolupráci s níže uvedenou sekvencí.

```
create sequence SEKVENCE_HLAVNIKNIHA start with 1 increment by 1  
nomaxvalue;
```

```
create or replace trigger TRIGGER_HLAVNIKNIHA  
before insert on HLAVNI_KNIHA  
for each row  
begin  
select SEKVENCE_HLAVNIKNIHA.nextval into :new.cislooperace from  
dual;  
End;
```

Příkladem použití funkce je funkce, která vyhledává účetní obrat na jednotlivých účtech. Znalost tohoto obratu je nezbytná pro závěrky.

```
create or replace FUNCTION F_VRAT_CASTKU(idz IN NUMBER) RETURN  
NUMBER AS  
pocet INTEGER;  
BEGIN  
SELECT (SUM(sm.catka) - SUM(sp.castka)) INTO pocet  
FROM hlavnikniha sp, sklad sm  
WHERE sp.pStorno = '0' AND (sp.madati = idz OR sp.dal = idz) AND  
sm.pStorno = '1' AND (sm.madati = idz OR sm.dal = idz);  
RETURN pocet;  
END F_VRAT_PO CET;
```

Další dvě uvedené funkce již pracují přímo pro uživatele. Zobrazují počet neuhrazených mezd, resp. sumu, která je zaměstnanci nárokována, ale ještě nebyla zaměstnancům vyplacena.

```
create or replace FUNCTION F_VRAT_CASTKU(idz IN NUMBER) RETURN  
NUMBER AS  
pocet INTEGER;  
BEGIN  
SELECT (SUM(sm.catka) - SUM(sp.castka)) INTO pocet  
FROM hlavnikniha sp, sklad sm  
WHERE sp.pStorno = '0' AND (sp.madati = idz OR sp.dal = idz) AND  
sm.pStorno = '1' AND (sm.madati = idz OR sm.dal = idz);  
RETURN pocet;
```

```
END F_V RAT_PO CET;
```

```
create or replace function NeuhrazenePenizeMzdy (fact IN NUMBER)  
RETURN NUMBER  
AS pocet NUMBER;  
BEGIN  
select sum(zmenaplatsu)+sum(hrubamzda) into fact from plat join  
clovek on plat.idosoby = clovek.idosoby  
where cislooperace=null;  
RETURN fact;  
END;
```



#### ***Příloha č.4: Přehled selectů pro kontrolu manipulace s daty***

Jak bylo zmíněno v práci, účetní kniha je chráněna tak, aby každá účetní operace nesla údaj o tom, kdo ji provedl. Aby bylo možné těchto dat využít, existuje několik modifikací selectů, které s údaji pracují. Vstup z uživatelské úrovně je označen „DATA“.

Tento select vybírá všechny účetní operace, které provedl vybraný uživatel.

```
select username, idoperace, castka, madati, dal from hlanikniha,
uzivatele where hlavnikniha join uzivatel on hlavnikniha.username
= uzivatele.username where Upper(username) = Upper(DATA);
```

Tento select vybírá všechny hotovostní operace, které provedl vybraný uživatel.

```
select username, idoperace, castka, madati, dal from hlanikniha,
uzivatele where hlavnikniha join uzivatel on hlavnikniha.username
= uzivatele.username where Upper(username) = Upper(DATA) AND
(madati = 211 OR dal=211);
```

Tento select vybírá všechny bezhotovostní operace, které provedl vybraný uživatel.

```
select username, idoperace, castka, madati, dal from hlanikniha,
uzivatele where hlavnikniha join uzivatel on hlavnikniha.username
= uzivatele.username where Upper(username) = Upper(DATA) AND
(madati = 221 OR dal=221);
```

Tento select vybírá všechny operace k nimž neexistuje žádný doklad a které provedl vybraný uživatel

```
select username, idoperace, castka, madati, dal from hlanikniha,
uzivatele where hlavnikniha join uzivatel on hlavnikniha.username
= uzivatele.username where Upper(username) = Upper(DATA) AND
(cislodokladu=null);
```

Tento select vybírá počet operací k nimž neexistuje žádný doklad a které provedl vybraný uživatel

```
select username, count(castka), count(cislodokladu) where
hlavnikniha join uzivatel on hlavnikniha.username =
uzivatele.username where Upper(username) = Upper(DATA);
```

Tento select počítá sumu všech operací k nimž neexistuje žádný doklad a které provedl vybraný uživatel

```
select username, sum(castka) where hlavnikniha join uzivatel on
hlavnikniha.username = uzivatele.username where Upper(username) =
Upper(DATA) and cislodokladu=null;
```

Tento select vypíše všechny uživatele a počet jejich operací bez dokladů.

```
select distinct username, castka where hlavnikniha join uzivatel
on hlavnikniha.username = uzivatele.username where
cislodokladu=null;
```

Tento select vypíše všechny uživatele a sumu jejich operací bez dokladů.

```
select distinct username, sum(castka) where hlavnikniha join
uzivatel on hlavnikniha.username = uzivatele.username where
cislodokladu=null order by sum(castka);
```

## ***Příloha č.5: Ukázka CSS***

```
body {
    margin: 0;
    padding: 0;
    background: #FFFFFF url(images/img1.gif) repeat-x;
    text-align: justify;
    font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
    font-size: 11px;
    color: #8F8F8F;
}

h2, h3, h4, h5, h6 {
    margin: 0;
}

h2 {
    text-transform: lowercase;
    letter-spacing: -1px;
    font-size: 26px;
    font-weight: normal;
    color: #31576F;
}

h2 em {
    font-weight: bold;
    font-style: normal;
}

h3 {
    height: 25px;
    background: #0077A7 url(images/img3.gif) repeat-x left
bottom;
    padding: 13px 0 0 20px;
    text-transform: lowercase;
    font-size: 11px;
    color: #FFFFFF;
}

p, ol, ul {
    line-height: 180%;
}
```

```
ul {
    margin-left: 0;
    padding-left: 0;
    list-style-position: inside;
}

a {
    color: #6B6B6B;
}

a:hover {
    text-decoration: none;
}

.image {
    float: left;
    margin: 0 20px 0 0;
}
```

Ukázka ¼ kódu CSS. Kód je validní dle CCS 2.0.

## ***Příloha č.6: PHP funkce při přihlášení se do systému a odhlášení od něj***

Kód souboru login.php:

```
<?php
if(isset($_POST['logout']) && $_POST['logout']==true)
{
    $authorized=false;
    $_SESSION['authorized']==false;
    session_destroy();
    include 'main/odhlaska.php';
    include 'main/refresh.php';
}
else
{
    if(isset($_POST['login']) && $_POST['login']==true)
    {
        include 'main/connect.php';
        $stmt = oci_parse($connOci, 'SELECT LOGIN,HESLO,ROLE FROM
LOGIN WHERE LOGIN = :login');
/*$hodina=$_POST['login'];*/
        oci_bind_by_name($stmt,
':login',strtoupper($_POST['name']));
        oci_execute($stmt, OCI_DEFAULT);
        oci_fetch($stmt);

        if(strtoupper($_POST['name'])==oci_result($stmt, "LOGIN") &&
$_POST['password']==oci_result($stmt, "HESLO") && $_POST['name']!
='')
        {
            $_SESSION["role"]=oci_result($stmt, "ROLE");
            $_SESSION["login"]=oci_result($stmt, "LOGIN");
            $_SESSION["stranka"]="prihlasen";
            $authorized=true;
            $_SESSION['authorized']=true;
            /*echo '<h3>Přihlášen</h3>';*/
            include 'main/prihlaska.php';
            include 'main/refresh.php';
        }

    }
    else
    {
        echo '<p>Zadali jste špatné jméno nebo heslo,
přihlášení se nezdařilo!</p>';
    }
}
```

```

        echo '<a href="index.php">Zpět</a>';
        session_destroy();
    session_start();
}
}
else
{
    if($authorized==true)
    {
        //uživatel přihlášen
        include 'main/prihlaska.php';
        /*echo '<h3>Přihlášen</h3>';
        echo '<p>Přihlášen: '.$_SESSION["login"].NL.'</p>';
        echo '<p>Práva      : '.$_SESSION["role"].NL.'</p>';
        echo '<p>_____</p>';
        echo
        '<form action="index.php" method="post">'.NL.
        '<input type="hidden" name="logout" value="true">'.NL.
        '<input type="submit" value="Odhlásit">'.NL.
        '</form>'.NL;*/

    }
    else
    {
        //uživatel odhlášen

        include 'main/odhlaska.php';
        /*
        echo '<h3>Přihlásit</h3>';
        echo
        '<form action="index.php" method="post">'.NL.
        '<input type="hidden" name="login" value="true">'.NL.
        'Vaše údaje: <input type="text" name="name">'.NL.NL.
        '-jméno <input type="password" name="password">'.NL.
        '-heslo <input type="submit" value="Přihlaš">'.NL.NL;*/
    }
}
}
?>

```

Kód souboru prihlaska.php

```

<?php
    echo '<h3>Přihlášen</h3>';

```

```

echo '<p>Přihlášen: '.$_SESSION["login"].NL.'</p>';
echo '<p>Práva      : '.$_SESSION["role"].NL.'</p>';
echo '<p>_____</p>';
echo
'<form action="index.php" method="post">' .NL.
'<input type="hidden" name="logout" value="true">' .NL.
'<input type="submit" value="Odhlásit">' .NL.
'</form>' .NL;

```

?>

#### Kód souboru odhlaska.php

```

<?php
echo '<h3>Odhlášen</h3>';
echo
'<p>Zde se můžete přihlásit:</p>' .NL.
'<form action="index.php" method="post">' .NL.
'<input type="hidden" name="login" value="true">' .NL.
'<p>Jméno: <input type="text" name="name"></p>' .NL.
'<p>Heslo: <br /> <input type="password"
name="password"></p>' .NL.
'<input type="submit" value="Přihlaš">' .NL.NL;

```

?>

### ***Příloha č.7: Funkce pro připojení k databázi***

```
function connect($user, $pass, $sid)
{
$this -> connection = @oci_connect($user, $pass, $sid);
if (!$this -> connection) $this -> error = oci_error();
return $this -> connection;
}
```

```
function persistant_connect($user, $pass, $sid, $charset)
{
$this -> connection = @oci_pconnect($user, $pass, $sid, $charset);
if (!$this -> connection) $this -> error = oci_error();
return $this -> connection;
}
```

```
function disconnect()
{ if($this -> connection != 0)
{
if (!oci_close($this -> connection))
{
$this -> error = oci_error();
return 0;
}
else
{
$this -> connection = 0;
return 1;
}}
else return 0;
}
```



### ***Příloha č.8: Příklad ochrany proti SQL Injection***

SQL injection je zabráněno nepřímým přístupem k OCI knihovně. Funkce `oci_bind_by_name` nám povoluje zadání pouze proměnných a nikoli konstant, omezuje některé klíčové znaky a provádí další kontroly, které zabraňují SQL injection (funkce mimo jiné nabízí ochranu před některými dalšími útoky, ale právě SQL Injection je nejzákladnější).

```
$stmt = oci_parse($connOci, "select cislooperace, cislomadati,
cislodal,
popis, suma, cislodokladu, username from
hlavnikniha join users on hlavnikniha.id_osoby = users.id_osoby
where Upper(cislodokladu) LIKE :pod2 or upper(popis) = :pod1 or
to_char(cislooperace)= :pod3
order by nazev");
oci_bind_by_name($stmt, ':pod1', strtoupper($_POST['prom']));
oci_bind_by_name($stmt, ':pod2', strtoupper($prom));
oci_bind_by_name($stmt, ':pod3', $_POST['prom']);
```

### ***Příloha č.9: Ukázka některých dalších kódů***

Tento kód je automaticky vkládán do každé stránky. V případě, že uživatel není přihlášen, není mu umožněno dané stránky procházet.

```
<?php
if(isset($_SESSION['authorized']) &&
$_SESSION['authorized']==true)
{
Kód stránek
}
else
{
include '/pages/accessdenied.php';
} ?>
```

Tento kód zobrazuje část obsahu souboru config.php a ukazuje definici pomocné konstaty i některých proměnných. V souboru je mimo jiné uložen přístup k databázi.

```
<?php
//constants
define('NL', "\n");

//web configuration
global $page;
global $config, $sid, $authorized;
$config['db']
['server']='oci:dbname=//sql101.upceucebny.cz:1521/oracle10';
$config['db']['username']='CENSORED';
$config['db']['password']='CENSORED';
$config['db']['port']='1521';
$config['db']['sid']='xe';
?>
```