Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Student	:	**Jan Faltis**
Vedoucí bakalářské práce	:	**Doc. Ing. Jan Pour, CSc.**
Oponent bakalářské práce	:	**Ing. Lucie Poludvorná**

TÉMA BAKALÁŘSKÉ PRÁCE

Analýza a implementace informačního systému pro malou firmu

ROK : 2010

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracoval(a) samostatně a že jsem uvedl(a) všechny použité prameny a literaturu, ze kterých jsem čerpal(a).

V Praze dne 7.5.2010 …… ……

podpis

Poděkování:

Rád bych poděkoval Doc. Ing. Janu Pourovi, CSc. za cenné rady a připomínky a za odborné vedení při psaní mé bakalářské práce.

Abstrakt (CZ)

Bakalářská práce se zabývá analýzou požadavku, návrhem řešení a implementací první verze informačního systému pro malou společnost.

Úvodní část je především zaměřena na získání požadované funkcionality, která je dále přetvořena na návrh. Následuje zvolení podmínek pro implementaci a popis implementované verze systému. V závěru je pak celý průběh práce zhodnocen z hlediska společnosti, tvůrce a dosažené funkcionality.

Abstract (EN)

This B.A. thesis deals with the preliminary analysis of a request as well as design and implementation of the first version of a solution for a small business information system. The introductory part focuses on achieving the required functionality which, in turn, forms the basis for a proposal. What follows is the selection of implementation conditions and the description of the implemented version of the system. In conclusion, the whole workflow is evaluated from the points of view of the company, the creator and the functionality achieved.

Obsah

1. Úvod 8

1.1 Vymezení tématu práce a důvod výběru tématu 8
1.2 Cíle práce 8
1.3 Způsob/metoda dosažení cíle 9
1.4 Předpoklady a omezení práce 9
1.5 Struktura práce 9
1.6 Definice pojmů 9

2. Analýza situace Společnosti, získání požadovaných vlastností Systému 10

2.1 Charakteristika Společnosti 10
2.2 Požadavky společnosti na informační systém 10
2.2.1 Obecné požadavky 10
2.2.2 Upřesnění významu entit a návrh jejich atributů 12
2.2.3 Popis systému práv uživatelů Systému 20
2.3 Návrh příkladů použití a vybraných procesů 21
2.4 Pracovní závěry sekce 28

3. Tvorba návrhu řešení 30

3.1 Architektura Klient - Server 30
3.2 Volba úložiště dat 31
3.3 Platformy řešení 32
3.4 Návrh struktury databáze 35
3.4.1 Konceptuální návrh databáze 36
3.4.2 Fyzický model databáze 37
3.5 Návrh klientského programu 38
3.6 Pracovní závěry sekce 38

4. Implementace řešení 40

4.1 Metodika tvorby Systému 40
4.2 První verze Systému 40
4.2.1 Funkcinalita 40
4.2.2 Úložiště dat 41
4.2.3 Přihlašování 41
4.2.4 Konflikt práce se Systémem 41
4.3 Popis klientského programu 43
4.3.1 Hlavní okno 43
4.3.2 Okno práce se společnostmi 45
4.3.3 Okno práce s osobami 46
4.3.4 Okno práce s pracovními výkazy 47
4.3.5 Okno práce s procesy 48

5. Závěr 49

6. Terminologický slovník 50

7. Seznam použitých zdrojů 52

1. Úvod

1.1 Vymezení tématu práce a důvod výběru tématu

Cílem bakalářské práce je analyzovat požadavek, navrhnout a implementovat první verzi Systému pro podporu řízení a provozu malé společnosti. Daná společnost, pro kterou je práce vyvíjena bude dále v textu označována jako „Společnost“. Konkrétní vyvíjený systém bude dále označován jako “Systém”. Specifikace Systému budou dodávány zároveň s tvorbou, protože Společnost nemá zkušenost s podobnými systémy. Pro tuto práci jsem se rozhodl z několika důvodů. Prvním je potřeba daného Systému ve Společnosti a tedy i potencionální užitečnost práce. Druhým je možnost vypracování celého Systému ve spolupráci se Společností, konzultacemi a tedy vyzkoušení procesu tvorby informačního systému z mnoha pohledů a v různých fázích a z toho vycházející zkušenosti v této tématice, kterých se mi v dosavadních projektech nepodařilo dosáhnout. Dalším je problematika informačních systémů, kdy existuje velké množství robustních a velmi dobře pracujících systémů, které jsou ale zároveň velké a neodpovídají potřebám malé společnosti, kdy je důraz vložen na jednoduchost a přehlednost.

1.2 Cíle práce

Cílem práce je analyzovat situaci ve Společnosti a získat přehled funkcionalit požadovaných od Systému. Následně vytvoření návrhu Systému, respektujícího požadovanou funkcionalitu a nakonec implementace daného Systému ve Společnosti. Mým přínosem při řešení těchto cílu je analýza situace, kdy v závislosti na konzultacích se zástupcem Společnosti vytvořím funkcionální model a jeho popis. Zástupce společnosti je v textu zmiňován jako Společností vybraný pracovník, který je oprávněn formulovat požadavky na Systém a dále označován jako „Zástupce“ Následná transformace tohoto modelu do návrhu Systému. Zvolení vhodného způsobu implementace tohoto návrhu a implementace první verze Systému, respektujícího možnosti Společnosti.

1.3 Způsob/metoda dosažení cíle

Práce bude probíhat v několika fázích, první je analýza požadovaných vlastností poptávaného Systému, na základě kterých bude vytvořen funkcionální model Systému, zvolena metodika práce s ním a po odsouhlasení Zástupcem začne být vyvíjen samotný Systém. Vývoj systému bude postupovat inkrementálním způsobem, podle kombinace metodik „programuj a opravuj“ [BUCHALCEVOVÁ, 2009] a vybraných zásad extrémního programovaní [BECK, 2002], především spolupráce se zadavatelem a vývoj jen důležitých částí Systému. Metodika byla zvolena z důvodů malého počtu řešitelů, možnosti častých schůzek se Zástupcem a malé kritičnosti Systému ve smyslu toho, že situace je ve Společnosti zatím zvládána za pomocí jiných technologií a zároveň na systému nezávisí lidské životy, ani velká finanční ztráta.

1.4 Předpoklady a omezení práce

Systém je omezen po stránce finanční, jeho provoz nesmí být finančně náročný, a tedy není možné využívat drahé technologické, nebo implementační řešení kromě řešení odpovídajících standardům Společnosti.

1.5 Struktura práce

Práce se skládá z několika po sobě jdoucích fází:

- Analýza situace Společnosti, získání požadovaných vlastností Systému
- Návrh metodiky práce a modelu Systému
- Návrh implementace Systému
- Implementace Systému
- Závěr

Všechny fáze budou probíhat v těsné součinnosti se Společností.

1.6 Definice pojmů

Použité termíny jsou vysvětleny v terminologickém slovníku na konci práce, popřípadě přímo v textu, je-li to vhodné. V případě často používaného slova (např. Systém) je vysvětlen při prvním výskytu jeho význam a dále je slovo uváděno s velkým písmenem na začátku.

2. Analýza situace Společnosti, získání požadovaných vlastností Systému

2.1 Charakteristika Společnosti

Společnost se zabývá především dodávkou služeb a zboží souvisejících s informačními a komunikačními technologiemi "Informační a komunikační technologie jsou hardwarové a softwarové prostředky pro sběr, přenos, uchování, zpracování a poskytování informací a pro vzájemnou komunikaci lidí a technologických komponent IS" [Voříšek 2008]. Zaměření Společnosti je na individuální přístup k zákazníkům a řešení „na klíč". Společnost se řadí mezi malé společnosti a celkově zaměstnává do 15 lidí. Dosavadní správa, která bude nahrazena Systémem, probíhá za pomoci nástrojů MS Office a nepopsaných řešení.

2.2 Požadavky společnosti na informační systém

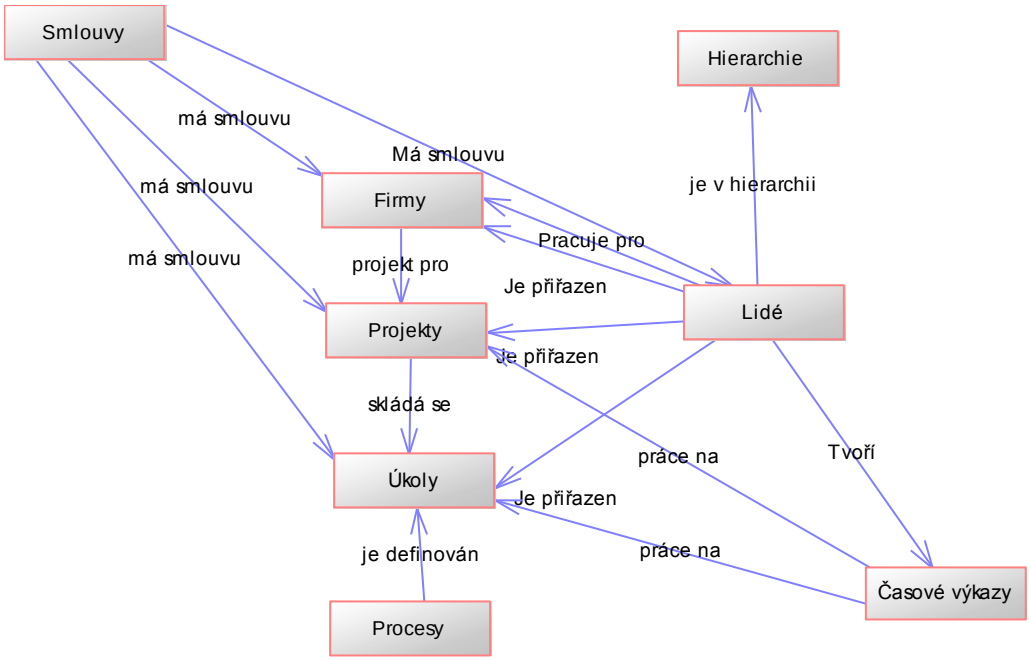
2.2.1 Obecné požadavky

Nejprve jsou v odražkách uvedeny základní vlastnosti požadovaného Systému, následuje detailnější rozbor. Tyto vlastnosti byly zjištěny na základě konzultací se Zástupcem a jsou tedy chápany jako zadání, kterého by mělo být dosaženo. Zadání bude dosahováno inkrementálním způsobem dle zvolené metodiky.

- Společnost potřebuje Systém pro evidenci kontaktů, projektů, úkolů, zaměstnanců a přidružených věcí.
- Jednoduchost, Systém bude přizpůsobený prostředí Společnosti. Samotná obsluha Systému bude intuitivní a rychlá, bez nutnosti složitého studia manuálů.
- Zabezpečení, Systém musí umožňovat jeho správu v závislosti na právech přihlášeného uživatele.
- Archivace, Systém musí archivovat změny v něm prováděné a být schopen tyto změny zobrazit.
- Flexibilita, Systém musí být schopen zvládat nedostatečně popsané úkoly, které nejsou detailně procesně popsány.
- Přenositelnost dat, Systém by měl být schopen exportovat data v podobné struktuře, která je používaná v MS Outlook, Nokia Ovi a rejstřících ISIR a ARES.
- Malá náročnost Systému na hardwarové vybavení.

Systém bude sloužit k evidenci veškerých kontaktů Společnosti, jak lidí, tak jiných společností, společnosti budou mít možnost označení na dodavatele a/nebo odběratele. Lidé by měli mít možnost přiřazení rolí (zaměstnanec, majitel) a příslušnosti k firmě. Zaměstnanci Společnosti budou zároveň uživatelé Systému a budou ho moci užívat podle svých práv. Vlastní zaměstnanci Společnosti budou organizováni dle hierarchie, která bude určovat zodpovědnost práce a práva vůči ostatním zaměstnancům Společnosti. Hierarchie by měla být vytvořena vzhledem k tendenci k projektovému řízení Společnosti, který vychází z jejího malého rozsahu "Projektová organizační struktura se vyznačuje tím, že napříč funkčními organizačními jednotkami jsou vytvářeny realizační týmy pro jednotlivé projekty. Projekty jsou výstupem strategického řízení, tudíž jsou realizovány za účelem dosažení stanoveného cíle. Problém nastává, když se do popředí zájmu manažerů dostává projekt a nikoliv jeho cíl." [VOŘÍŠEK, 2008]. Nebezpečí orientace na projekty, místo na cíle se Společnost vyhýbá opět svou velikostí a tím, že se neutvářejí projektové skupiny, kterým by šlo jen o vyřešení projektu, namísto například zisku. Navíc Společnost je díky své velikosti stále silně kontrolována svým vedením. Systém tuto kontrolu ulehčí a zpřesní. Ke klientům se budou vázat projekty a úkoly. Projekty se mohou skládat z více úkolů, ale nemusí mít žádný. Některé často prováděné úkoly budou zároveň procesy ve smyslu toho, že budou podrobně popsány a budou obsahovat firemní návod na vyřešení daného úkolu. Ke společnostem, projektům a úkolům budou evidovány uzavřené smlouvy. Systém bude umožňovat generování informačních výpisů o společnostech, lidech, projektech, úkolech a procesech, které budou dále specifikovány Zástupcem.

Na základě těchto specifikací byl vytvořen hrubý funkcionální model Systému (obrázek 2-1), zatím bez povinností a atributů, které budou definovány dále.



Obrázek 2-1 Konceptuální návrh systému

2.2.2 Upřesnění významu entit a návrh jejich atributů

Na základě dalších konzultací se Zástupcem jsem vytvořil návrh atributů jednotlivých entit z konceptuálního modelu, které zde jsou postupně rozepsány. K entitám je přidán další komentář ohledně jejich funkčnosti a jejich vazeb na další entity. V tomto stádiu se nejedná o databázový návrh, ale pouze o funkcionální. Tedy v samotném návrhu řešení a následné implementaci je předpokládán rozpad, nebo naopak integrace entit a doplnění dalších atributů. Všechny entity by také měly podléhat archivaci a mělo by být možno zobrazit změny na nich provedené.

- ***Entita Lidé***

  Entita představuje jednoho člověka, který bude dále používán v Systému při správě zaměstnanců a úkolů a zároveň slouží jako kontakty. Jedná se o klíčovou entitu celého systému s mnoha vazbami. Funkcionalita exportu je možná jako další rozšíření aplikace a předpokládá se jak export do normální aplikace MS Outlook, tak do aplikace MS Outlook Mobile, používané na mobilních telefonech vybavených systémem Windows Mobile. Atributy byly zvoleny tak, aby odpovídaly především aplikaci MS Outlook Mobile, ale zároveň byly některé nepodstatné vynechány.

Vazby na jiné entity:

- Vazba „je v hierarchii“ bude platit pro lidi pracující pro Společnost, kteří budou pomocí této vazby organizováni do hierarchie, která určí jejich práva a povinnosti při práci.
- Vazba „má smlouvu“ na entitu Smlouvy, která přiřazuje smlouvy, které se člověka týkají (smlouva o provedení práce atp.).
- Vazba „pracuje pro“ na entitu Firmy, která popisuje, že člověk je zaměstnancem dané firmy.
- Vazba „je přiřazen“ na entitu Firmy, která popisuje, že člověk je přiřazen jako řešitel práce pro danou firmu, tedy člověk může vytvářet, spravovat a prohlížet projekty a úkoly pro danou firmu.
- Vazba „je přiřazen“ na entitu Projekty, která popisuje, že člověk je přiřazen jako řešitel práce pro daný projekt, může tedy spravovat a prohlížet daný projekt a vytvářet, spravovat a prohlížet úkoly k němu přiřazené.
- Vazba „je přiřazen“ na entitu Úkoly, která popisuje, že člověk je přiřazen jako řešitel práce pro daný úkol, který může spravovat.
- Vazba „tvoří“ na entitu Časové výkazy, která popisuje, jak člověk trávil svůj čas.

Atributy:

Atributy převzaté z aplikace Outlook Mobile:

- Titul, atribut popisující tituly před jménem člověka.
- Jméno, atribut popisující první jméno člověka.
- 2. Jméno, atribut popisující druhé jméno člověka.
- Příjmení, atribut popisující příjmení člověka.
- Přezdívka, atribut popisující přezdívku člověka.
- Titul za, atribut popisující tituly za příjmením člověka.
- Obrázek, atribut přiřazující obrázek člověka.
- Společnost, atribut popisující společnost, ke které je člověk přiřazen.
- Oddělení, atribut popisující oddělení společnosti, ke které je člověk přiřazen.
- Funkce, atribut popisující funkci, kterou člověk v dané společnosti vykonává.
- Zobrazit, atribut popisující, jak má být kontakt na člověka zobrazován.
- Telefon zaměstnání., atribut popisující telefon do zaměstnání.

- Fax zaměstnání, atribut popisující fax do zaměstnání.
- Adresa zaměstnání, atribut popisující adresu do zaměstnání.
- Email, atribut popisující emailovou adresu člověka.
- Mobil, atribut popisující telefonní číslo na mobilní telefon člověka.
- Web stránka, atribut popisující webovou stránku člověka.
- Pracoviště, atribut popisující pracoviště člověka.
- Telefon domů, atribut popisující telefonní číslo domů.
- Adresa domů, atribut popisující adresu domů.
- Kategorie, atribut popisující kategorii, do které kontakt patří, mezi možnostmi by měly být alespoň „Dovolená, Osobní, Pracovní, Sezónní“, mělo by být umožněno přidávat další kategorie.
- Jiná adresa, atribut popisující další možnou adresu člověka.
- Asistent, atribut popisující asistenta člověka.
- Asistent telefon, atribut popisující telefon na asistenta.
- Manažer, atribut popisující přiřazeného manažera.
- Rodné číslo, atribut popisující rodné číslo člověka.
- Narozeniny, atribut popisující datum narození člověka.

V samotné aplikaci Outlook Mobile se vyskytují další atributy, které lze rozdělit na dvě skupiny:

1) Opakující se: tedy atributy, které jsou podobné již definovaným atributům, ale umožňují zapsat další možnost. Například se jedná o atributy email2, email3, telefonní číslo2, telefonní číslo3. Možnost jejich vyplnění by měla existovat, ale Systém by je neměl definovat explicitně, z důvodu nepřehlednosti velkého listu atributů. Například by tedy mělo být umožněno přidávat větší počet dalších emailových adres s různými názvy (email do práce, email pro formuláře atp.). Stejné chování by mělo být u atributů adres, telefonů a podobných.
2) Další, pro aplikaci irelevantní: tedy atributy, které mají smysl pro aplikaci Outlook, ale nemají reálný smysl pro Systém.

Atributy přiřazené kvůli funkcionalitě vyžadované Společností:

- Uživatel, atribut popisující, zdali je člověk uživatelem Systému.
- Login, atribut popisující jméno, pod kterým se uživatel přihlašuje do Systému.
- Heslo, atribut popisující heslo, kterým se uživatel přihlašuje do Systému.

- ***Entita Firmy***

Entita představuje jednu společnost, která je vedena v Systému. Entita umožňuje rozlišení na dodavatelskou a/nebo odběratelskou. Odběratelská společnost je ta, ke které se dále budou vázat projekty, a tedy se jedná o společnost, které je něco dodáváno. Dodavatelská společnost je společnost, z níž je možno přidělovat zaměstnance pro řešení projektů a úkolů, tedy samotná Společnost bude společností dodavatelskou. Entita bude svázána s lidmi, kdy jednotliví lidé budou přiřazeni ke společnosti, ve které pracují a zároveň budou moci být přiřazeni řešitelé, kteří spravují projekty pro společnost.

Vazby na jiné entity:

- Vazba „má smlouvu“ na entitu Smlouvy, popisující smlouvy vázající se k dané společnosti.
- Vazba „pracuje pro“ na entitu Lidé, již popsána výše.
- Vazba „je přiřazen“ na entitu Lidé, již popsána výše.
- Vazba „projekt pro“ na entitu Projekty, popisující, které projekty jsou přiřazeny a tedy vykonávány pro danou společnost.

Atributy:

Atributy převzaté z ISIR:

- Název, atribut popisující zkrácený název společnosti.
- Právní forma, atribut popisující právní formu společnosti.
- IČ, atribut popisující IČ společnosti.
- DIČ, atribut popisující DIČ společnosti.
- Adresa, atribut popisující adresu sídla společnosti.
- Jednatel, atribut popisující jednatele společnosti.
- Předmět podnikání, atribut popisující předmět podnikání společnosti.

Atributy přiřazené kvůli funkcionalitě vyžadované Společností:

- Dodavatel, atribut popisující, zdali je společnost dodavatelem.
- Odběratel, atribut popisující zdali je společnost odběratelem.

♦ *Entita Projekty*

Entita popisuje projekty řešené pro přiřazenou společnost. Projekt by zde měl být „chápaný jako jedinečný proces koordinovaných činností s daty zahájení a ukončení, prováděný pro dosažení cíle, vyhovující specifikovaným omezením v nákladech a zdrojích“ [ČSN / ISO 10 006]. Entita by měla mít možnost přidání více poznámek a případně dalších souborů.

Vazby na jiné entity:

- Vazba „má smlouvu“ na entitu Smlouvy, popisující smlouvy vázající se k danému projektu.
- Vazba „skládá se“ na entitu Úkoly, popisující úkoly podřízené tomuto projektu.
- Vazba „je přiřazen“ na entitu Lidé, již popsána výše.
- Vazba „práce na“ na entitu Časové výkazy, která popisuje, že na projektu byl stráven nějaký čas.

Atributy:

- Název projektu, atribut popisující název projektu.
- Datum začátku, atribut popisující datum začátku projektu.
- Datum konce, atribut popisující datum konce projektu.
- Cíl, atribut popisující cíle projektu.
- Poznámka, atribut popisující poznámky k projektu.
- Stav, atribut popisující stav projektu.
- Správce projektu, atribut popisující osobu zodpovědnou za průběh projektu.
- Kontaktní osoba odběratele, atribut popisující kontaktní osobu u odběratelské společnosti, která má být konzultována v rámci tohoto projektu.

- ***Entita Úkoly***

  Entita popisuje jednotlivé úkoly, ze kterých se skládají projekty. Entita je velmi podobná entitě Projekt, ale představuje menší díl práce, popřípadě již standardizovaný pracovní proces.

  Vazby na jiné entity:

  - Vazba „skládá se“ na entitu Projekty, již popsána výše.
  - Vazba „je definován“ na entitu Procesy, určující, že úkol je standardizovaným úkolem, který je popsán a přesně definován. Entita Procesy zde bude sloužit jako definice daného úkolu již s řešením.
  - Vazba „má smlouvu“ na entitu Smlouvy, popisující smlouvy vázající se k danému úkolu.
  - Vazba „je přiřazen“ na entitu Lidé, již popsána výše.
  - Vazba „práce na“ na entitu Časové výkazy, která popisuje, že na úkolu byl stráven nějaký čas.

  Atributy:

  - Popis, atribut popisující úkol.
  - Datum začátku, atribut popisující datum začátku úkolu.
  - Datum konce, atribut popisující datum konce úkolu.
  - Cíl, atribut popisující cíle úkolu.
  - Poznámka, atribut popisující poznámky k úkolu.
  - Stav, atribut popisující stav úkolu.
  - Správce projektu, atribut popisující osobu zodpovědnou za průběh úkolu.
  - Kontaktní osoba odběratele, atribut popisující kontaktní osobu u odběratelské společnosti, která má být konzultována v rámci tohoto úkolu.

- ***Entita Procesy***

  Entita představuje jeden proces, který byl již standardizován a tedy dobře popsán v rámci Společnosti. Proces by měl tedy sloužit k podpoře zaměstnanců při plnění úkolů, ke kterým je proces přiřazen. Zároveň slouží jako důležitý prvek kumulace znalostí Společnosti, kdy se snaží zachytit důležité informace, které se zrodily v neformalizované části informačního systému. “Informace, které nejsou dosud zaznamenány v databázi, jiné elektronické podobě a ani nejsou na žádném formuláři. Může se jednat o zkušenosti uložené v hlavách

zaměstnanců (obchodníků, konstruktérů, projektantů ale i výrobních dělníků a také manažerů), které nejsou využívané operativně v okamžiku potřeby a jsou předmětem managementu znalostí“ [BASL, 2008].

Vazby na jiné entity:

- Vazba „je definován“ na entitu Úkoly, již popsána výše.

Atributy:

- Název, atribut popisující název procesu.
- Trvání, atribut popisující standardizovanou dobu trvání procesu.
- Potřebné zdroje, atribut popisující potřebné zdroje k úspěšnému splnění procesu.
- Postup, atribut popisující předepsaný postup vedoucí k úspěšnému splnění procesu.
- Výstup, atribut popisující standardizovaný výstup procesu.
- Vstupní situace, atribut popisující vstupní situaci, na kterou by měl proces navazovat.
- Asistující, atribut popisující potřebné osoby, nutné ke splnění procesu (např. při montáži nového serveru je potřeba technik).

- ***Entita Smlouvy***

Entita představuje jednu smlouvu, která byla uzavřena a týká se nějakým způsobem Společnosti.

Vazby na jiné entity:

- Vazba „má smlouvu“ na entitu Lidé, již popsána výše.
- Vazba „má smlouvu“ na entitu Firmy, již popsána výše.
- Vazba „má smlouvu“ na entitu Projekty, již popsána výše.
- Vazba „má smlouvu“ na entitu Úkoly, již popsána výše.

Atributy:

- Název, atribut popisující název smlouvy.
- Datum podpisu, atribut popisující datum podpisu smlouvy.
- Datum platnosti, atribut popisující datum platnosti smlouvy.

- Text, atribut popisující vybraný důležitý text smlouvy.
- Soubor, atribut odkazující na naskenovaný soubor smlouvy.
- Kategorie, atribut popisující kategorii smlouvy (dohody, smlouvy o spolupráci atp.).

- ***Entita Časové výkazy***

Entita slouží pro ukládání časových výkazů zaměstnanců, kteří takto budou vykazovat svůj strávený pracovní čas. Pomocí entity bude možné zjišťovat, kolik času bylo stráveno na projektech a úkolech a zároveň kdy a co zaměstnanec dělal. Entita tedy bude sloužit jako důležitý nástroj účetnictví a managementu Společnosti a dá se u ní předpokládat další rozšíření.

Vazby na jiné entity:

- Vazba „tvoří“ na entitu Lidé, již popsána výše.
- Vazba „práce na“ na entitu Projekty, již popsána výše.
- Vazba „práce na“ na entitu Úkoly, již popsána výše.

Atributy:

- Datum, atribut určující rok, měsíc a den.
- Čas od, atribut určující časový okamžik, od kterého výkaz popisuje čas strávený přiřazeným zaměstnancem.
- Čas do, atribut určující časový okamžik, do kterého výkaz popisuje čas strávený přiřazeným zaměstnancem
- Popis, atribut určující popis daného výkazu.

- ***Entita Hierarchie***

Entita slouží pro popis hierarchie Společnosti. Entita tedy určuje vždy funkci, kterou má člověk ve Společnosti a jeho podřízené uživatele.

Vazby na jiné entity:

- Vazba „je v hierarchii“ na entitu Lidé, již popsána výše.

Atributy:

- Podřízení, atribut určující podřízené daného uživatele.
- Funkce, atribut určující funkci daného uživatele.

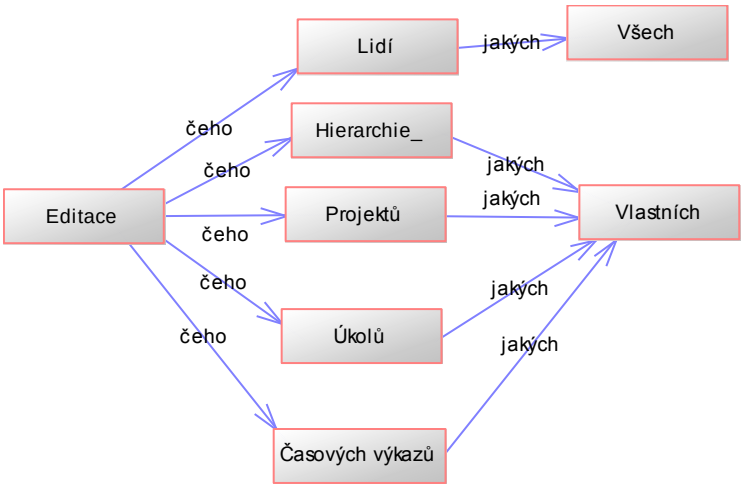
2.2.3 Popis systému práv uživatelů Systému

Uživatelé Systému budou mít přiřazena práva, která budou určovat jejich oprávnění pracovat se Systémem. Pro některé konkrétní skupiny práv budou vytvořeny role, tedy bude možno přiřadit zaměstnanci standardizovanou roli, která mu automaticky přiřadí odpovídající práva. Vedle práv práce s entitami, ještě budou práva pro práci v hierarchii. Ta budou ovlivňovat, koho může uživatel ovlivňovat a tedy například přiřazovat k projektům a úkolům. Uživatel může takto pracovat jen s lidmi jemu podřízenými, tedy uživatel nemůže měnit popis uživatele jemu nadřízeného, nesmí ho nikam přiřazovat a nesmí pracovat s jeho výkazy. Práva byla formulována na přání Zástupce. Práva by se dala rozdělit podle důležitosti takto:

1) Právo přiřazování práv, je nejvyšší možné právo určené vlastníkům Systému.

2) Právo vytváření a editace jednotlivých entit. Právo se následně bude dělit na jednotlivé entity a následně po nich na právo práce se všemi, nebo jen vlastními entitami. Tedy konkrétně uživatel může mít přiřazeno například právo editace úkolů omezené jen na vlastní úkoly, tedy úkoly, ve kterých figuroval. Právo editace automaticky přiřazuje právo čtení. Příklad přidělených práv zaměstnance, který se stará o některé projekty a zároveň může spravovat uživatele je znázorněn na obrázku 2-2.

3) Právo čtení jednotlivých entit. Právo se bude opět dělit na jednotlivé entity a následně po nich na právo práce se všemi, nebo jen vlastními entitami.

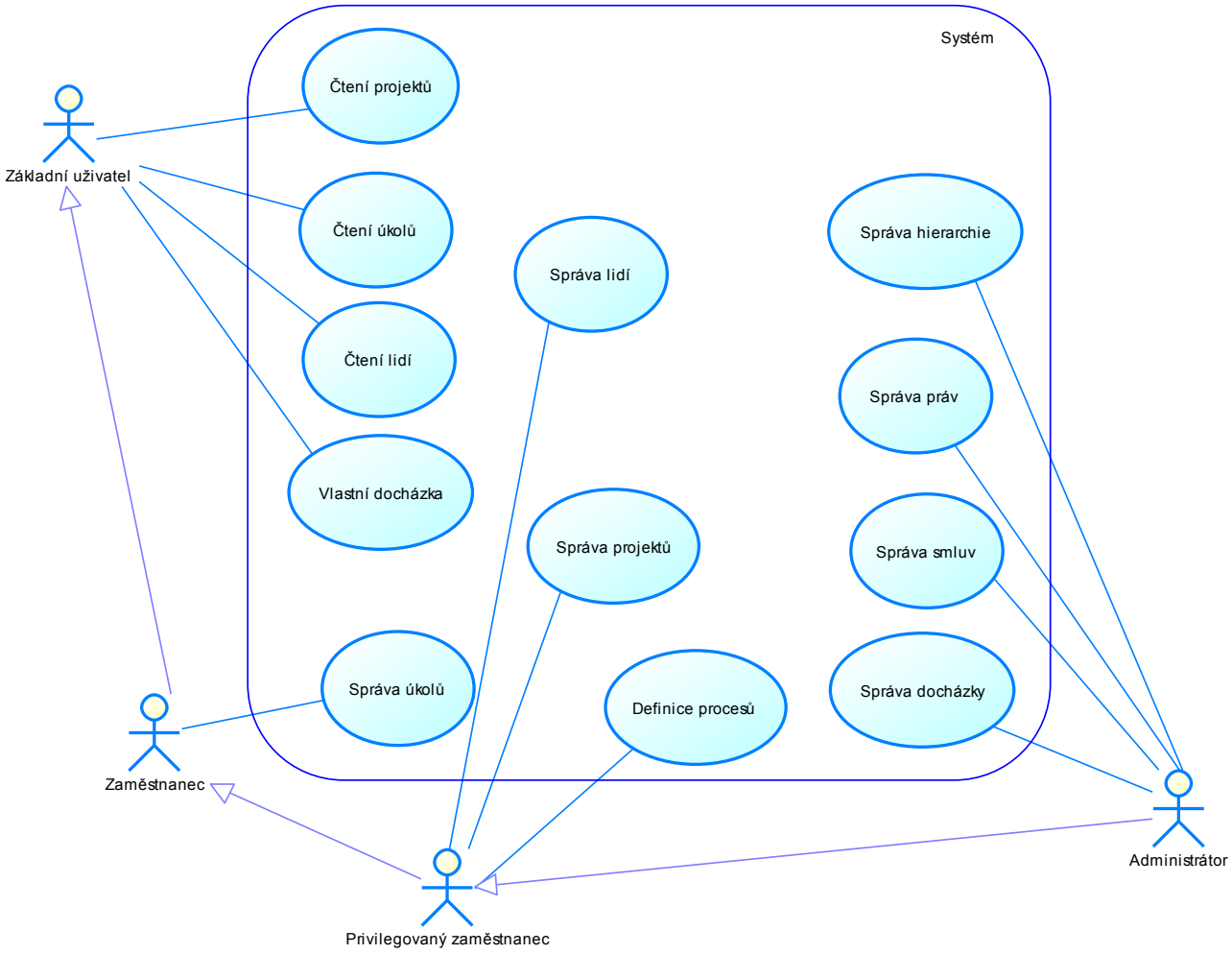


Obrázek 2-2 Příklad práv uživatele

2.3 Návrh příkladů použití a vybraných procesů

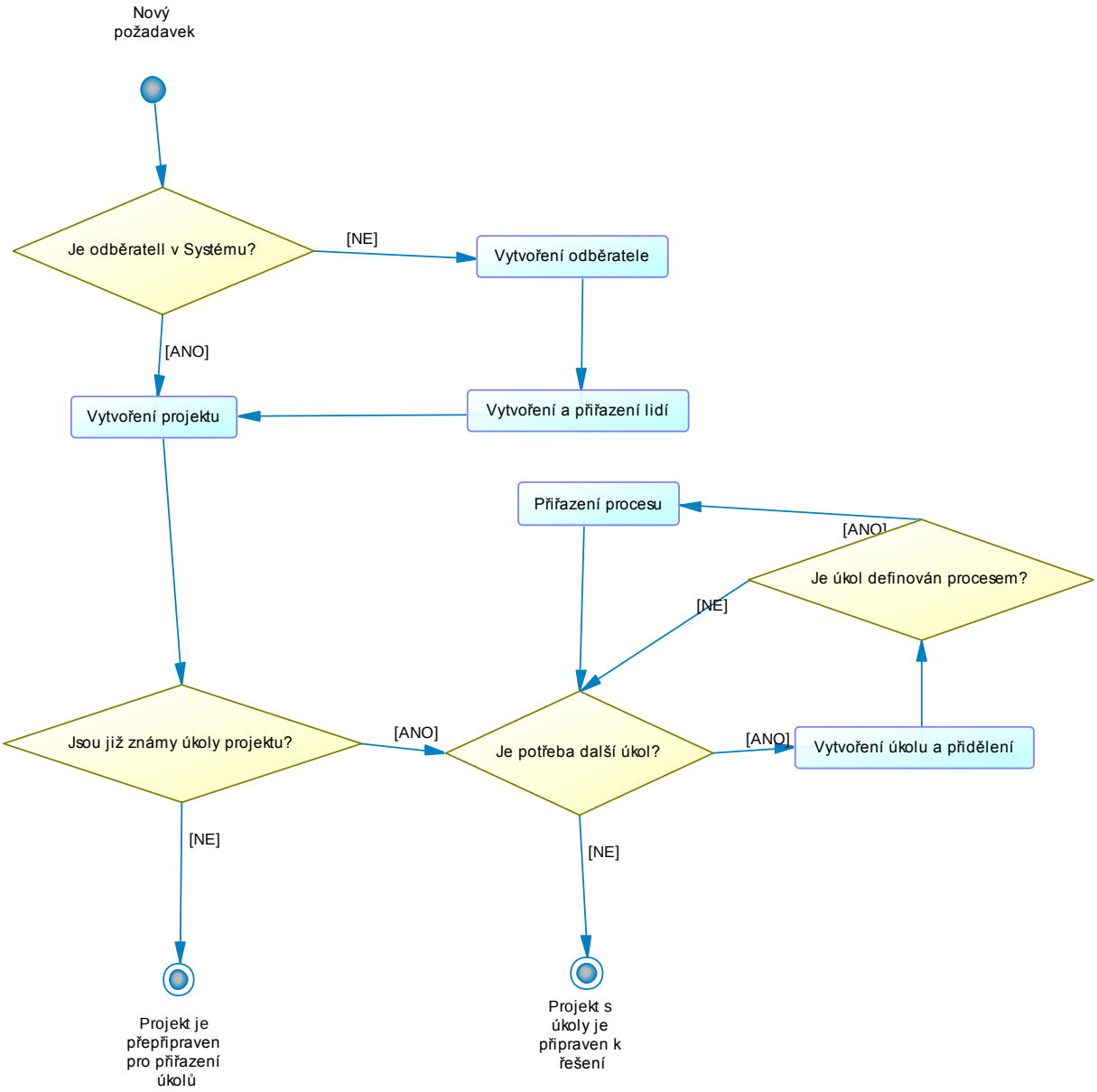
Na základě předchozích specifikací byl vytvořen use-case diagram (obrázek 2-3), popisující obecně možnosti práce se systémem. Vzhledem k požadavku na systém práv, ve kterém neexistují jednoznačné role, ale jsou přidělována práva konkrétním uživatelům, jsou aktéři Systému popsáni spíše rámcově, než vyčerpávajícím výčtem. Pro zpřehlednění diagramu byla použita dědičnost u typových uživatelů, tedy každý další potomek dědí již přidělená práva předka. Dědičnost byla použita k zpřehlednění celého diagramu. Uživatelé v diagramu by mohli být charakterizováni takto:

- Základní uživatel je uživatel, který má právo nahlížet na některé jemu přidělené části Systému, ale nesmí nic měnit. Jedná se o uživatele, který má ve Společnosti spíše pomocný charakter.
- Zaměstnanec je uživatel, který pracuje s úkoly.
- Privilegovaný zaměstnanec je zkušený zaměstnanec, který již může být zodpovědný za plnění projektů, správu lidí v Systému a díky svým zkušenostem může definovat procesy Systému.
- Administrátor je vlastníkem celého Systému, tedy například ředitel Společnosti. Má právo přiřazování práv a od toho odvozená jakákoliv práva práce se Systémem (může si je přidělit). Může spravovat hierarchii Systému, docházku, ze které si může vytvářet výpisy pro zlepšení informovanosti o vytížení zaměstnanců na jednotlivých projektech a úkolech, spravuje smlouvy, které jsou informačně přiřazené k odpovídajícím entitám.



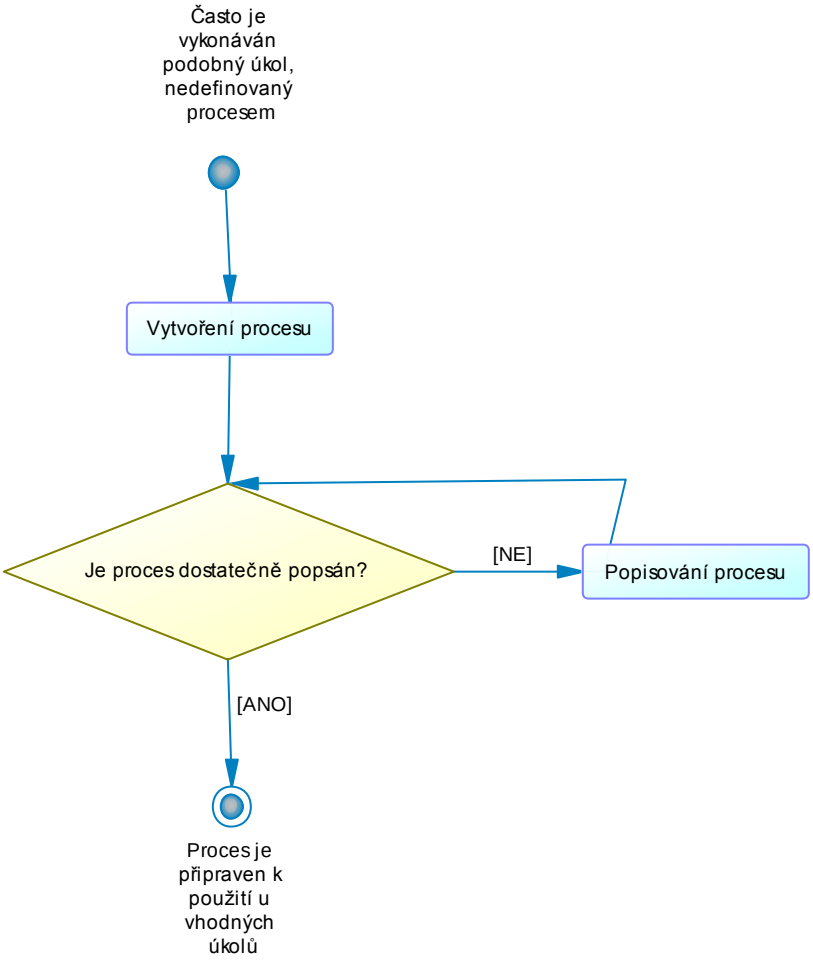
Obrázek 2-3 Use case diagram

Dále byl vytvořen konceptuální procesní model popisující postup při novém požadavku odběratele, vlastníkem procesu by v tomto případě mohl být uživatel s přiřazeným odběratelem, který je zodpovědný za projekty k němu přiřazené. Výstupem daného modelu je buď založený projekt s odpovídajícími úkoly a zaměstnanci, nebo předpřipravený projekt v případě, že úkoly potřebné k jeho řešení ještě nejsou známy. Na této úrovni se nedá předpokládat vysoká možnost standardizace, a tedy by se jednalo o unikátní a na přípravu a odbornost náročnou práci. Procesní model je na obrázku 2-4.



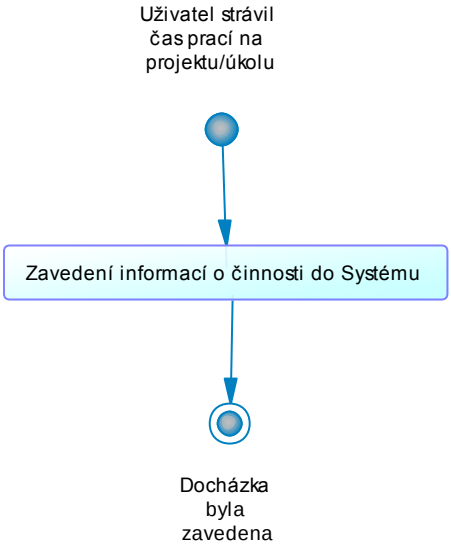
Obrázek 2-4 Procesní model popisující řešení nového požadavku

Dále byl vytvořen konceptuální procesní model popisující postup při častém opakování podobného úkolu. Vlastník by mohl být zkušený uživatel Systému, který je schopen dostatečně popsat problematiku a nalézt vhodné standardizované řešení. Výstupem je proces popisující často vykonávaný úkol. Proces je zobrazen na obrázku 2-5.



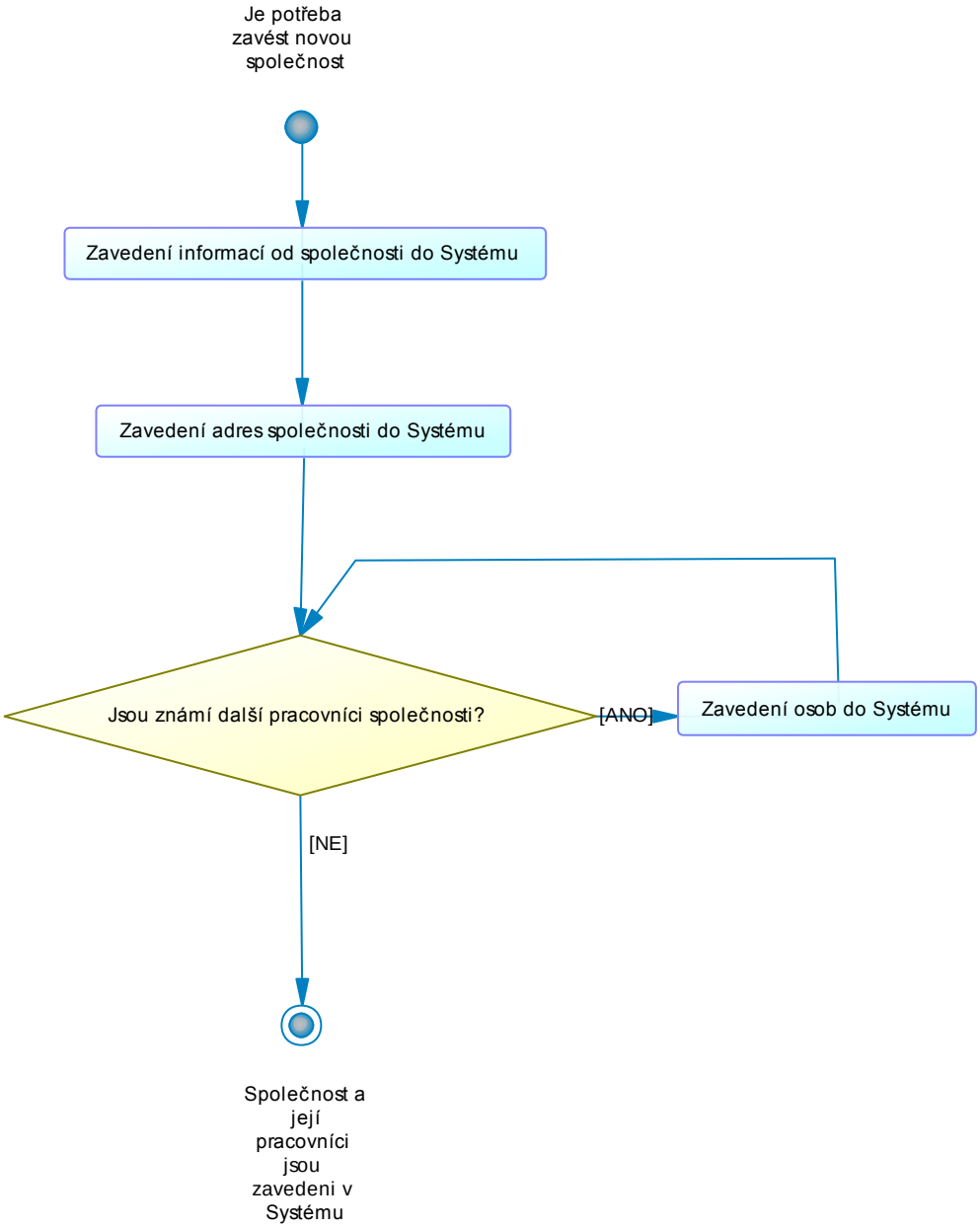
Obrázek 2-5 Vytvoření nového procesu

Dále byl vytvořen konceptuální procesní model popisující postup při evidenci docházky. Vlastník by mohl být jakýkoliv uživatel Systému, který má evidovat svou docházku. Výstupem je evidence daného úseku docházky. Proces je zobrazen na obrázku 2-6.



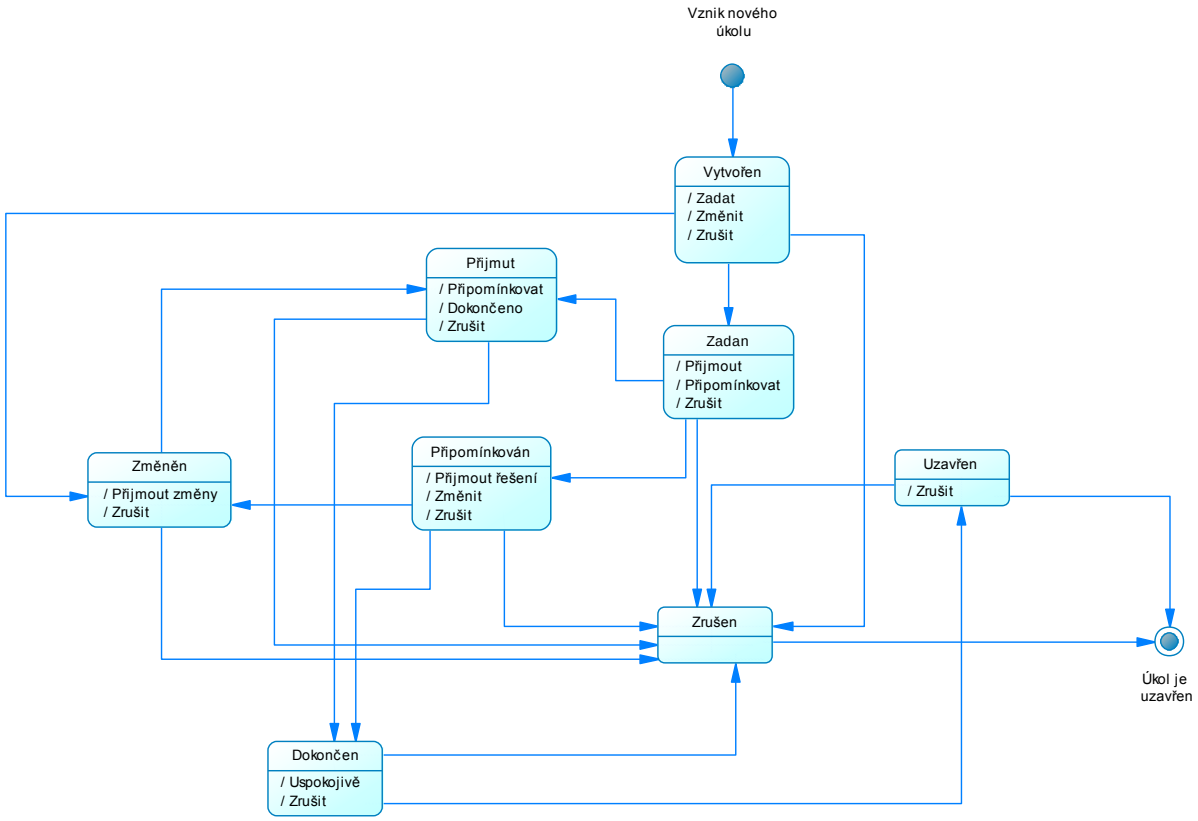
Obrázek 2-6 Evidence docházky

Dále byl vytvořen konceptuální procesní model popisující postup při zavádění nové společnosti a osob s ní svázaných, tedy pracovníků, do Systému. Vlastník by mohl být jakýkoliv uživatel Systému, který je zodpovědný za zavádění těchto dat. Výstupem je zavedení dané společnosti a osob s ní spjatých. Jedná se o extrémně důležitý proces, sloužící pro funkci sdílení kontaktů ve Společnosti. Proces je zobrazen na obrázku 2-7.



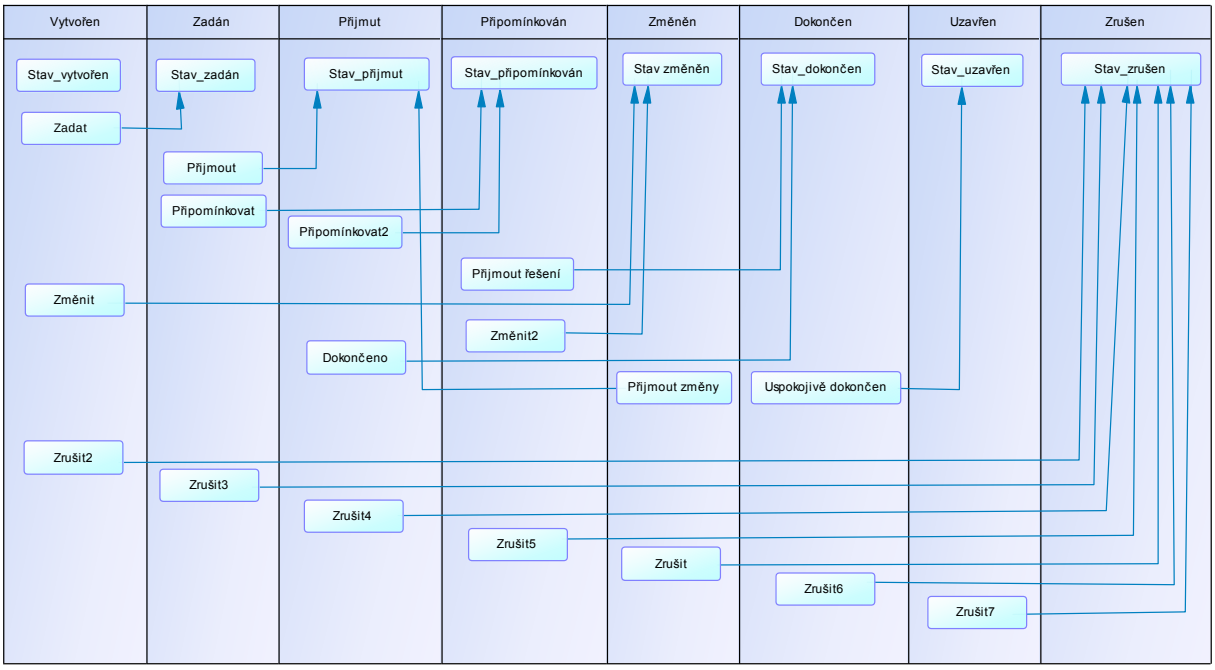
Obrázek 2-7 Zavedení společnosti do Systému

Dále byl vytvořen stavový diagram, ukazující přechod stavů úkolu a tak i jeho životní cyklus. Stav je vždy znázorněn „bublinkou“ a pod ní jsou za lomítky možnosti změny stavu. Diagram je na obrázku 2-8.



Obrázek 2-8 Stavový diagram

Vzhledem k ne zcela vhodné čitelnosti stavového diagramu jsem na přání Zástupce vytvořil i konceptuální model, ukazující možnosti úkolu měnit svůj stav (ač tento model není vhodný). Jednotlivé stavy jsou vždy znázorněny „plaveckými dráhami“. Vždy po změnění stavu je aktuální stav symbolizován kolonkou „stav_jméno“. Dále jsou vždy v odpovídající plavecké dráze uvedeny možné akce, které se s úkolem dají provést a následně šipka znázorňující do jakého stavu akce povede. Diagram je popsán obrázkem 2-9.



Obrázek 2-9 Diagram stavů úkolu

2.4 Pracovní závěry sekce

Tvorba části analýzy je velmi důležitá pro celý Systém z toho důvodu, že funkčně popisuje důležité vlastnosti. Tedy samotná implementace Systému se inkrementálně bude přibližovat funkcionalitě popsané v analýze. Čím více je analýza popisná, tím se snižuje riziko nepřesnosti a nepochopení požadavků Společnosti. Na druhou stranu zvolená metodika vycházející částečně z extrémního programování [BECK, 2002] vede k tomu, že by sekce neměla být popsána vyčerpávajícím způsobem a naprosto detailně, ale spíše konceptuálně. To z důvodu, že až samotná inkrementální implementace Systému ve spolupráci se Zadavatelem bude volit, které části Systému jsou nejdůležitější, která funkce je potřebná a která je nepodstatná. Důvody jsou poměrně hezky popsány v [BECK, 2002] a jsou to především „Směr projektu lze měnit. Strategie řízení projektu má větší cenu, může-li zákazník měnit požadavky během projektu. Čím častěji a razantněji lze požadavky měnit, tím lépe“ a „eliminace rizika směrem dolů“ ukazující, že pokud nevíme, jakou hodnotu by funkčnost měla, je vhodné počkat s její implementací.

Tvorba úkolů, projektů a procesů pomůže Společnosti v úsilí na zavedení normy ČSN/ISO 10006, Systémy managementu jakosti – Směrnice pro management jakosti projektů

[ČSN/ISO 10006], především v sekci 5 „Odpovědnost Managementu“, zabývající se procesy a řízením při tvorbě projektů a sekci 8 „Měření, analýza a zlepšování“.

Samotné procesy potom Společnosti umožní schraňovat informace neformalizované části informačního systému [BASL, 2008].

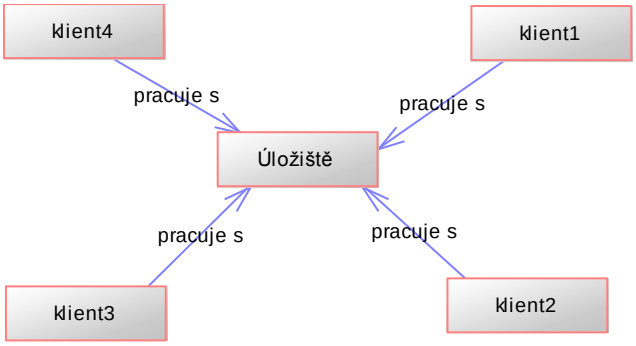
Při tvorbě diagramů jsem čerpal z publikací [ARLOW,NEUSTADT, 2008] zabývající se UML2 a [PAGE-JONES, 2001] zabývající se objektovým návrhem v UML.

3. Tvorba návrhu řešení

Na základě analýzy jsem vytvořil návrh řešení Systému. Při návrhu se objevilo několik variant různých možných řešení. Tyto návrhy byly porovnány a vhodné vybrány. U posouzení je vždy zdůvodněno, proč byl daný návrh vybrán jako nejvhodnější.

3.1 Architektura Klient - Server

Z požadavků Společnosti jednoznačně vyplývá nutnost jednotného úložiště dat, které je sdíleno mezi uživateli, odpovídající obrázku 3-1.



Obrázek 3-1 Architektura klient-server

Je tedy jasné, že úložiště dat musí být umístěno na serveru, na který se uživatelé Systému přihlašují a kde sdílejí svá data. Pro implementaci tedy bude zvolena architektura klient – server. Od zvolené architektury se odvíjí dvě pro nás použitelná řešení.

a) Tlustý klient a jednoduchý server

   Řešení předpokládá jen velmi lehkou serverovou část a „tlusté“ klientské programy, které se připojují, samy si žádají o data a samy zpracovávají business logiku. Největší výhodou řešení je jeho lehčí implementace, kdy není třeba řešit komunikaci mezi serverovou a klientskou částí a problémy z toho vycházející. Řešení je dále výhodné tím, že server nemusí být výkonný, není potřeba implementovat serverový program spravující logiku Systému a tato zátěž se tak přesouvá na klientské programy. Ze stejného důvodu vycházejí i nevýhody tohoto řešení, kterými je větší počet přenášených dat a nedokonalé zapouzdření Systému.

b) Lehký klient a složitý server

   Řešení předpokládá přesunutí co největší logické části na serverovou část Systému. Výhodou tohoto řešení je zapouzdření systému, kdy klientské programy jen posílají určité dotazy definované komunikačním protokolem a na základě těchto dotazů získávají již hotové odpovědi. Není tedy potřeba velká logická část v klientských programech. Toto řešení se zároveň také více přibližuje architektuře orientované na služby, protože funkce serverového programu jdou lehce nabídnout jako služby využitelné jinými stranami. Nevýhodou je větší náročnost na serverovou část Systému, která musí být schopna spravovat požadavky. Navíc je řešení mnohem složitější na implementaci a náročnější na čas.

Z těchto dvou řešení byla vybrána možnost „tlustého“ klientského programu a spolupracujícího úložiště dat. Logika Systému tedy bude realizována v klientské části programu. Řešení bylo vybráno z důvodu menší časové náročnosti. Vzhledem k omezenému času pro tvorbu Systému je preferováno rychlejší nasazení, před pozdějším. Řešení lze lehce převést na druhou možnost díky tomu, že logika Systému bude v případě převodu jen vyjmuta z klientského programu a vložena na server. Navíc zvolené řešení nevyžaduje pořízení nového fyzického serveru (nebo vyhrazení větší části jeho kapacity). Vlastnosti řešení jsou dobře zmíněny v [Distributed Application Architecture], což je část knihy zabývající se tvorbou programů na stránkách společnosti Sun. Zmíněná část se zabývá aplikační architekturou.

3.2 Volba úložiště dat

Z předcházející části tedy vyplývá nutnost pořízení takového úložiště dat, které dokáže pracovat na serveru a přijímat více uživatelů. Jako vhodné řešení byla vybrána SQL relační databáze z důvodů toho, že se jedná o vyzkoušený a robustní systém uchování dat. SQL standard je dobře popsán a struktura i data úložiště jsou relativně dobře přenositelná mezi různými konkrétními implementacemi (Oracle, MySql, Postgress), pokud nevyužívají mnoho funkcí konkrétních implementací, které jiné neumožňují. Relační databáze je dále extrémně lehce rozšiřitelná, na rozdíl např. od objektové, nebo relačně-objektové a dobře vyhovuje

transakčnímu přístupu, který bude v Systému využíván. Dále existuje mnoho vyzkoušených a spolehlivých databázových serverů, které lze použít a není třeba tak řešit tuto část Systému. Také existují vhodné nástroje pro design databáze a následný export kódu do SQL skriptu vhodného pro tvorbu databáze. Ostatní možnosti nejsou v této části uvedeny, protože se jedná povětšinou o možnosti výrazně méně vhodné pro Systém.

3.3 Platformy řešení

Z analýzy vyplývající architekturu klient – server lze implementovat pomocí mnoha platforem. Vzhledem k možnosti komunikace mezi různými platformami může být i každá část Systému implementována jinak, podle vlastností od ní očekávaných a zároveň platformou poskytovaných. V sekci bude rozebrána nejdříve klientská část a následně serverová část systému.

a) Klientská část systému

   Klientská část systému bude umožňovat samotnou práci se Systémem pro uživatele. Bude umět zobrazovat grafickou reprezentaci stavu entit popsaných výše a ovlivňovat je podle metody práce se Systémem. Obecně lze vhodné platformy rozdělit na dvě, podle způsobu práce s nimi. První jsou programovací jazyky vyšších generací jako Java, jazyky .NET, které umožňují snadnou tvorbu grafických aplikací. Druhou je tvorba klientské části systému pomocí skriptovacího jazyka PHP jako webového klienta, jehož výhodou je velmi snadná dostupnost odkudkoliv a zároveň ovladatelnost pomocí internetového prohlížeče, tedy odpadá nutnost použití samotného klientského programu.

   Zástupce v této části vyjádřil preferenci klientského programu, s tím, že v budoucnu by rád implementoval i klientskou část systému ovladatelnou pomocí internetového prohlížeče, byla tedy vybrána první možnost.

Z uvedených jazyků byly vybrány dva jazyky, mezi kterými jsme se rozhodovali:

- Java, výhodou tohoto jazyka je jeho nezávislost na operačním systému, dále to, že se jedná o čistě objektový jazyk, který podporuje moderní programovací trendy. Má vhodné nástroje pro připojení na databázi, je velmi otevřený a umožňuje modifikaci prakticky jakékoliv části již využívaných funkcí jazyka, umožňoval by jednoduchý přechod na architekturu „lehkého“ klienta, má vhodné nástroje pro dokumentaci programu a existuje velmi živá komunita, zabývající se řešeními pro tento jazyk. Mezi nevýhody patří neexistence vhodných a univerzálních řešení pro komunikaci především s programy MS Office a nutnost tvorby i poměrně jednoduchých komponent. Mezi subjektivní výhody bych uvedl mou zkušenost s tímto jazykem z výuky na VŠE a jiných projektů.
- Visual Basic 2008, dále v textu označovaný jako VB9, extrémní výhodou tohoto jazyka je jeho excelentní spolupráce s programy MS Office, velice snadná tvorba programu pomocí již hotových komponent, spolupráce s ostatními programy využívajícími platformu .NET a vhodnost tohoto systému pro systém Windows. Jako nevýhodu bych uvedl závislost na systému Windows a oproti Javě menší otevřenost prostředí Visual Studia, které je dodávaným prostředím pro jazyk.

Původně byl z dostupných jazyků vybrán jazyk Java, ale z důvodu změny zadání od Společnosti, která se rozhodla pro unifikaci používaných řešení na produkty společnosti Microsoft, bylo toto rozhodnutí změněno na jazyk VB9. Jednalo se však o poměrně velký problém, protože jsem se nikdy vývojem v daném jazyce nezabýval, ale vývoj v jazyce Java by neměl vzhledem k unifikaci řešení pro Společnost smysl.

b) Serverová část systému

Serverová část Systému by při zvolené architektuře měla být minimální a tedy obsahovat jen databázový server, umožňující připojení více uživatelů. Opět se objevilo více možností, které jsou dále popsány, a následně je zvolena nejvhodnější. Uvádím zde jen několik vhodných řešení, ač databázových serverů existuje velká řada a principielně lze použít skoro jakýkoliv. Každý popisovaný server má samozřejmě svou databázi, v dalším textu jsou povětšinou uváděny jako jedna komponenta.

- MySQL, jedná se o volně distribuovaný, robustní server s velmi dobrou dokumentací. K serveru existuje mnoho různých řešení administrace od prostředí realizovaného čistě PHP skriptem, přes doplňky vývojových prostředí, po standardní aplikace a případnou příkazovou řádku, umožňující připojení na server. Server má obecně dobrou podporu spolupráce ze strany dalších aplikací a nástrojů a jedná se o prověřené a časté řešení.
- MS SQL Server, velmi robustní server od společnosti Microsoft, jedná se o vhodnější řešení pokud je preferována jeho spolehlivost a excelentní spolupráce s ostatními programy a nástroji společnosti Microsoft. Jelikož je server komerční aplikací, je mnohem vhodnější u „kritičtějších“ aplikací, protože u jeho vývoje je dán větší důraz na jeho spolehlivost (a v nejhorším lze škodu vymáhat). K serveru také existuje velice dobrá podpora.
- PostgreSQL, opět volně distribuovaný, navíc open-source databázový server. Vyznačuje se tím, že má na rozdíl od ostatních poměrně velice striktní implementaci SQL, která se blíží standardu. Díky open-source má server tu výhodu, že je „otevřenější“ a lze lépe zjistit, jak jsou jeho konkrétní funkce implementovány a případně existuje určitá šance je změnit, tato vlastnost ale přichází i s rizikem toho, že bude nasazena chybná verze.

Pro serverovou část Systému byl původně vybrán MySQL server, ale následně byl změněn na žádost Zástupce na MS SQL server, pro jeho skvělou spolupráci se zvoleným programovacím jazykem klientské části a ostatními produkty firmy Microsoft. Tento přístup je zvláště výhodný, pokud si uvědomíme, že v případě dalšího vývoje mohou být určité funkce aplikací Microsoft změněny a pokud by byl Systém implementován pomocí např. jazyka Java a databáze MySQL, tak by musely být změněny i ty části Systému, které s nimi spolupracují. Navíc například pro čtení dokumentů MS office existují sice

v Javě vhodná komerční řešení, ale jejich licence je poměrně drahá a funkčnost je zaručena jen na stávající formáty. V případě implementace pomocí aplikací společnosti Microsoft je funkčnost pravděpodobně zaručena a i v případě natolik velkých změn, aby společnost Microsoft porušila pravidlo zpětné kompatibility, je velmi velká šance, že nutné změny budou lehce implementovány.

3.4 Návrh struktury databáze

Návrh struktury databáze se primárně řídí několika požadavky Společnosti na databáze jí využívané:

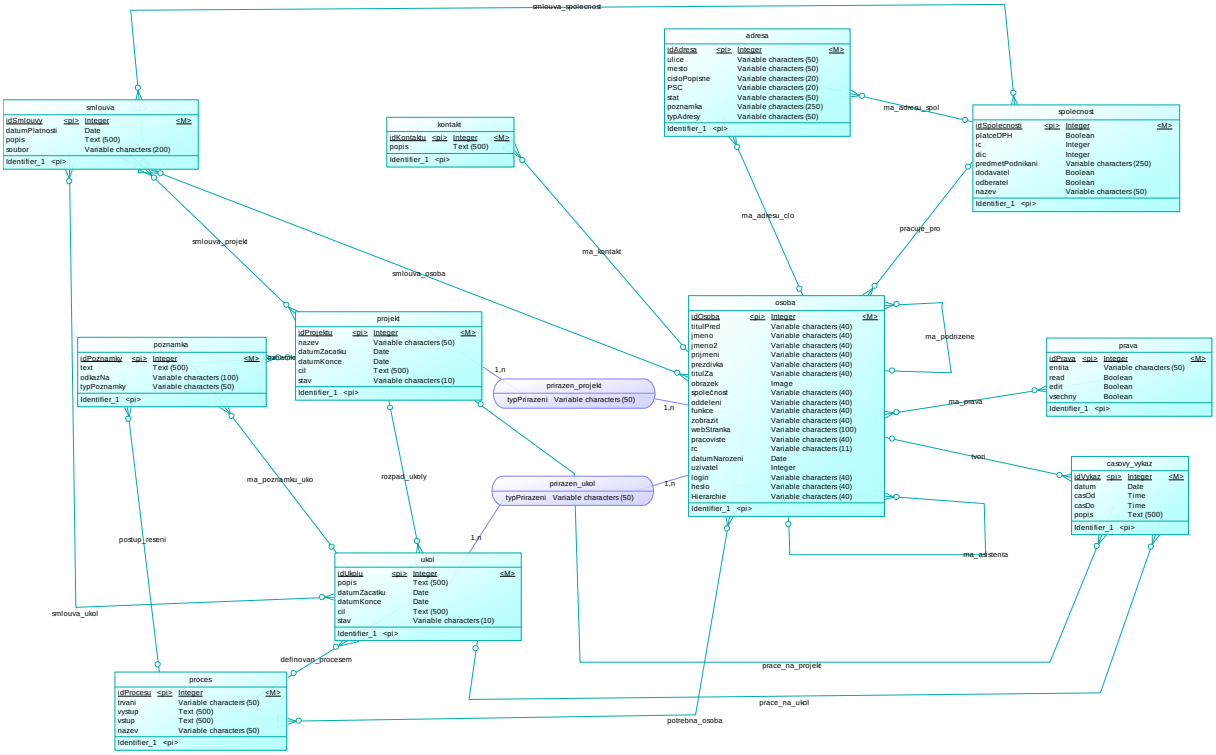
a) Jmenné konvence:
   - Jména tabulek začínají malým písmenem, a pokud jsou složená z více slov, tak každé další slovo následuje bez mezery a začíná velkým písmenem.
   - Nepoužívá se diakritika, všechny entity jsou bez háčků, čárek a jediným povoleným znakem vyjma písmen a čísel je podtržítko a pomlčka.
   - Jména atributů podléhají stejným pravidlům jako jména tabulek.
   - Jména indexů začínají vždy písmenem a pomlčkou „i-“, následuje popis indexu podle pravidel pro jména tabulek.
   - Jména relací mezi tabulkami začínají vždy malým písmenem, pokud se skládají z více slov je mezi nimi vždy podtržítko a další slova začínají vždy malým písmenem. V případě často používaného slova, které je ve smyslu toho, na co se relace váže, se toto slovo neskloňuje, tedy v případě mnoha relací na entitu „clovek“, je toto slovo v názvu relací neskloňované.
   - Každá tabulka má vytvořený umělý primární klíč, typu integer, který vždy začíná slovem „id“ a následuje jméno tabulky, začínající velkým písmenem.

b) Konvence databázových relací:
   - Relace nikdy nejsou povinné, tedy databáze může uložit vytvořenou entitu bez jakýchkoliv vazeb. Tato konvence vychází z nutnosti spolupráce a sdílení databáze mezi více programy a občas i mezi více společnostmi, při které se různí názory a přístup programátorů různých aplikací a snaží se udělat velmi benevolentní prostředí. To bohužel zvyšuje šanci na nekonzistenci dat.
   - Relace jsou vhodné, ale v určitých případech je lepší v zájmu jednoduchosti práce a menším nárokům na databázový server vhodné je nepoužít a raději přidat atribut odkazující na jinou tabulku, toto rozhodnutí musí být pečlivě zváženo a zdůvodněno.

c) Konvence návrhu
   - Návrh by neměl používat vlastní datové typy atributů, z důvodu jeho přenositelnosti a otevřenosti různým systémům, které by mohly mít s jejich interpretací problémy.
   - Návrh by měl být až na výjimky plynoucí z konvencí relační a neměl by obsahovat vlastnosti objektových databází, jako jsou pole a listy entit.

3.4.1 Konceptuální návrh databáze

Na základě požadavků na program, způsobu implementace a konvencí návrhu zmíněných výše byl vytvořen první konceptuální návrh databáze. Návrh byl odmítnut Zástupcem, z důvodu příliš vysoké složitosti a komplikované práce s ním, vyžadující pro získání informací komplexní složené SQL příkazy. Po konzultaci jsem tedy návrh přepracoval a vytvořil řešení maximalizující jednoduchost práce s databází. Druhý, zjednodušený návrh se nachází na obrázku 3-2.

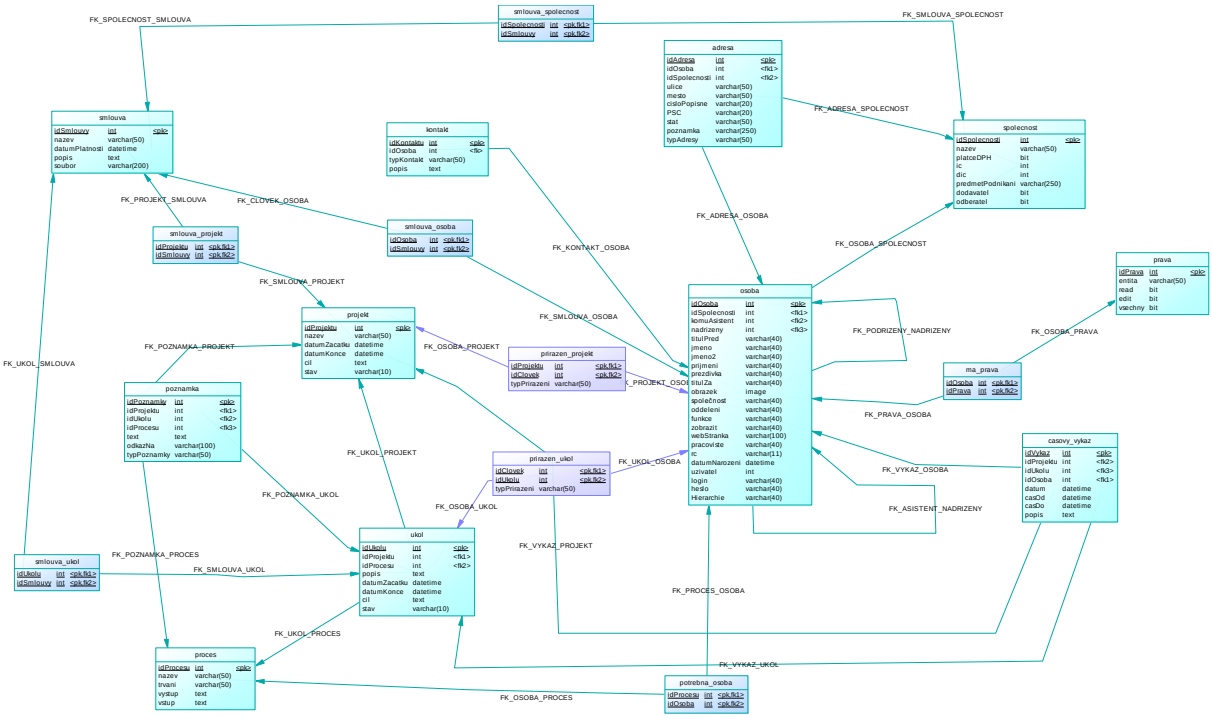


Obrázek 3-2 Návrh relační databáze

Vytvořený návrh odpovídá funkcionalitě od Systému požadované a zároveň odpovídá požadavkům a konvencím Společnosti pro tvorbu databází. Návrh je konceptuálním návrhem databáze vytvořeným v programu PowerDesigner verzi 12.5 a je návrhem ještě plně platformě nezávislým. Entity konceptuálního návrhu neodpovídají přesně popisu entit v části analýzy, ale splňují funkčnost od nich požadovaných.

3.4.2 Fyzický model databáze

Z konceptuálního návrhu byl pomocí programu PowerDesigner vygenerován fyzický model databáze pro MS SQL server 2005. Tento model je již tedy platformě závislý a je určen jen pro daný systém. Fyzický model byl upraven pro lepší použitelnost při práci s databází, konkrétně byly upraveny názvy relací mezi tabulkami. Relace byly přejmenovány podle vzoru „FK_“ + název tabulky, od které relace vede + “_” + název tabulky, ke které relace vede. V případě vazeb typu M:N konceptuálního návrhu, ze kterých vzniká ve fyzickém modelu vazební tabulka, platí, že název relací je odvozen od původních tabulek a ne vazebních. V modelu byla také upravena dodatečná pole tabulky „osoba“, sloužící k odkazování na nadřízené a asistenty tak, aby byl jejich význam zřetelnější. Fyzický model je na obrázku 3-3.



Obrázek 3-3 Fyzický model databáze

3.5 Návrh klientského programu

Klientský program byl vyvíjen s ohledem na moderní programovací techniky [PECINOVSKY, 2007]. Byla oddělena část datová a část zobrazení dat. Vzhledem k inkrementálnímu vývoji neexistuje žádný model popisující samotnou aplikaci, která je uzpůsobována podle požadavků Společnosti. Významným faktorem jsou také zásady extrémního programování [BECK, 2002] a to především jednoduchost a vhodnost řešení v poměru k úsilí nutnému pro jeho tvorbu, tedy aplikace je budována s otázkou „co je nejjednodušší řešení, které ještě bude fungovat“.

3.6 Pracovní závěry sekce

Zvolení dané architektury je velmi důležité před samotnou implementací, samotné zvolení nejvhodnějších prostředků (platforem) zaručuje nejlepší možný postup při vývoji Systému. Zde sice bylo toto rozhodnutí poměrně dáno přáním Společnosti pro použití nástrojů společnosti Microsoft, ale i tak bylo důležité. Bohužel toto rozhodnutí zapříčinilo poměrně pomalý vývoj Systému, z důvodu mé naprosté nezkušenosti se zvoleným serverem i jazykem. Odmítnutí změny a implementace v původně zvolených variantách se mi ale nezdálo vhodné,

protože Společnosti by ve výsledku nic nepřineslo a mě toto přizpůsobení umožnilo nahlédnout do prostředí .NET a začít se učit používat nové nástroje.

4. Implementace řešení

4.1 Metodika tvorby Systému

Systém bude tvořen hybridním způsobem, mezi metodikami Programuj a Opravuj a Inkrementální přírůstkovou metodikou, jak bylo popsáno výše. Tato variace vychází z toho, že Společnost samotná nemá dostatečné zkušenosti s používáním podobných systémů a lze tedy předpokládat, že první verze budou poměrně odlišné od ideální funkcionality. Metodika také čerpá z extrémního programování (viz výše) a jeho zásady budování toho, o co je v konkrétní čas konkrétní zájem. Tedy další verze budou vyvíjeny podle zkušeností z práce s nimi a tak přímo upravovány uživatelům. Jedná se o vhodnou metodu zpětné vazby. Jde také o to, že samotným nasazením nekončí životní cyklus Systému, ale Systém bude dodatečně vyvíjen a upravován podle připomínek z práce s ním vycházejících „Životní cyklus systému je časový úsek, který začíná úmyslem vytvořit systém a končí, když se software přestane používat“ [BUCHALCEVOVÁ, 2009]. Podobný přístup můžeme vidět například u pojišťoven, nebo bankovních systémů, které jsou vyvíjeny pro konkrétní banku interním IT týmem a které musí reagovat na změny jak legislativní, změny vycházející z požadavků businessu, tak změny vyžadované samotnými uživateli. Pro Systém budou především významné změny vycházející od jeho uživatelů, podle kterých bude dále formována jeho funkcionalita v jednotlivých přírůstcích.

4.2 První verze Systému

4.2.1 Funkcinalita

Podle návrhů řešení byla vytvořena první implementace Systému. Funkcionalita Systému v této verzi neodpovídá plně návrhu, to je způsobeno tím, že původně bylo počítáno s jazykem Java a první model byl vytvořen v jazyce Java. Společnost ale změnila požadavek na jazyk VB9, popřípadě jiný jazyk platformy .NET. Protože s touto platformou nemám zkušenosti, je verze minimalistická a její funkcionalita odpovídá tomu, co Společnost považuje za nejpřínosnější, tedy management společností, osob a jejich pracovních výkazů, projektů, úkolů a procesů.

4.2.2 Úložiště dat

Podle návrhu byl jako úložiště dat použit MS SQL server 2008. Na serveru byla podle návrhu vygenerována databáze a byli nastaveni uživatelé a jejich práva pro práci s ní. Problémem se v této části ukázalo časové zpoždění, se kterým byla Společnost schopna zajistit provoz SQL serveru, a to se opět promítlo do nižší funkcionality první verze klientského programu, který byl vyvíjen až proti funkčnímu serveru. V této části mi velmi pomohla publikace zabývající se problematikou MS SQL serveru 2005 [STANEK, 2006].

4.2.3 Přihlašování

Dále se objevily nejasnosti ve způsobu přihlašování do Systému. Přihlašování závisí na přihlašování k SQL serveru a Společnost se zatím nerozhodla, jestli bude používat při všech svých řešeních přihlašování pomocí účtu Windows, nebo bude přihlášení záviset na uživatelských účtech SQL serveru, nebo by mělo záviset jen na osobách a jejich přihlašovacích údajích v databázi. Vzhledem k tomu, že Společnost chce zavést jednotnou politiku přihlašování ve svých řešeních, tak samotné přihlašování nebylo v Systému zatím zavedeno a používá se přístup jednoho uživatele databáze. Toto řešení je samo o sobě neuspokojivé a jen dočasné z důvodu toho, že jakmile bude politika přihlašování jednotná, bude implementováno přihlašování jí odpovídající. Rád bych zde zmínil přínos pro Společnost, která se začala zabývat tématikou unifikace přihlašování do aplikací.

4.2.4 Konflikt práce se Systémem

Vzhledem k tomu, že Systém umožňuje práci více uživatelů najednou a klientský program se k datovému úložišti připojuje jen při načítání a úpravě dat a jinak pracuje nad daty načtenými z databáze, ale uloženými v dočasných strukturách, může nastat konflikt a nekonzistence dat. Konflikt může nastat například ve chvíli, kdy uživatel A načte data a pracuje nad nimi, zatímco uživatel B je změní. Když by uživatel A chtěl data uložit do úložiště, mohla by nastat nekonzistence a případně by byla ztracena změna uživatele B. K vyřešení tohoto problému byla zvažována dvě řešení. Problematika je podrobněji vysvětlena v jakékoliv publikaci věnující se práci

s databázemi, popřípadě objektově-relačnímu mapování, probíhajícímu mezi databázemi a objektově orientovanými programy a považuji za zbytečné ji tady více rozebírat.

a) Pesimistické zamykání záznamů

   Funguje tak, že záznamy zamčené jedním uživatelem jsou neměnné, dokud on svou práci s nimi neskončí a zámek neuvolní. Řešení bylo odmítnuto jako nevhodné z důvodu toho, že by umožňovalo vznik dlouho uzamčených dat jedním uživatelem a znemožňovalo práci dalším. S přibývající funkcionalitou Systému by se také mohla objevit možnost „dead-locku“ tedy mrtvého zámku, kdy jeden uživatel čeká na uvolnění záznamů, které potřebuje k dokončení své práce, zatímco uživatel druhý, který má dané záznamy zamčeny čeká na uživatele prvního, až uvolní záznamy, které potřebuje k dokončení práce své.

b) Optimistické zamykání záznamů

   Funguje tak, že uživatelé zamykají záznamy jen ve chvíli samotného zápisu do úložiště. Tedy při normální práci se Systémem si záznamy načtou a zámek trvá co nejkratší moment až ve chvíli jejich ukládání. V případě, že záznamy byly mezitím změněny, jsou právě ukládané změny odmítnuty a záznamy jsou znovu načteny. To zabraňuje nekonzistencím v úložišti a ztrátě změn a dat.

Bylo vybráno řešení optimistických zámků z důvodu vyloučení existence dead-locku a malé šance existence konfliktu.

4.3 Popis klientského programu

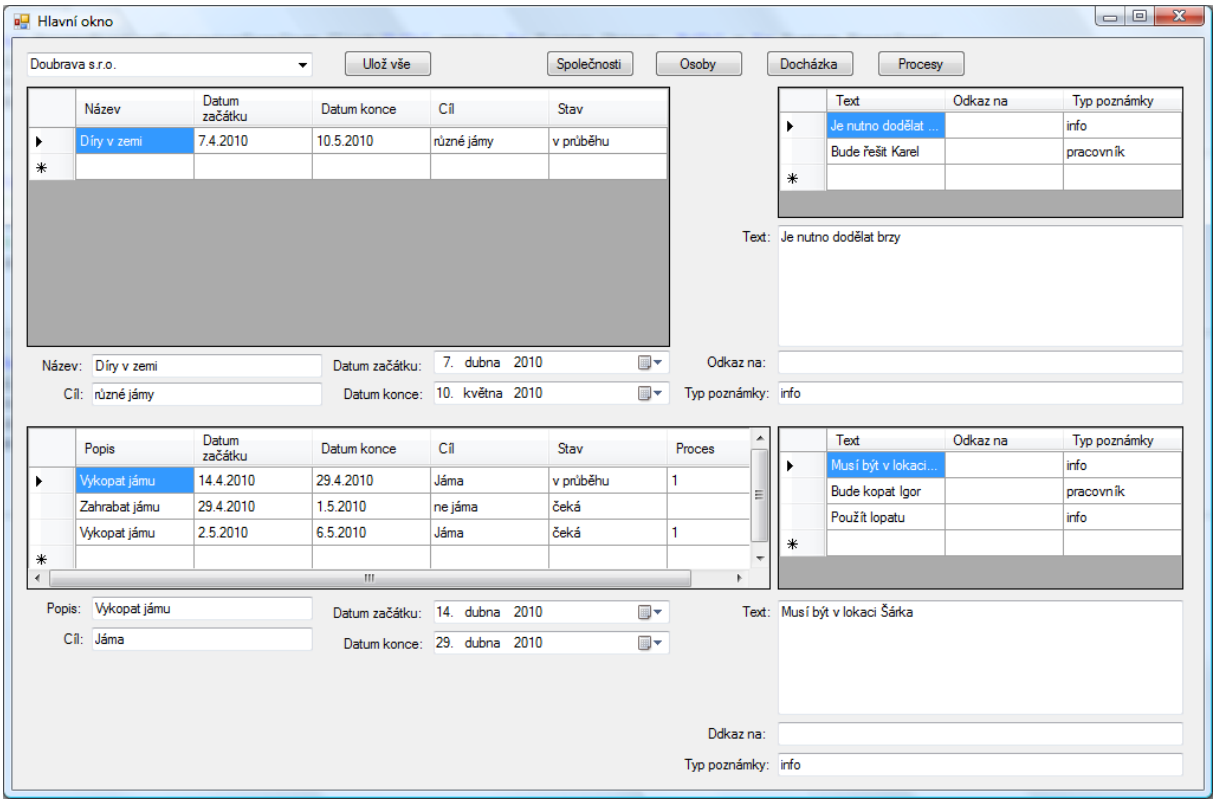
Při tvorbě klientského programu jsem silně čerpal z publikací [HALVORSON, 2008] zabývající se vývojem ve VB9, [FUCHS,BARCHFELD, 2010] zabývající se vhodnými řešeními ve VB9 a [HAMILTON, 2008] zabývající se platformou ADO.NET 3.5.

4.3.1 Hlavní okno

Hlavní okno je výchozím oknem celé aplikace. Slouží pro přechod k dalším oknům a je v něm umístěna správa projektů, úkolů a poznámek k nim svázaných. Okno je popsáno po jednotlivých logických částech.

a) Volba společnosti, je ovládací combo box umístěný v levém horním rohu. Slouží k volbě společnosti, jejíž projekty chceme zobrazit.
b) Tabulka projektů, je tabulka umístěna hned pod volbou společnosti v levém horní části okna. Tabulka slouží k volbě projektu dané společnosti, kterým se chceme zabývat, k zakládání projektů a k jejich editaci. Pro snadnější editaci jsou pod ní umístěny ještě prvky obsahující název, cíl, datum začátku a datum konce projektu (časová data se dají velmi snadno zvolit pomocí rozbalovacího kalendáře).
c) Tabulka poznámek vázajících se k projektům je umístěna vedle tabulky projektů a obsahuje poznámky právě zvoleného projektu. Tabulka slouží ke zvolení poznámky a její následné editaci, popřípadě vytvoření nové. Pro snadnější editaci jsou pod ní umístěny ještě prvky obsahující text, odkaz a typ dané poznámky.
d) Tabulka úkolu vázajících se k projektům je umístěna pod tabulkou projektů a obsahuje úkoly právě zvoleného projektu. Tabulka slouží ke zvolení úkolu a jeho editaci, popřípadě vytvoření nového. Pro snadnější editaci jsou pod ní umístěny ještě prvky obsahující popis, cíl, datum začátku a datum konce úkolu.
e) Tabulka poznámek vázajících se k úkolům, je umístěna vedle tabulky úkolů a obsahuje poznámky právě zvoleného úkolu. Tabulka slouží ke zvolení poznámky a její následné editaci, popřípadě vytvoření nové. Pro snadnější editaci jsou pod ní umístěny ještě prvky obsahující text, odkaz a typ dané poznámky.
f) Ovládací tlačítka sloužící pro přechod na další okna aplikace se nacházejí v horní střední části okna, hned vedle ovládacího combo boxu společností. Slouží pro otevření a práci s dalšími okny, která budou popsána.

Okno se nachází na obrázku 4-1.



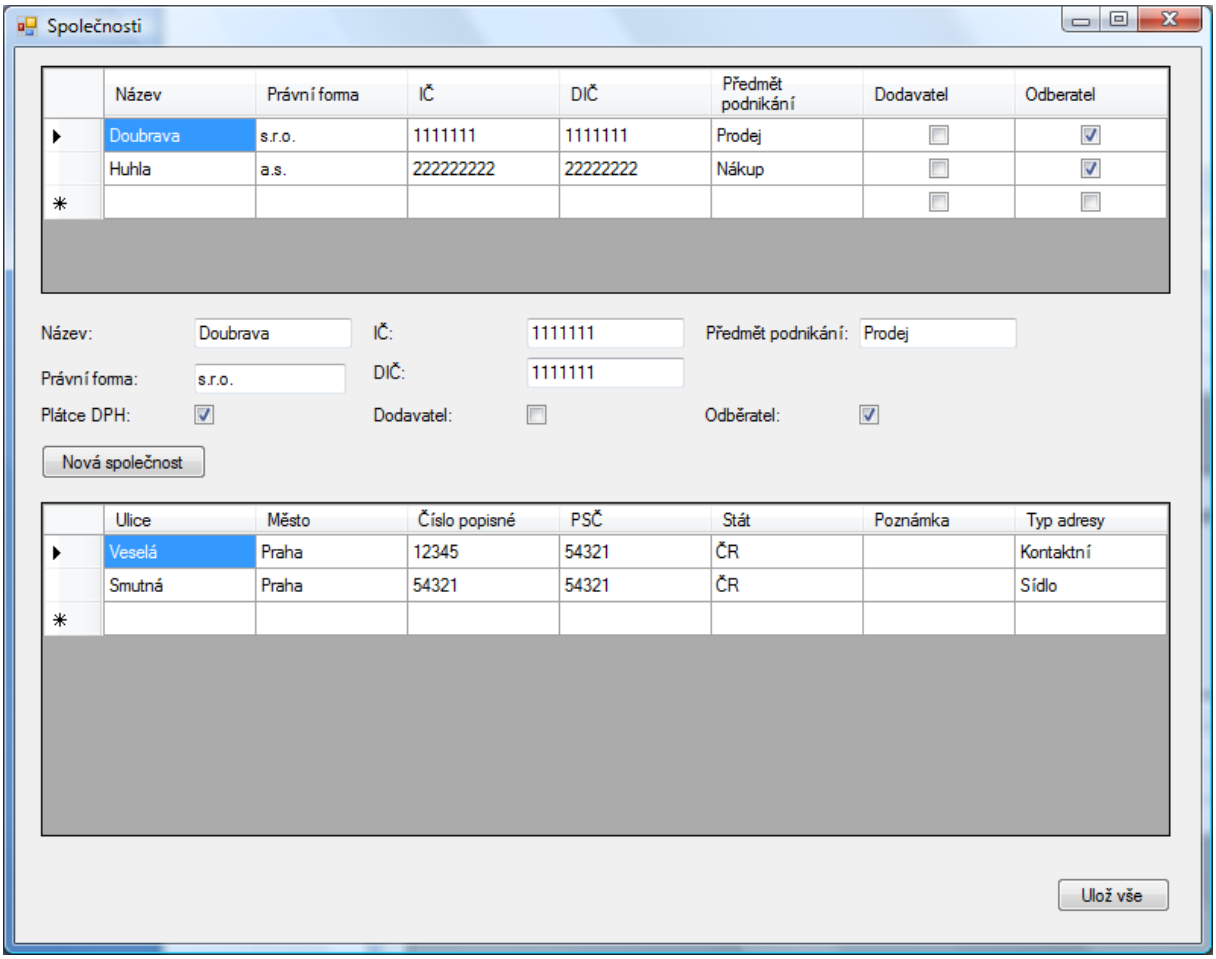
Obrázek 4-1 Hlavní okno

4.3.2 Okno práce se společnostmi

Okno umožňuje přidávat a editovat společnosti evidované v Systému. Okno obsahuje v horní části tabulku, určenou pro orientaci mezi společnostmi a tříditelnou podle sloupce tabulky. Pod tabulkou se nachází políčka, určená pro změnu, nebo vytvoření společnosti. Políčka jsou automaticky formátována pro příjem jen vhodných hodnot.

V dolní části okna se nachází tabulka s adresami zvolené společnosti, adresy lze přidávat rovnou pomocí tabulky.

Okno se nachází na obrázku 4-2.

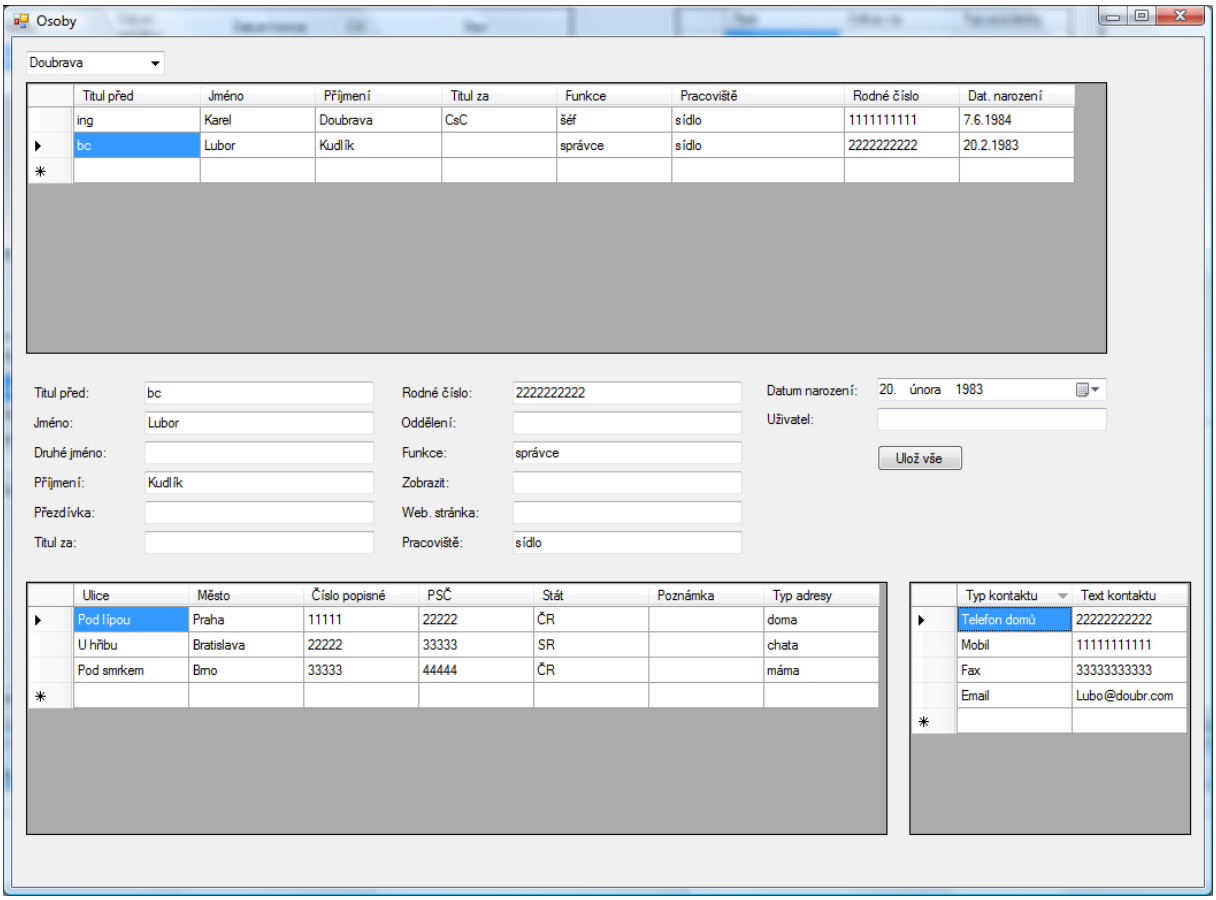


Obrázek 4-2 Okno společnosti

4.3.3 Okno práce s osobami

Okno umožňuje přidávat a editovat osoby vytvořené v Systému. Osoby jsou tříděny podle společnosti, ke které jsou evidovaní a která se vybírá pomocí ovládacího prvku combo boxu v levé horní části okna. Pod combo boxem se nachází tabulka osob, obsahující často využívaná políčka osob, která se dají přímo editovat. Pod tabulkou se nachází políčka všech ve verzi relevantních detailů osoby, pro detailnější specifikaci. Ještě níže se nachází tabulka adres dané osoby, umožňující přímo přidávání a editaci. Vedle tabulky adres se nachází tabulka kontaktů, umožňující přidávat kontakty na osobu.

Okno se nachází na obrázku 4-3.

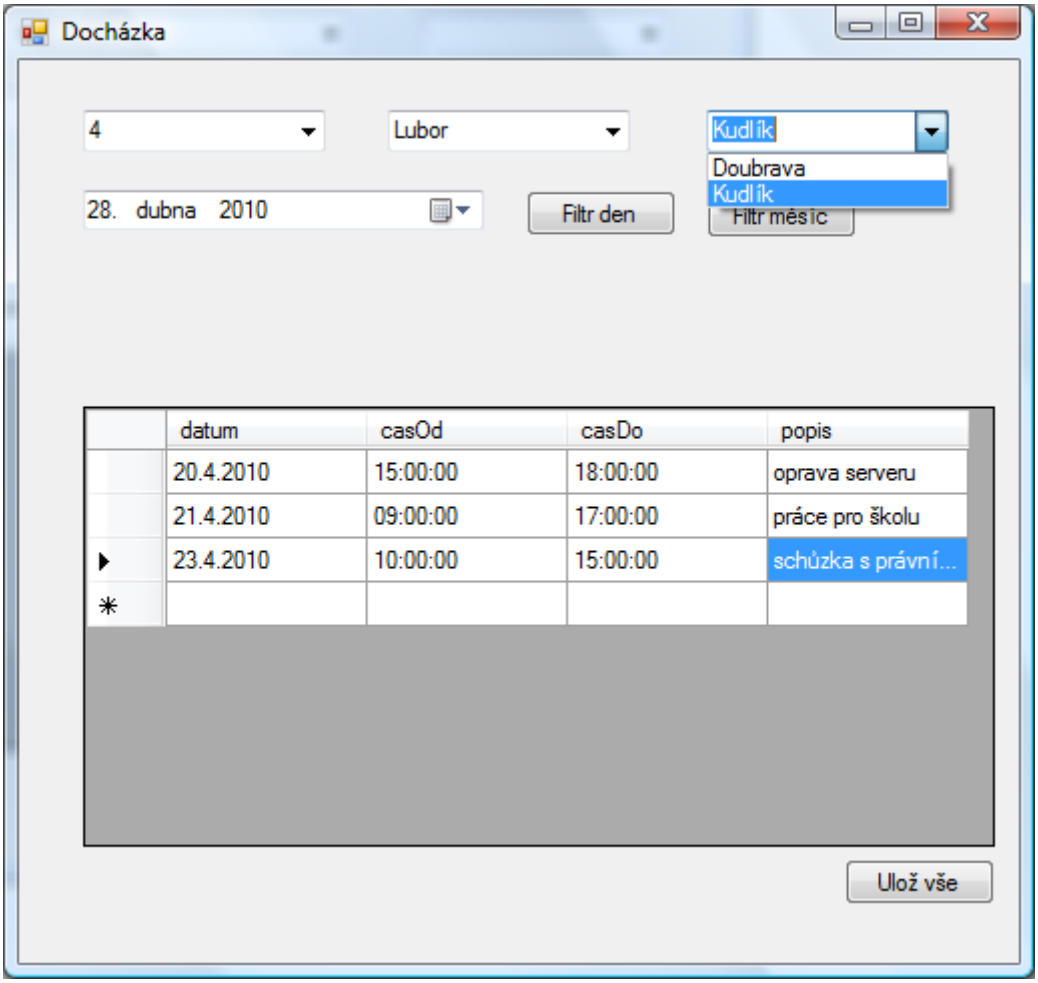


Obrázek 4-3 Okno osob

4.3.4 Okno práce s pracovními výkazy

Okno umožňuje přidávat a editovat časové výkazy vázané na osoby, které slouží k evidenci práce odvedené osobami. V horní části okna se nachází filtrační combo boxy, sloužící ke zvolení osoby podle jejího id, jména, nebo příjmení. Pod combo boxy se nachází ovládací prvek umožňující zvolit den a následně pomocí tlačítek omezit zobrazené výkazy na daný den, nebo měsíc.

Okno se nachází na obrázku 4-4

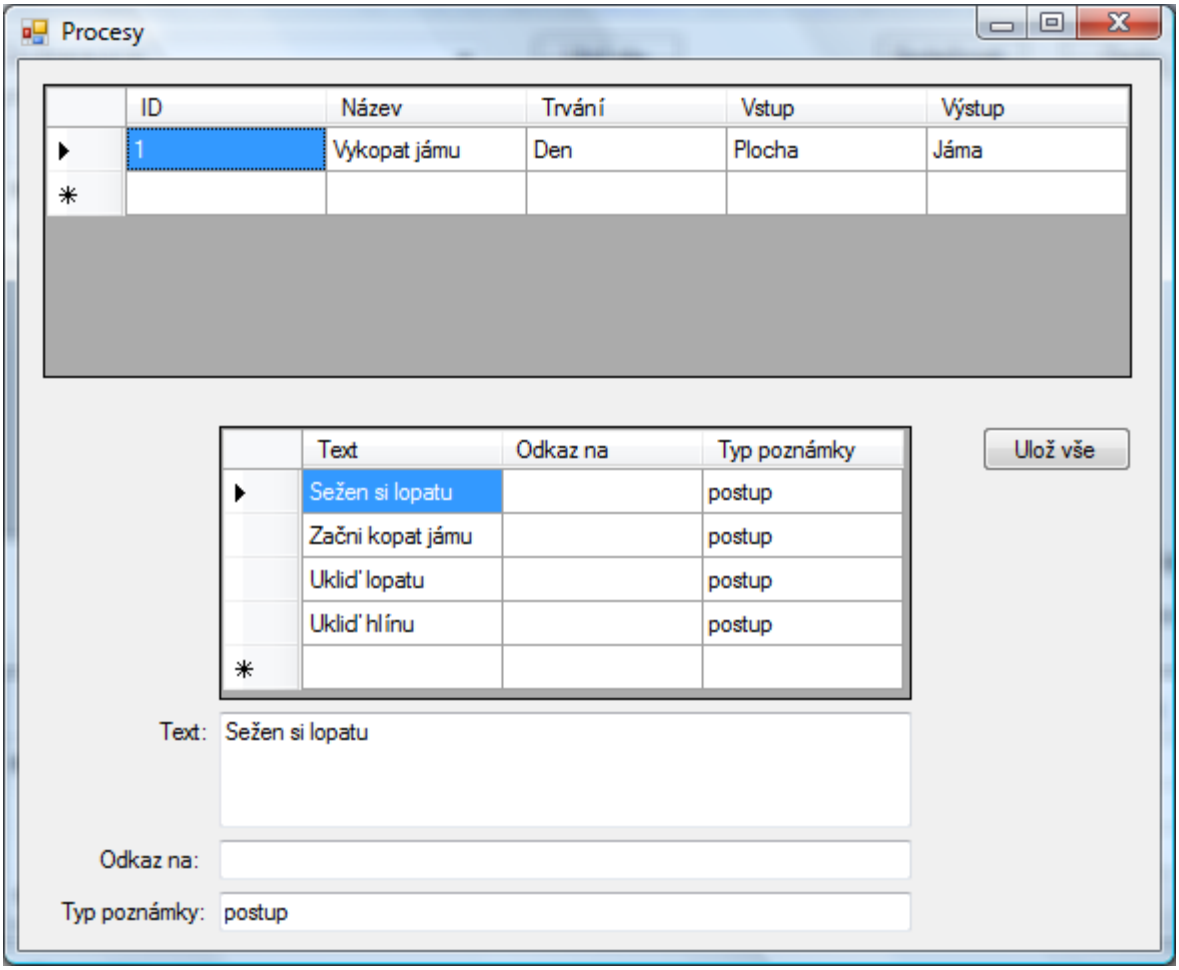


Obrázek 4-4 Okno docházky

4.3.5 Okno práce s procesy

Okno umožňuje přidávat a editovat standardizované procesy. V horní části okna se nachází tabulka procesů určená pro jejich správu. Pod ní se nachází tabulka poznámek vázaných na daný proces, umožňující tvorbu nových, nebo editaci poznámek. Pod ní se nacházejí pole pro snadnější práci s poznámkami.

Okno se nachází na obrázku 4-5



Obrázek 4-5 Okno procesů

5. Závěr

Z hlediska Společnosti byla vytvořená implementace úspěšná. Bylo dosaženo kritických funkcionalit Systému a zbývající budou dodány a vytvořeny v dalších verzích. Společnost získala Systém, který jí pomůže v evidenci kontaktů, odvedené práce zaměstnanců, projektů vytvářených pro zákazníky a umožní jí standardizovat časté projekty. Zároveň se zamezí dosavadním občasným ztrátám dat, definujících určité úkoly, nebo řešení.

Z hlediska autora byla práce také úspěšná, protože jsem si vyzkoušel tvorbu informačního systému. Naučil jsem se základům použití jazyka VB9, vyzkoušel jsem si možnosti MS SQL serveru 2008 a výhody některých aspektů extrémního programování, které bych rád aplikoval na svou další práci, především zaměření na kritickou funkcionalitu.

Z hlediska dosažené funkcionality byla práce poměrně úspěšná. Vytvořený Systém pracuje s většinou entit, kromě entit práv, která zatím nemají smysl kvůli neuskutečněnému rozhodnutí Společnosti stran způsobů přihlašování a entit smluv, které jsou brány jen jako doplněk Systému pro snadnější orientaci, ale nejedná se o kritický problém. Bohužel nebyla vytvořena provazující logika mezi osobami a projekty/úkoly/osobami. Ta bude dodána v další verzi. Logika chybí z toho důvodu, že se nám nepodařilo navrhnout okna a metodiku práce takovou, která by tuto logiku již implementovala a také z důvodu mých nedostatečných zkušeností s jazykem VB9 a neochotě udělat nekoncepční řešení. Také bych zde rád upozornil na změnu programovacího jazyka v průběhu řešení a problémy s tím spojené.

Z výše uvedených důvodů si myslím, že práce byla úspěšná.

6. Terminologický slovník

Pojem	Definice	Zdroj
.NET	Platforma řešení a programů společnosti Microsoft.	[autor]
Analýza	Postup, jehož výstupem je logický (esenciální) model vytvářeného systému	[BUCHALCEVOVÁ 2009]
Architektura	Zobrazení komponent systému.	[autor]
Combo box	Ovládací prvek jazyka VB9.	[autor]
Data	Jakékoliv fyzicky (materiálně) zaznamenané znalosti (vědomosti), poznatky, zkušenosti nebo výsledky pozorování procesů, projevů, činností a prvků reálného světa; data mohou být zaznamenána jak nepočítačovým způsobem, tak na počítačových mediích	[BUCHALCEVOVÁ 2009]
Databáze	Objekt uspořádání dat, který se nachází na počítačovém médiu a je vhodný pro práci s těmito daty, snaží se jí umožnit.	[autor]
Dead-lock	Stav, kdy jeden klient čeká na uvolnění zámků druhého, zatímco druhý čeká na prvního.	[autor]
Extrémní programování	Metodika pro malé až středně velké týmy, které vyvíjejí programové vybavení a musí se vyrovnat se zadáním, které se rychle mění, nebo není jasné.	[BECK, 2002]
Implementace	Proces realizace, zavedení navržené entity do reálného světa.	[autor]
Metodika	Návod, stanovující principy vytváření systému.	[autor]
Model	Dílčí pohled naa vytvářený systém.	[BUCHALCEVOVÁ 2009]
Proces	Soubor vzájemně souvisejících nbo vzájemně působících činností, které přeměňují vstupy na výstupy.	ČSN/ISO 10006
Projekt	Jedinečný proces sestávající z řady koordinovaných a řízených činností s daty zahájení a ukončení, prováděný pro dosažení cíle, který vyhovuje specifickým požadavkům, včetně omezení daných časem, náklady a zdroji.	ČSN/ISO 10006
Server	Počítač, nabízející určitou funkcinalitu, schopný obsloužit vně přicházející požadavky; program, nabízející určitou funkcionalitu, schopný obsloužit vně přicházející požadavky.	[autor]
Společnost	Zadavatelská společnost, pro kterou je celý Systém vyvíjen.	[autor]
Systém	Systém pro sběr, přenos, uchování, zpracování a poskytování dat (informací, znalostí) využívaných při činnosti podniku. Jeho komponentami jsou informační a komunikační technologie, data a lidé. Jeho cílem je efektivní podpora informačních, rozhodovacích a řídících procesů na všech úrovních organizace	[VOŘÍŠEK 2008]
Use-case diagram	Jeden z diagramů UML, vyjadřuje chování systému a vztah k okolí	[BUCHALCEVOVÁ 2009]

VB9	Microsoft Visual Basic 2008, programovací jazyk, součást platformy .NET.	[autor]
Zástupce	Zástupce společnosti, který formuluje požadovanou funkcionalitu a se kterým jsou konzultována možná řešení.	[autor]

7. Seznam použitých zdrojů

ARLOW, J. NEUSTADT, I. UML 2 a unifikovaný proces vývoje aplikací. 1.vyd. Brno: Computer Press, 2008. ISBN 978-80-251-1503-9

BASL, J. BLAŽÍČEK, R. Podnikové informační systémy. 2.vyd. Praha: Grada, 2008. ISBN 978-80-247-2279-5

BECK, K. Extrémní programování. 1.vyd. Praha: Grada, 2002. ISBN 80-247-0300-9

BUCHALCEVOVÁ, A. Metodiky budování informačních systémů. 1.vyd. Praha: Oeconomica, 2009. ISBN 978-80-245-1540-3

Distributed Application Architecture [online] [cit. 2010-04-20]. Dostupný z WWW: <http://java.sun.com/developer/Books/jdbc/ch07.pdf>

ČSN/ISO 10006 Systémy managementu jakosti – Směrnice pro management jakosti projektů, ČNI 2003

FUCHS, J. BARCHFELD, A. Visual Basic Velká kniha řešení. 1.vyd. Brno: Computer Press, 2010. ISBN 978-80-251-2212-9

HALVORSON, M. Microsoft Visual Basic 2008 Krok za krokem. 1.vyd. Brno: Computer Press, 2008. ISBN 978-80-251-2221-1

HAMILTON, B. ADO.NET 3.5 Cookbook™, Second Edition. 2.vyd. Sebastopol: O'Reilly, 2008. ISBN 978-0-596-10140-4

PAGE-JONES, M. Základy objektově orientovaného návrhu v UML. 1.vyd. Praha: Grada, 2001. ISBN 80-247-0210-X

PECINOVSKÝ, R. Návrhové vzory. 1.vyd. Brno: Computer Press, 2007. ISBN 978-89-251-1582-4

STANEK, W. Microsoft SQL Server 2005 Kapesní rádce administrátora. 1.vyd. Brno: Computer Press, 2006. ISBN 80-251-1211-X

VLASÁK, R. BULÍČKOVÁ, S. Základy projektování informačních systémů. 1.vyd. Praha: Karolinum, 2003. ISBN 80-246-0727-1

VOŘÍŠEK, J. a kol. Principy a modely řízení podnikové informatiky. 1.vyd. Praha: Oeconomica, 2008. ISBN 978-80-245-1440-6.