

Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky
Katedra informačních technologií

Student:	Jan Kaltoun
Vedoucí bakalářské práce:	Ing. Dušan Chlapek, Ph.D.
Recenzent bakalářské práce:	Mgr. Stanislav Komínek

TÉMA BAKALÁŘSKÉ PRÁCE

Analýza a vývoj IS/ICT řešení pro projekt Sanace dluhů v ČR

Rok: 2008

Prohlášení

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

V Praze 11.12.2008

.....

Jan Kaltoun

Poděkování

Úvodem bych rád poděkoval Ing. Dušanu Chlapkovi, Ph.D. za vedení této bakalářské práce a pomoc a trpělivost při jejím zpracování.

Abstrakt

Tato bakalářská práce se zabývá analýzou a vývojem IS/ICT v rámci projektu Sanace dluhů v České republice řešeného Sdružením pro probaci a mediaci v justici, o.s. společně se státní Probační a mediační službou ČR a švýcarským sdružením VEBO.

Cílem této bakalářské práce bylo navrhnout a realizovat IS/ICT podporu projektu Sanace dluhů. Splnění tohoto cíle vyžadovalo realizaci mnoha činností od návržení architektury IS, přes naprogramování IASW, až po tvorbu dokumentace a školení uživatelů.

Úvodní analytická část práce popisuje výběr metodiky a následně se podrobněji zabývá metodikou, která byla pro projekt zvolena a jejím přizpůsobením. Zmiňuje pak také využití agilní techniky a modely užívané při analýze. V další kapitole autor popisuje a dokumentuje analýzu požadavků klienta a současného stavu.

V následujících částech se pak práce mimo jiné zabývá návrhem výsledného řešení, vybranými tématy týkajícími se vlastního vývoje, procesem předání vyvinutého řešení uživatelům a možnostmi budoucích rozšíření aplikace.

Přínos autora v této bakalářské práci spočíval v provádění aktivit specifikovaných zvolenou metodikou a v prezentaci jejich výstupů. Autor v rámci zpracování této bakalářské práce porozuměl projektu Sanace dluhů, důkladně zanalyzoval požadavky na jeho IS/ICT podporu, reagoval na jejich změny v průběhu času a na základě těchto požadavků vypracoval produkt, který odpovídá představám Žadavatele. Tento produkt autor následně uvedl do provozu po hardwarové i softwarové stránce a do jeho užívání zaškolil jeho budoucí uživatele.

Klíčová slova

projekt, IS/ICT, architektura, vývoj, software, databáze, konceptuální datový model, fyzický datový model, user interface flow diagram, use case diagram, procesní model, IASW, metodiky, modely, AUP, FDD, Scrum, XP, probace, sanace

Abstract

This paper describes the analysis and development of IS/ICT for project Sanace dluhů in the Czech republic which is being realised by The Association for Probation and Mediation in Justice in cooperation with Czech Probation and Mediation Service and Swiss association VEBO.

The goal of this project was to design and realize the IS/ICT support for project Sanace dluhů. In order to meet this goal many activities had to be performed such as designing the architecture of the information system, programming the IASW and the creation of documentation and training of future users.

The first part of this paper describes the process of choosing the methodology and continues with more detailed description of the chosen methodology and with the description of how the methodology was adapted to fit the project. In this part there are also mentioned the agile techniques and models used during the analysis. In the next chapter the author describes and documents the analysis of clients' requests and the present state.

Following chapters describe among other topics the design of the final solution, selected topics dealing with the application development and with the process of delivery of the developed solution to end-users. This part of the paper also discusses the future possibilities of extending the application.

Author's contribution lies in performing the activities specified by the chosen methodology and in presentation of their outputs. In the process of creation of this paper the author understood the project Sanace dluhů, thoroughly analysed the requests for its IS/ICT support, reacted on the changes in requests and created a product based on these requests that fulfills clients' vision. Author then put the product into operation considering both hardware and software and trained its users.

Keywords

project, IS/ICT, architecture, software, development, database, conceptual data model, physical data model, user interface flow diagram, use case diagram, process model, IASW, methodologies, models, AUP, FDD, Scrum, XP, probation, financial rehabilitation

Obsah

1	Úvod	8
1.1	Vymezení tématu práce a důvod jeho výběru	8
1.2	Zadavatel práce, jeho zástupce a další zúčastnění	8
1.3	Projekt Sanace dluhů v ČR	8
1.4	Cíle práce	10
1.5	Očekávané výstupy práce	10
1.6	Předpoklady a omezení práce	11
1.7	Struktura práce	11
2	Výběr a přizpůsobení metodiky	12
2.1	Důvod pro volbu agilní metodiky	12
2.2	Kritéria výběru metodiky	13
2.3	Přehled metodik	14
2.3.1	Extrémní programování	14
2.3.2	Feature-Driven Development	14
2.3.3	Scrum	15
2.3.4	Agile Unified Process	15
2.3.5	Další zvažované metodiky	16
2.3.6	Vyhodnocení	16
2.4	Přizpůsobení metodiky	17
2.5	Využití agilní techniky a jejich přizpůsobení	18
2.5.1	Agile Change Management	18
2.5.2	Agile Model Driven Development	18
2.5.3	Test-Driven Development	18
2.6	Modely využití při analýze a jejich notace	19
2.7	Procesní model	19
2.7.1	Use Case Diagram	19
2.7.2	User Interface Flow Diagram	19
2.7.3	Konceptuální datový model	20

2.7.4	Fyzický datový model	21
2.7.5	Free-form diagram	22
3	Analýza požadavků a současného stavu	23
3.1	Evidence požadavků	23
3.1.1	Komunikace se Zadavatelem	23
3.1.2	Katalog požadavků	23
3.2	Rozpočet a přizpůsobení řešení projektu	24
3.2.1	Platforma aplikace	25
3.3	Bezpečnostní rizika	25
4	Návrh řešení	27
4.1	Architektura IS/IT	27
4.2	Globální architektura	28
4.3	Dílčí architektury	28
4.3.1	Funkční architektura	28
4.3.2	Procesní architektura	28
4.3.3	Datová architektura	29
4.3.4	Softwarová architektura	29
4.3.5	Technologická architektura	30
4.3.6	Hardwarová architektura	30
4.4	Funkční součásti software	30
4.5	Softwarové a hardwarové požadavky projektu	32
4.5.1	Softwarové požadavky Řešitele pro zpracování projektu	32
4.5.2	Softwarové požadavky navrženého řešení	33
4.5.3	Hardwarové požadavky vyvinutého řešení	34
4.6	Uživatelské rozhraní aplikace	34
5	Vývoj navrženého řešení.	36
5.1	Datová základna	36
5.1.1	Přenesení Fyzického datového modelu do prostředí MS SQL Server	36
5.1.2	Vrstva uložených procedur	36

5.2	Zvolené IDE a programovací jazyk	36
5.3	Zabezpečení aplikace	36
5.3.1	Uživatelské účty aplikace	36
5.3.2	Komunikace po síti	37
6	Předání řešení uživatelům	38
6.1	Akceptační řízení	38
6.1.1	Způsoby ukončení akceptačního řízení	38
6.1.2	Klasifikace vad	38
6.1.3	Kritéria ukončení akceptačního řízení	39
6.2	Školení uživatelů	39
7	Rozšíření aplikace a další funkcionalita	40
8	Dokumentace systému.	42
8.1	Uživatelská příručka	42
8.2	Programátorská dokumentace	42
9	Závěr	43
A	Konceptuální datový model	51
B	Fyzický datový model	52
C	User Interface Flow Diagram.	53
D	Use Case diagramy	54
E	Protokol o akceptaci.	60
E.1	První akceptační řízení	60
F	Testovací scénář	62
G	Katalog požadavků	63
H	Ukázky obrazovek aplikace.	69

1 Úvod

Úvodní část této bakalářské práce vymezuje její téma a popisuje důvody, které měl autor pro jeho volbu. V následující části představuje autor Zadavatele práce, jeho zástupce a další osoby, které významně ovlivnily proces zpracování této bakalářské práce a zejména jedné z jejích součástí — vyvinuté aplikace. Po představení Zadavatele následují informace o projektu Sanace dluhů v ČR, jehož IS/ICT podpora byla v rámci této bakalářské práce zpracovávána.

V následujících třech částech pak autor popisuje cíle této práce, její očekávané výstupy a její předpoklady a omezení. Poslední část úvodu pak nastiňuje strukturu této bakalářské práce.

1.1 Vymezení tématu práce a důvod jeho výběru

Tématem této bakalářské práce je Analýza a vývoj IS/ICT pro projekt Sanace dluhů v ČR. Toto téma je třeba, jak lze odvodit již z jeho názvu, uvažovat ve dvou rovinách — v rovině analytické a v rovině vlastního vývoje.

V rámci analytické roviny bylo třeba zanalyzovat současný stav IS/ICT Zadavatele a jeho požadavky na IS/ICT podporu projektu Sanace dluhů. Po důkladné analýze následoval návrh řešení. Rovina vývoje se pak zabývala uvedením navrženého IS/ICT do provozu a vývojem IASW vycházejícího z informací získaných při činnostech analytické roviny.

Důvodů pro volbu tohoto tématu lze nalézt hned několik. V prvé řadě jsem hledal téma, jehož zpracování bude mít reálný dopad nejen v teoretické, ale i v praktické rovině a jehož výstupy budou po dokončení jeho zpracování někde využívány. Kromě toho se mi toto téma zdálo ideální z toho důvodu, že využívá prakticky veškeré znalosti, které jsem získal v předešlých třech letech studia předmětů Fakulty informatiky a statistiky Vysoké školy ekonomické v Praze, od znalosti metodik a příslušných modelů a jejich tvorby až po základy programování v jazyce C#.

1.2 Zadavatel práce, jeho zástupce a další zúčastnění

Zadavatelem této bakalářské práce a tedy odběratelem analýzy a výsledného software je Sdružení pro probaci a mediaci v justici, o.s. — občanské sdružení založené v roce 1994, jehož posláním je iniciování a rozvoj nových metod prevence kriminality a řešení následku trestné činnosti, které zohledňují zájmy a potřeby obětí, pachatele, příp. jejich sociálního prostředí [SPJ, 2008]. Zástupcem Zadavatele, který v průběhu řešení projektu poskytoval veškeré potřebné informace a koordinoval naplňování požadavků kladených na Zadavatele, byl Mgr. Stanislav Komínek — vedoucí projektu Sanace dluhů.

Externím konzultantem projektu byl pan Peter Gründler — prezident asociace VEBO (Verein zur Entwicklung der Bewährungshilfe in Osteuropa) a vedoucí sanačního oddělení probační služby v kantonu Curych. Ten projektu propůjčil své zkušenosti a významně svými připomínkami ovlivnil vývoj výsledného software. Dalšími osobami více či méně aktivními při řešení projektu byli interní a externí zaměstnanci Zadavatele, a to zejména z oblasti účetnictví a správy hardware a software.

1.3 Projekt Sanace dluhů v ČR

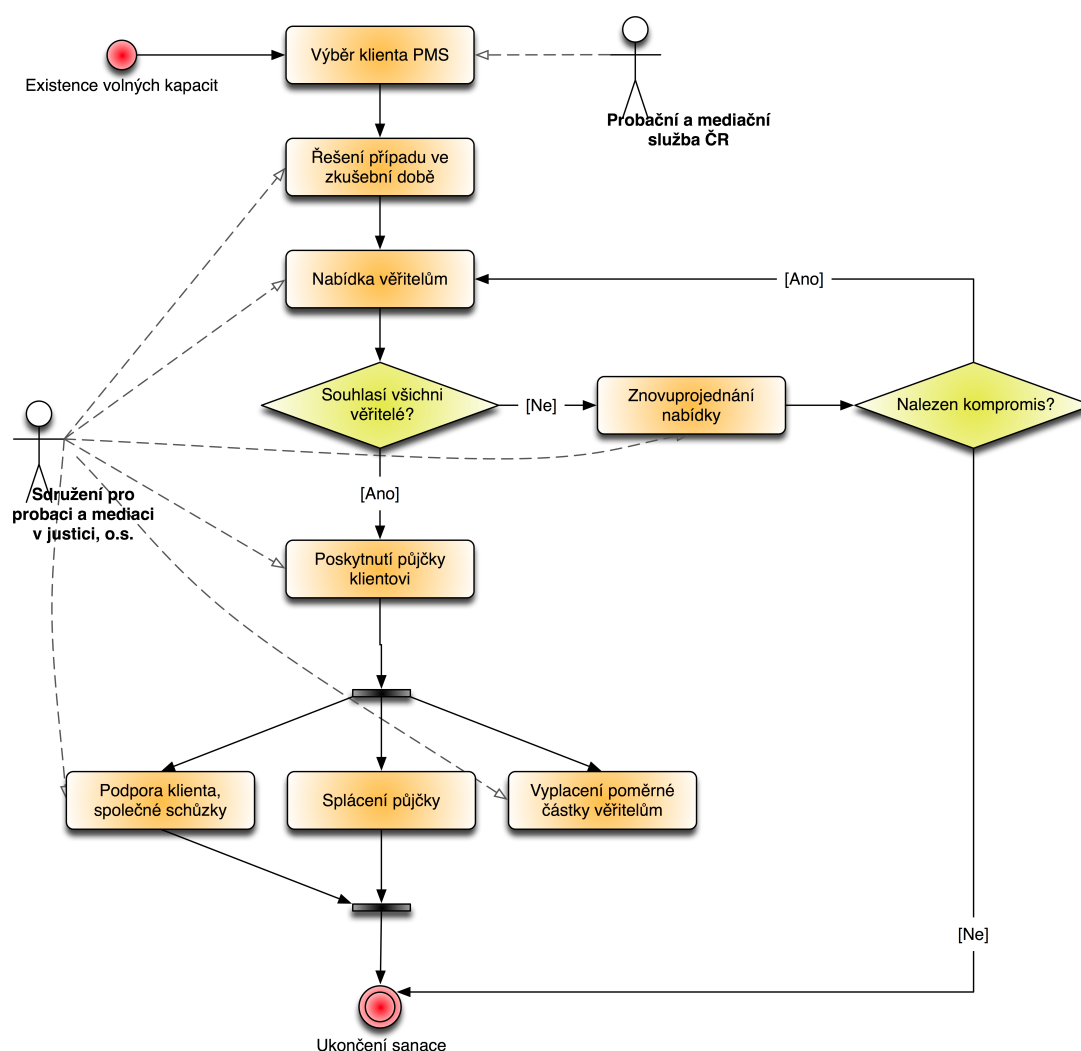
Projekt Sanace dluhů realizuje Sdružení pro probaci a mediaci v justici, o.s. společně se státní Probační a mediační službou ČR a švýcarským sdružením VEBO. Jedná se o službu, která je určena pro vybrané klienty Probační a mediační služby ČR. Cílem je řešit předluženost lidí po výstupu z výkonu trestu odnětí svobody, podpořit jejich resocializaci, legalizovat zaměstnanost a dosáhnout vyšší výnosnost pro jejich věřitele.

Inspiraci čerpá projekt Sanace dluhů ve Švýcarsku, kde systém efektivního řešení předluženosti

bývalých pachatelů trestných činů úspěšně funguje přes 30 let. Věřitelům je průměrně nabízeno 25% jejich pohledávky, což je většinou více než by od této skupiny dlužníků dokázali získat jinou cestou.

Sanace dluhů je službou nejen pro věřitele a jejich dlužníky, ale i pro stát a jeho občany. Podporuje vůli propuštěných řešit své závazky, preventivně chrání společnost před kriminalitou, efektivně motivuje k legalizaci zaměstnanosti a zamezuje dalšímu navyšování nákladů státního rozpočtu.

Ve dvouleté pilotní fázi je služba nabízena podmíněně propuštěným klientům PMS ČR v regionech Ústeckého a Středočeského kraje. Realizátoři věří, že projekt přispěje nejen ke zmapování možností využití systému Sanace dluhů, ale především již v pilotní fázi přispěje k úspěšnému řešení finanční situace 120 klientů PMS ČR a současně přinese všem věřitelům vyšší uspokojení pohledávek.



Poznámka:
Klient vystupuje ve všech procesech

Obrázek 1: Proces sanace dluhů | Zdroj: Autor (podle informací od Mgr. Stanislava Komínka)

Služba je poskytována všem zúčastněným zdarma. Vstup do projektu je jak pro dlužníky, tak pro věřitele dobrovolný. Pečlivě vybraní klienti musí mít legální zaměstnání, samostatně neřešitelné dluhy a vůli splatit je dle svých reálných možností. Věřitelům nabízejí klienti své volné zdroje, které jsou schopni našetřit ve zpravidla tříletém období. Při souhlasu všech věřitelů je jim částka jednorázově vyplacena ze sanačního fondu a dlužník stejnou částku splácí bezúročně zpět do fondu.¹

Vlastní proces sanace je ilustrován na Obr. 1. Notaci tohoto modelu je věnována zvláštní kapitola dále v textu.

1.4 Cíle práce

Cílem řešeného projektu bylo navrhnout a realizovat IS/ICT podporu projektu Sanace dluhů řešeného Zadavatelem. Vzhledem k tomu, že se projekt Sanace dluhů a projekt jeho IS/ICT podpory vyvíjely paralelně, nebylo možné navázat na jakoukoliv předešlou práci jiných autorů a bylo nutné projekt zpracovat od naprostého počátku, tedy od návržení architektury IS, přes naprogramování IASW až po tvorbu dokumentace a školení uživatelů. Měřítkem úspěšnosti splnění cílů této bakalářské práce tedy byla úspěšnost návrhu IS/ICT řešení a jeho následné implementace.

1.5 Očekávané výstupy práce

Řešený projekt lze rozdělit na dvě logické části, resp. skupiny činností — na analytickou a na vývojařskou část. Každá z těchto částí má určité výstupy, které bylo třeba při zpracování projektu vytvořit.

Výstupy *analytické části* jsou zejména následující dokumenty:

- tento text
- katalog požadavků (součástí příloh této bakalářské práce)
- konceptuální datový model (součástí příloh této bakalářské práce)
- fyzický datový model (součástí příloh této bakalářské práce)
- user interface flow diagram (součástí příloh této bakalářské práce)
- use case diagramy (součástí příloh této bakalářské práce)
- testovací scénáře pro akceptaci (příklad součástí příloh této bakalářské práce, ostatní dostupné Zadavateli)
- protokoly o akceptaci (příklad součástí příloh této bakalářské práce, ostatní dostupné Zadavateli)
- uživatelská příručka (dostupná Zadavateli)
- prezentace pro školení uživatelů (dostupná Zadavateli)

Výstupy *vývojařské části* lze charakterizovat následujícím výčtem. Pro zachování stručnosti bakalářské práce jsou výstupy této části dostupné pouze Zadavateli.

- hardware a podpůrný software nakonfigurovaný dle požadavků projektu
- vyvinutá aplikace
- instalační skripty
- programátorské testy
- dokumentace kódu aplikace
- dokumentace uložených procedur databáze

¹Informace pro tuto kapitolu poskytl Mgr. Stanislav Komínek.

1.6 Předpoklady a omezení práce

Hlavním předpokladem úspěšného ukončení zpracovávaného projektu byla častá a pružná komunikace se zástupci Zadavatele po celou dobu řešení projektu. Bez splnění tohoto předpokladu by projekt skoro jistě skončil neúspěchem vzhledem k tomu, že požadavky na zpracovávané řešení bylo možné získat pouze při diskuzích se zástupci Zadavatele.

Nejvýraznějším omezením, které bylo třeba brát v úvahu zejména při analýze hardwarových a softwarových požadavků projektu byl fakt, že Zadavatel nebyl na projekt schopen vyčlenit žádné finanční prostředky. Efekty vyplývající z této situace jsou popsány v kapitole 3.2.

1.7 Struktura práce

V úvodu této bakalářské práce je vymezeno její téma, její cíle a očekávané výstupy, je v něm také představen projekt Sanace dluhů a jeho řešitel. Druhá část práce popisuje výběr metodiky a následně se podrobněji zabývá metodikou, která byla pro projekt zvolena a jejím přizpůsobením. Dále pak tato část popisuje agilní techniky, kterých bylo v rámci projektu využito a modely užívané při analýze. Ve třetí části autor popisuje a dokumentuje analýzu požadavků klienta a současného stavu.

Čtvrtá část se zabývá návrhem výsledného řešení, zejména pak návrhem IS/IT architektury a funkčních součástí vyvinuté aplikace. Pátá část stručně zmiňuje zajímavá témata týkající se vlastního vývoje. V šesté a sedmé části je popsán proces předání vyvinutého řešení uživatelům a možnosti budoucích rozšíření aplikace. Osmá část pak popisuje dokumentaci projektu.

V závěru práce autor shrnuje dosažení stanovených cílů, hodnotí využitelnost dosažených výsledků a diskutuje svůj přínos k bakalářské práci a její přínos pro něj. Závěr pak následují přílohy této bakalářské práce, které doplňují zejména její analytickou a návrhovou část.

2 Výběr a přizpůsobení metodiky

Jak již vyplývá z jejího názvu, zabývá se tato kapitola tématem metodiky. V následujících odstavcích jsou nejprve popsány důvody pro volbu agilní metodiky, které následuje popis procesu výběru specifické metodiky použité při zpracování této bakalářské práce. Po definování kritérií, podle kterých byla metodika vybírána následují krátké charakteristiky zvažovaných metodik a vyhodnocení.

Druhá polovina této kapitoly pak popisuje způsob, jakým byla metodika upravena, zabývá se použitými agilními technikami a konečně popisuje druhy modelů využitých při analýze a jejich notace.

2.1 Důvod pro volbu agilní metodiky

Aby bylo možné jasně definovat plán zpracování projektu, bylo v jeho počátku nutno zvolit metodiku, jíž se projekt bude řídit. Již v této části vyvstal při komunikaci se Zadavatelem problém, jež projekt provázal po celou dobu jeho života — neúplně zpracovaný plán projektu Sanace dluhů, a tedy nejisté požadavky na zpracováváný software, jež měl tento projekt podpořit z infromatického hlediska. Tento problém nebylo možné identifikovat ihned v počátku získávání informací o požadavcích, nebo dokonce z úvodního zadání bakalářské práce, vzhledem k tomu, že projekt Sanace dluhů zpočátku jako celek působil dopracovaně a základní požadavky na funkcionalitu aplikace zároveň vycházely z funkcionality aplikace švýcarského projektu a zdály se tedy být dostatečně jasné. Problém se projevoval zejména při jednáních se Zadavatelem a zástupcem švýcarského projektu, a to nejčastěji následujícími způsoby:

- Interními změnami v návrhu projektu Sanace dluhů.
- Doporučeními na změny v návrhu projektu Sanace dluhů vznesené zástupcem švýcarského projektu.
- Logickými nedostatky v pochopení některého z procesů švýcarského modelu pracovníky Zadavatele.
- Doporučeními na změny v aplikaci vznesené zástupcem švýcarského projektu.

Dalším problémem vedoucím k nutnosti úprav požadavků na aplikaci prakticky po celou dobu jejího vývoje bylo, že zástupce švýcarského projektu, jehož komentáře k projektu byly pro Zadavatele klíčové, se do České republiky k jednání o aplikaci dostavoval nepravidelně a s velkými časovými odstupy.

Výše uvedené vedlo ke skutečnosti, že se požadavky na vytvářenou aplikaci velice často měnily, čemuž se samozřejmě bylo nutno přizpůsobit. Řešení takového projektu se však diametrálně odlišuje od řešení projektu, kterým se zdál být v počátku jeho zpracování. Původně projekt zahrnoval využití rigoróznějšího přístupu v počáteční fázi, zejména s podrobně nadefinovaným katalogem požadavků a podrobnými modely. Tento přístup také předpokládal pouze nepřilíš rozsáhlé změny v požadavcích na funkcionalitu v průběhu vlastního vývoje. V důsledku tedy bylo třeba zvolit přístup odlišný a ideálním řešením bylo využití některé z agilních metodik.

Při zpracování projektu byly tedy samozřejmě respektovány základní principy agilního vývoje software, které vycházejí z Agile manifesta vytvořeného organizací Agile Alliance [Tu, Fr, Ru, 2002]. Toto manifesto a principy jež ho rozvádějí prezentují základní myšlenky, předpoklady a doporučení pro správnou aplikaci agilního přístupu.

1. „Naší prioritou je uspokojit zákazníka pomocí brzkých a plynulých dodávek hodnotného software.“
2. „Obchodníci a vývojaři musí v průběhu projektu spolupracovat.“
3. „Vítejte požadavky na změny, a to i v pokročilých fázích vývoje.“

4. „Často dodávejte funkční software.“
5. „Funkční software je primárním měřítkem postupu.“
6. „Stavte projekty kolem motivovaných jedinců. Poskytněte jim prostředí a podporu kterou potřebují a důvěřujte jim, že práci odvedou.“
7. „Nejlepší architektury, požadavky a návrhy jsou produktem samoorganizujících se týmů.“
8. „Nejúčinnější a nejefektivnější metodou předávání informací vývojařskému týmu (a také uvnitř týmu) je konverzace z očí do očí.“
9. „Agilní procesy podporují udržitelný vývoj.“
10. „Nepřetržitý důraz na excelentní technické zpracování a dobrý design zvyšuje agilitu.“
11. „Jednoduchost je esencí.“
12. „Projektové týmy v pravidelných intervalech hodnotí svou efektivitu a podle výsledků upravují své chování.“

Lze konstatovat, že na řešený projekt bylo možné aplikovat každý z těchto dvanácti principů. Zejména principy č. 3, 5 a 8 se ukázaly být klíčovými pro jeho úspěch. Jinými slovy, projekt byl zaměřen na dodání software, který by co nejlépe odpovídal aktuálním požadavkům Zadavatele, cestou častých změnových požadavků evidovaných zejména při osobních jednáních.

2.2 Kritéria výběru metodiky

Vzhledem k tomu, že agilních metodik je velké množství, bylo třeba stanovit kritéria pro výběr jedné z nich. Řešený projekt byl velice specifický zejména disponibilními lidskými zdroji, časem na jeho zpracování a požadavky na artefakty². Kritéria byla vzhledem k specifikům projektu stanovena následujícím způsobem:

- **K1 — Metodika je agilní.**
Toto kritérium bylo zařazeno z nutnosti vypořádání se s častými změnami.
- **K2 — Metodika v ideálním případě pokrývá všechny disciplíny projektu od modelování až po deployment.**
Toto kritérium bylo zařazeno z důvodu komplexnosti projektu.
- **K3 — Metodiku lze přizpůsobit na jednočlenný tým.**
Kritérium jednočlenného týmu bylo vyžadováno skutečností, že jediným řešitelem projektu byl autor této bakalářské práce.
- **K4 — Metodika nemá striktně dány žádné časové intervaly.**
Absence striktních časových intervalů byla nutná z důvodu vytížení autora zejména navštěvováním školy a prací.
- **K5 — Metodika se vyznačuje vysokou ceremonií³.**
Vysoká ceremonie byla vyžadována zejména faktem, že projekt byl řešen v rámci bakalářské práce.

²artefakt — výstup projektu

³vysokou ceremonií rozumíme velké množství aktivit a artefaktů v projektu

2.3 Přehled metodik

2.3.1 Extrémní programování

Extrémní programování je metodikou, jež se v posledních letech dostala do popředí zájmu o agilní metodiky a jeho použití tedy samozřejmě bylo jednou ze zvažovaných variant. Jejím autorem je Kent Beck, který ji vytvořil v 90. letech 20. století. Extrémní programování je metodikou která se, jak již vyplývá z názvu, silně zaměřuje na vlastní programování za použití specifických technik [Jeffries, 2001] jako jsou zejména následující:

- programování v páru
- refaktORIZACE
- společné vlastnictví kódu
- Test-Driven development

Stejně tak jako u dalších uvedených agilních metodik nebylo možné extrémní programování při porovnávání kandidátů na metodiku principiálně vyloučit vzhledem k tomu, že agilní metodiky už z principu sdílejí stejné cíle a prezentují pouze jiné způsoby jak těchto cílů dosáhnout. Uvědomíme-li si však, že jednou ze základních praktik [Jeffries, 2001] extrémního programování je párové programování, zatímco projekt byl zpracováván pouze jednou osobou a že jedním z předpokladů úspěchu je on-site přítomnost zástupce Zadavatele, vhodnost využití této metodiky rapidně klesá. Extrémní programování bylo tedy vyloučeno zejména (avšak nejen) z důvodu nemožnosti aplikovat jednu z jeho základních praktik a jeden z jeho základních předpokladů.

2.3.2 Feature-Driven Development

Feature-Driven Development je agilní metodikou, která vzešla z rukou Jeffa De Luca a Petera Coad. Již z názvu vyplývá, že základním principem FDD je zaměření se na užitné vlastnosti (features) výsledného produktu. Užitná vlastnost, je tedy v FDD základní jednotkou práce, přičemž je definována jako následující funkce [Coffin, Lane, 2006]:

`<akce><výsledek>[čeho|k čemu|pro|od]<objekt>`

FDD definuje 5 procesů projektu, jež je třeba implementovat ve striktním pořadí. Tyto procesy jsou [Coffin, Lane, 2006]:

- Vytvoření obecného modelu
- Sestavení seznamu užitných vlastností
- Plánování podle užitných vlastností
- Návrh podle užitných vlastností
- Výstavba podle užitných vlastností

Pro každou užitnou vlastnost je také definováno 6 milníků, pomocí kterých lze zachytit, v jaké části vývojového cyklu se právě daná užitná vlastnost nachází. Tyto milníky jsou [Coffin, Lane, 2006]:

- Domain Walkthrough
- Návrh
- Inspekce návrhu
- Kódování
- Inspekce kódu

- Povýšení k buildu

FDD dále přináší specifický dokument — Feature Set Progress Report, který je unikátní pro tuto metodiku a který prezentuje zejména postup vývoje užitečných hodnot rozdělených do logických oblastí. Od ostatních agilních metodik se dále odlišuje systémem rolí a odpovědností [Coffin, Lane, 2006], který však v rámci řešeného projektu netřeba příliš rozebírat vzhledem k tomu, že Řešitelem byla pouze jedna osoba.

Závěrem lze konstatovat, že proti implementaci metodiky Feature-Driven Development hovoří zejména její zaměření pouze na vlastní vývoj a ignorování ostatních fází projektu.

2.3.3 Scrum

Scrum je framework zaměřený na management agilních projektů. Charakteristické jsou pro něj 2 až 4 týdny dlouhé iterace zvané Sprints a krátká každodenní jednání projektového týmu - Scrum meetings [Coffin, Lane, 2006]. Scrum se zabývá pouze managementem projektu, technické aspekty neřeší vůbec a je ho tedy vhodné kombinovat s některou z dalších metodik.

Scrum by byl pro řešený projekt zvolen (v kombinaci s některou z dalších metodik) pravděpodobně v případě, že by řešitelský tým měl více členů⁴ a/nebo že by při jeho zpracování hrál významnou roli čas. Vzhledem k tomu, že taková situace nenastala, Scrum by oproti zvolené metodice nepřinesl žádné významné výhody.

2.3.4 Agile Unified Process

Metodiku AUP lze, vzhledem k množství artefaktů z ní vyplývajících, charakterizovat jako metodiku s vysokým stupněm ceremonie. Byla vytvořena Scottem Amblerem v roce 2005 na základě jeho zkušeností s RUP, jako výsledek snah o aplikaci agilních principů na RUP. Scott Ambler o metodice uvádí, že je „sériová z širokého úhlu pohledu, iterativní z bližší perspektivy, v čase dodávající inkrementální verze produktu“ [Ambler, 2008].

Princip „Sériová z širokého úhlu pohledu“ vyplývá ze čtyř fází⁵ AUP [Ambler, 2008]:

- **Zahájení** — V této fázi je třeba identifikovat rozsah projektu a potenciální architekturu systému. Dále je třeba zajistit financování projektů a souhlas investorů.
- **Zdokonalení** — Cílem této fáze je ověřit architekturu systému.
- **Konstrukce** — V této fázi je cílem vytvořit fungující software na bázi přírůstků, které jsou v souladu s prioritami investorů.
- **Přechod** — Cílem poslední fáze je schválit výsledný systém a umístit ho do produkčního prostředí.

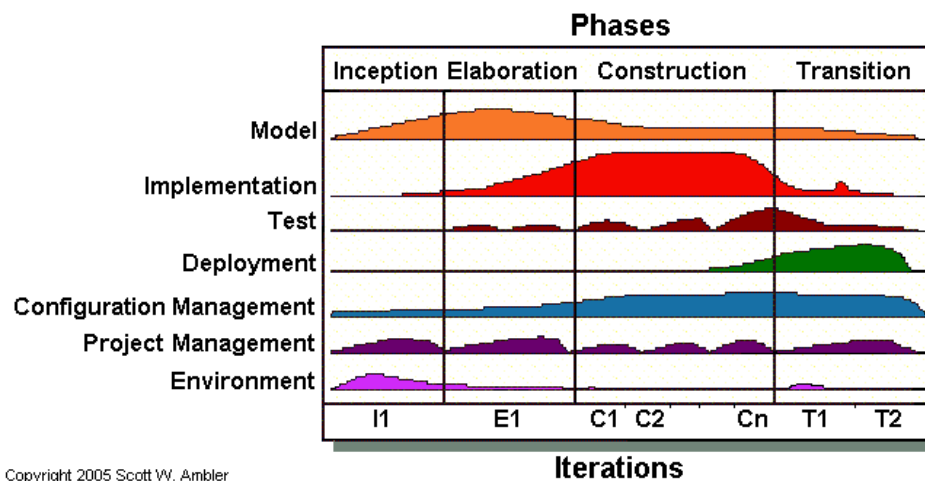
Princip „Iterativní z bližší perspektivy“ vychází ze sedmi disciplín AUP [Ambler, 2008]:

- **Modelování** — Tato disciplína se zabývá pochopením činnosti organizace, domény problémů, která je řešena projektem a identifikací schůdného řešení těchto problémů.
- **Implementace** — Cílem této disciplíny je přetransformovat model(y) do spustitelného kódu a provést základní testování (zejména pomocí Unit testů).
- **Testování** — Cílem této disciplíny je provést objektivní hodnocení pro zajištění kvality. Tato činnost zahrnuje hledání defektů, kontroly jestli systém funguje tak jak byl navržen a kontroly jsou-li splněny všechny požadavky.

⁴doporučený počet je 4-9 členů [Coffin, Lane, 2006]

⁵každou fází lze dále rozložit do iterací

- **Uvedení do provozu** — Tato disciplína zahrnuje vytvoření plánu dodání systému a jeho provedení s cílem poskytnout systém koncovým uživatelům.
- **Management konfigurace** — Cílem této disciplíny je spravovat přístup k artefaktům projektu. Toto zahrnuje nejen sledování verzí artefaktů v čase, ale také kontrolu a spravování změn v nich.
- **Projektový management** — Cílem této disciplíny je řídit aktivity, které probíhají v projektu jako např. správu rizik, řízení lidských zdrojů a koordinaci s lidmi a systémy mimo projekt pro zajištění dodání systému v termínu a v rámci rozpočtu.
- **Postředí** — Cílem této disciplíny je podporovat činnosti v projektu tím, že týmu budou v případě potřeby dostupné odpovídající procesy, směrnice a nástroje (hardware, software a.j.).



Obrázek 2: Životní cyklus AUP | Zdroj: [Ambler, 2008]

Posledním zde zmíněným principem je „V čase dodávající inkrementální verze produktu“. Tento princip zajišťuje dodávání verzí software na základě přírůstků, přičemž jsou tyto verze rozděleny do dvou skupin — Vývojové verze a Produkční verze. Vývojové verze jsou principiálně stejné jako ty produkční, s tím rozdílem, že Vývojová verze nebyla podrobena procesu zajištění kvality, testování a procesu uvedení do produkčního prostředí. Doba zpracování prvních Produkčních verzí je zpravidla delší než je tomu u následujících. V řešeném projektu následovalo vydání Produkční verze vždy po dokončení přírůstkové skupiny (resp. přírůstkových skupin) — viz. Obr. 6.

Metodika AUP doporučuje využití agilních technik jako je Agile Model Driven Development, Test Driven Development, Agile Change Management a Database Refactoring. Zároveň podporuje vytváření modelů a dokumentů, které jsou „Jen tak tak dostačující“, tedy takových modelů a dokumentů které obsahují jen to, co je v tu kterou chvíli opravdu třeba [Ambler, 2008].

2.3.5 Další zvažované metodiky

Kromě metodik AUP, XP, FDD a Scrum byly zvažovány i některé další agilní metodiky jako Lean, nebo DSDM, v tomto dokumentu jsou však v zájmu zachování stručnosti popsány pouze ty subjektivně nejzajímavější z nich.

2.3.6 Vyhodnocení

Z předchozího popisu a zamítnutí několika zvažovaných metodik je zřejmé, že metodikou zvolenou pro řešený projekt se stala metodika AUP, a to proto že nejlépe odpovídá (hodnocení je

samozřejmě částečně subjektivní) kritérium definovaným v kapitole 2.2. Tato skutečnost je ilustrována v Tab. 1.

	K2	K3	K4	K5
Scrum	-	-	-	-
AUP	x	x	x	x
XP	-	-	x	-
FDD	-	x	x	-

Tabulka 1: Vyhodnocení metodik

2.4 Přizpůsobení metodiky

AUP Product poskytovaný Scottem Amblerem lze ihned využít při řešení projektu, je však doporučeno přizpůsobit metodiku vlastním potřebám. Pro projekt řešený v rámci této bakalářské práce byl tedy AUP Product zjednodušen a přizpůsoben následujícím způsobem:

- Fáze AUP a jejich cíle byly ponechány beze změn.
- Z jednotlivých milníků byly odstraněny aspekty financování vývoje.
- Disciplíny AUP byly ponechány beze změn, se zvýšením intenzity disciplíny Model v prvních dvou fázích pro reflektování podrobnějšího modelování v počátku projektu.
- Všechny role převzala jedna osoba řešící projekt.
- Deliverables (artefakty projektu) byly definovány tak jak je uvedeno v Tab. 2.
- Sekce *Guidances* byla kompletně vyřazena, vzhledem k jedné osobě přebírající veškeré role.

název	popis
Systém	produkční verze software
Zdrojový kód software	zdrojový kód aplikace v jazyce C# a uložené procedury v jazyce T-SQL
Regresní sada testů	akceptační testy, programátorské testy
Instalační skripty	SQL pro založení databáze (vyvinutá aplikace instalaci nevyžaduje)
Systémová dokumentace	uživatelská příručka a programátorská dokumentace
Poznámky k verzi	poznámky k produkční verzi software
Model požadavků	katalog požadavků, konceptuální datový model, procesní modely, use case diagramy, akceptační testy
Modely návrhu	fyzický datový model, user interface flow diagram

Tabulka 2: Artefakty projektu

Jak již bylo uvedeno v jejím popisu, AUP doporučuje využití agilních technik a některé z nich — Agile Change Management, Agile Model Driven Development a Test Driven Development — byly využity i v řešeném projektu.

Poslední významnější úpravou AUP bylo zavedení konzultací Vývojových verzí software se zástupcem Zadavatele. Důvodem pro zavedení tohoto opatření bylo poskytnutí možnosti kontroly

straně Zadavatele a snaha eliminovat co největší množství nesrovnalostí mezi vizí Zadavatele a reálným stavem ještě před uvedením Produkční verze.

2.5 Využití agilní techniky a jejich přizpůsobení

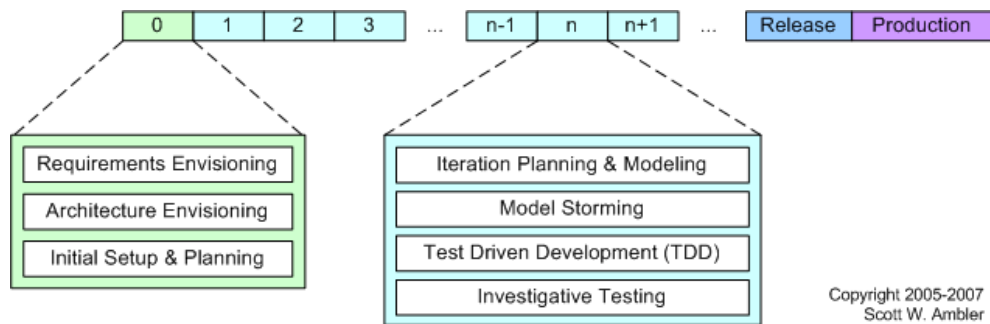
2.5.1 Agile Change Management

Z důvodu častých změn v požadavcích ze strany Zadavatele bylo třeba implementovat přístup, který by na tyto změny dokázal reagovat. Ideální návod na vypořádání se s častými změnami v požadavcích dává technika ACM, která byla naimplementována způsobem respektujícím následující principy:

- Priority zpracování požadavků lze měnit pouze v rámci jednotlivých přírůstkových skupin a ne mezi nimi.
- Technika respektuje podrobnost již vytvořených modelů z počátku řešení projektu.
- Požadavky jsou v rámci každé iterace zmrazeny.

2.5.2 Agile Model Driven Development

Tato technika využívá modelů jako hlavního nástroje pro vývoj software. Rozděluje vývoj do N iterací (viz. obr. 3), kdy v nulté iteraci je proveden tzv. Envisioning, tedy high-level modelování požadavků a počáteční modelování architektury. Ve vývojových iteracích jsou pak požadavky domodelovány (a tím také zanalyzovány a navrženy) do takové úrovně, aby je bylo možné v dané iteraci dokončit. Vývojové iterace také obsahují tzv. Model Stormingy, kdy několik členů týmu řeší pomocí modelování předem definovaný problém. Po modelování následuje vlastní programování, ideálně za využití techniky Test-Driven Development.



Obrázek 3: Životní cyklus AMDD | Zdroj: [Ambler, 2008]

V případě řešeného projektu byla tato technika implementována prakticky beze změn, pouze fáze envisioningu byla z již uvedených důvodů podrobnější než technika doporučuje.

2.5.3 Test-Driven Development

Tato technika významným způsobem snižuje počet chyb, které se dostanou do finální verze software prezentované klientovi. Jejím principem je vytváření testů před vlastním programováním, které jsou následně aplikovány na naprogramovanou část software. Projde-li test (tedy jeho výsledek je shodný s očekávaným výsledkem), testovaná část software je v pořádku. Neprojde-li test, je třeba provést změny v kódu a test spustit znovu — tyto dva kroky jsou následně opakovány dokud není výsledek testu shodný s očekávaným výsledkem.

2.6 Modely využití při analýze a jejich notace

2.7 Procesní model

Procesní model v tomto projektu slouží k zachycení procesů, jejich návaznosti a subjektů, které v těchto procesech figurují. Notace, která je použita v procesních modelech této bakalářské práce je mírně modifikovanou verzí notace software PowerDesigner.

- **Počátek** je reprezentován červeným kruhem
- **Konec** je reprezentován červeným kruhem s černou kružnicí uvnitř
- **Subjekt** je reprezentován postavičkou a s příslušnými procesy je spojen přerušovanou čarou s prázdným trojúhelníkem na konci
- **Proces** je reprezentován oranžovým obdélníkem s oblými rohy
- **Rozhodování** je reprezentováno žlutým rovnoběžníkem
- **Přesun** mezi jednotlivými elementy je znázorněn čarou s plným trojúhelníkem na konci

2.7.1 Use Case Diagram

Tento typ diagramu slouží pro zachycení způsobů využití projektovaného systému. Notace Use Case diagramů je komplikovanější, než notace zde prezentovaná, v modelech tohoto projektu však nejsou využívány všechny elementy, které Use Case diagramy nabízejí, a proto jsou v tomto odstavci popsány pouze prvky, které byly v některém z diagramů využity. Pro zpedrobnění Use Case diagramů je v tomto projektu využíván diagram popsáný v následujícím odstavci — User Interface Flow Diagram.

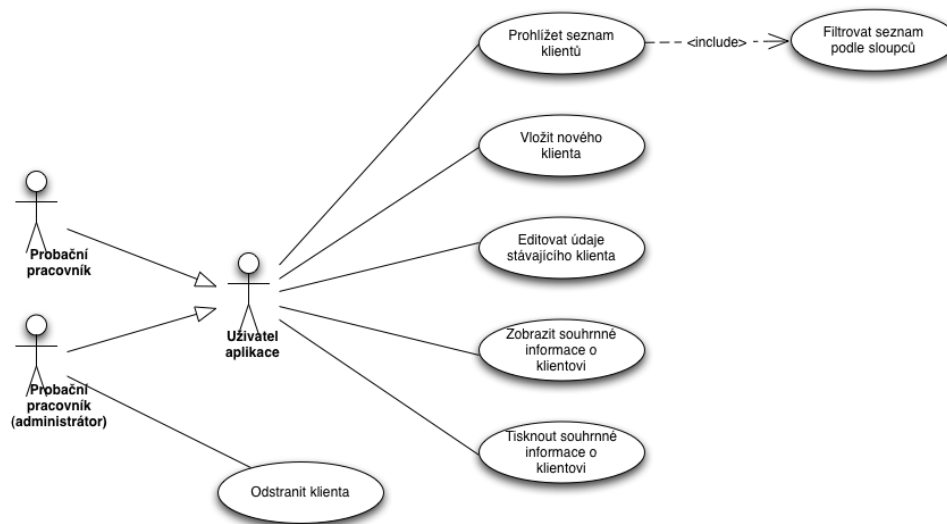
- **Use case** je případ užití, činnost kterou má pro účastníka smysl provádět a je reprezentován elipsou.
- **Účastník** je subjekt, který nějakým způsobem interaguje s modelovaným systémem a je reprezentován postavičkou.
- **Asociace** je vyjádření vztahu mezi jednotlivými use casey a/nebo účastníky a je reprezentována čarou.
- **Zahrnutí (include)** je vyjádření vztahu mezi jednotlivými use casey a říká, že use case A zahrnuje use case B. Zahrnutí je reprezentováno přerušovanou čarou s šipkou na konci a s popiskem «include».
- **Generalizace** je vyjádření vztahu mezi dvěma a více use casey, nebo účastníky a reprezentuje, jak již vyplývá z názvu, generalizaci (resp. specializaci) mezi uvedenými elementy. Generalizace je reprezentována čarou s prázdným trojúhelníkem na konci.

Obrázek č. 4 uvádí jednoduchý příklad Use Case Diagramu s dvěma aktéry, generalizací a několika use casey s jedním případem zahrnutí.

2.7.2 User Interface Flow Diagram

Tento diagram slouží k zachycení pohybu uživatele v aplikaci a je tak excelentním nástrojem pro zachycení výstupů diskuzí s klientem nad uživatelským rozhraním. Notace tohoto diagramu byla inspirována vzorovým diagramem tohoto typu od Scotta W. Amblera [Ambler, 2007], a to zejména z toho důvodu, že UML zatím tento typ diagramu neobsahuje (ačkoliv existují způsoby jak tyto informace zachytit za použití jiného typu modelu). Tato notace je velice jednoduchá a obsahuje prakticky pouze dva prvky.

- **Obrazovka uživatelského rozhraní** je reprezentována obdélníkem.



Obrázek 4: Příklad Use Case Diagramu | Zdroj: Autor

- **Pohyb uživatele** je reprezentován jednosměrnou šipkou a může obsahovat komentář (název ovládacího prvku UI). Šipky jsou pouze jednosměrné vzhledem k tomu, že je vždy předpokládána možnost návratu.

Dodatečně může tento diagram také sloužit k ověření použitelnosti návrhu UI, a to zhodnocením počtu jeho prvků a jejich provázanosti [Ambler, 2007].

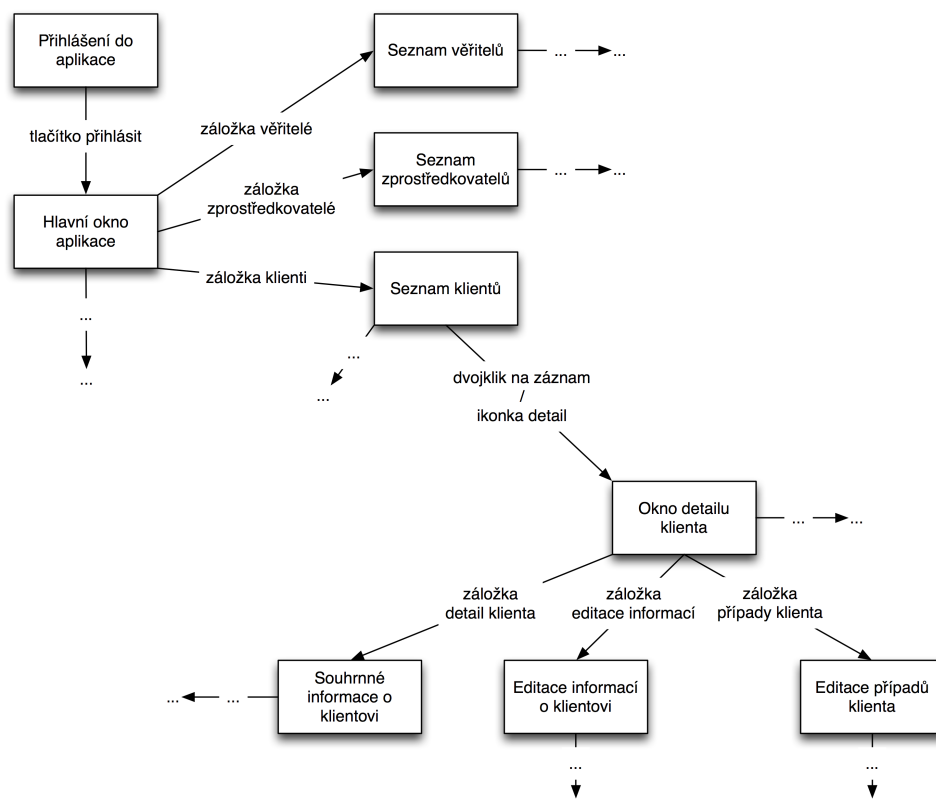
Na obr. 5 je prezentován příklad User Interface Flow Diagramu. Šipky a elementy označené „...“ reprezentují pokračování diagramu, které je však pro zachování stručnosti příkladu vypuštěno.

2.7.3 Konceptuální datový model

Tento model popisuje na konceptuální úrovni abstrakce entity vztahující se k zkoumané části reality a vztahy mezi nimi. Využívá se při návrhu datové základny projektu a měl by předcházet tvorbě fyzického datového modelu, který je popsán v následujícím oddílu. Konceptuální datový model nikdy nebere v úvahu pozdější konkrétní implementační detaily. Notace využitá pro tento typ modelu je následující.

- **Entity** — Jednotlivé entity jsou reprezentovány zeleně podbarvenými obdélníky. Každý z atributů entity má několik viditelných atributů (název, typ⁶ a informaci o povinnosti atributu vyjádřené symbolem <M> a o tom, je-li atribut součástí primárního identifikátoru vyjádřené jeho podtržením a symbolem <pi>).
- **Vztahy entit** — Entity spolu mohou vstupovat do vztahů reprezentovaných spojnicemi daných entit, které mají vždy definován název a kardinalitu, tedy počet vztahů typu reprezentovaného tou kterou spojnici, do kterých může entita vstupovat. Každé vyobrazení vztahu je dále na každém z konců doplněno o krátký slovní popis vlivu jedné entity na druhou (vždy u počáteční entity) a o vyjádření kardinality číselnou formou a v notaci Crow's feet (vždy u konečné entity). Speciálním případem vztahu použitého v modelech tohoto projektu je *dědičnost*

⁶je-li jako typ atributu uveden český název, znamená to, že pro skupinu atributů, do které tento atribut náleží byla vytvořena doména



Obrázek 5: Příklad User Interface Flow Diagramu | Zdroj: Autor

reprezentovaná půlelipsou, která může být proškrtnuta v případě, že byl specifikován atribut rozlišující dědicí entity.

2.7.4 Fyzický datový model

Je-li v projektu vytvořen konceptuální datový model, fyzický datový model obvykle vzniká jeho přenesením (ručně, nebo pomocí některého z CASE⁷ nástrojů) do prostředí specifického DBMS⁸. Fyzický datový model (v prostředí relační databáze) popisuje jednotlivé tabulky databáze a vztahy mezi nimi a popřípadě i další elementy relační databáze. Notace využitá pro tento typ modelu je popsána v následujícím seznamu.

- **Tabulky** jsou reprezentovány modře, nebo zeleně podbarvenými obdélníky. *Sloupce tabulky* mají několik viditelných atributů (název, datový typ a informaci o tom, je-li sloupec součástí primárního klíče vyjádřenou symbolem <pk>, popř. informaci o tom, je-li sloupec cizím klíčem — vyjádřenou symbolem <fkN>).
- **Views** jsou v modelu reprezentovány modře podbarvenými obdélníky s zakulacenými rohy a lze na nich přečíst název, jejich atributy (ačkoliv bez datových typů) a tabulku, od které jsou odvozeny.

⁷Computer-aided Software Engineering

⁸Database Management System

- **Vztahy mezi elementy** jsou vyjádřeny orientovanými šipkami, u kterých je vždy veden název constraintu⁹ a které jsou na svých koncích doplněny o krátký slovní popis jejich vlivu na daný element (vždy na bližším konci).

Ačkoliv je možné fyzický model doplnit o Uložené procedury, ty jsou vzhledem k jejich množství evidovány ve zvláštním dokumentu.

2.7.5 Free-form diagram

Free-form diagramy, jak už vyplývá z jejich názvu nemají jasně specifikovanou notaci a slouží zejména k popisu skutečností, na jejichž zachycení by běžně bylo třeba využít více modelů různých typů.

⁹omezení dat

3 Analýza požadavků a současného stavu

Tato kapitola popisuje proces analýzy požadavků předkládaných klientem a analýzy současného stavu IS/ICT, který ve vztahu k projektu Sanace dluhů v ČR panoval ve společnosti Zadavatele.

Autor se v následujících odstavcích nejprve zabývá způsobem, jakým probíhala komunikace se Zadavatelem a dále způsobem evidence požadavků v Katalogu požadavků. V několika odstavcích jsou poté zmíněna některá další zajímavá témata, jako jsou problémy týkající se rozpočtu a nutné přizpůsobení projektu, zvolená platforma vyvíjené aplikace a bezpečnostní rizika.

3.1 Evidence požadavků

3.1.1 Komunikace se Zadavatelem

Výměna informací týkajících se analýzy a vývoje aplikace probíhala po několika komunikačních kanálech včetně telefonu a elektronické pošty. Nejdůležitějším z nich se však logicky ukázal být přímý kontakt na osobních schůzkách se zástupcem Zadavatele, při kterých byly informace získávány zejména pomocí brainstormingu nad požadavky a nápady Zadavatele a pomocí model stormingu, zpravidla za použití pouze tužky a papíru, kdy použitými modely byly zpravidla free-form diagramy, jednoduché konceptuální datové modely a jednoduché procesní modely. Zaznamenané poznámky a vytvořené modely poté sloužily jako podklady pro návrh implementace a následné programování diskutované funkcionality. Dílčí modely a zaznamenané poznámky nebyly po dokončení implementace odpovídající funkcionality archivovány, změny které zachycovaly však byly vzhledem k potřebě tvorby dokumentace promítnuty do celkových modelů systému, resp. do katalogu požadavků.

3.1.2 Katalog požadavků

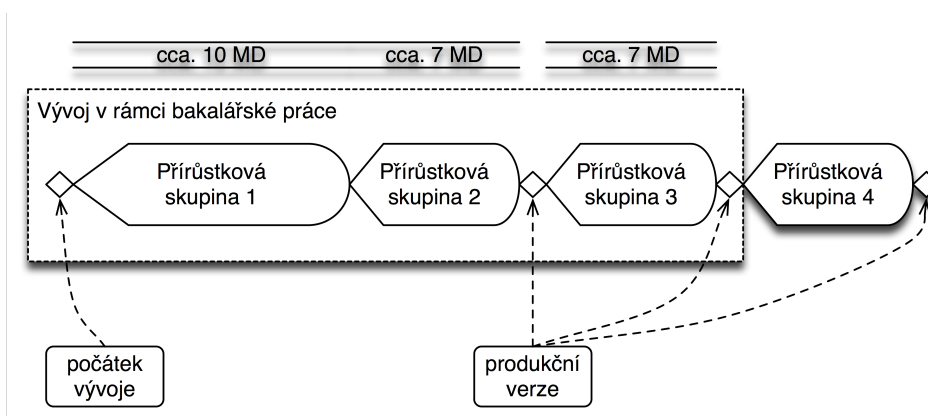
Jako hlavní dokument pro evidenci informací o požadované funkcionalitě systému sloužil katalog požadavků, jež byl průběžně doplňován po celou dobu života projektu. Pro každý z požadavků byly vedeny údaje uvedené v následujícím seznamu:

- identifikační číslo
- název
- podrobný popis
- přírůstková skupina
- typ požadavku
- poslední aktualizace
- zdroj / autor
- stav požadavku
- vazby na ostatní požadavky KP

Atribut *podrobný popis* sloužil pro evidenci detailních informací o požadavku, jako je popis ovládacích prvků aplikace vycházejících z daného požadavku, popis vyplývajících atributů datových entit na konceptuální úrovni, údaje zobrazované v rámci funkcionality daného požadavku uživateli a další. Atribut *přírůstková skupina* sloužil pro rozdělení požadavků do čtyř skupin následujícím způsobem:

- **Přírůstková skupina 1–3** — Požadavky budou implementovány v rámci bakalářské práce.
- **Přírůstková skupina 4** — Požadavky budou implementovány, avšak již mimo bakalářskou práci.

Na Obr. 6 lze vidět, že implementace první přírůstkové skupiny trvala delší dobu než implementace skupin následujících. Tato situace nastala z toho důvodu, že v rámci první přírůstkové skupiny



Obrázek 6: Katalog požadavků a verze software | Zdroj: Autor

bylo nutné položit základy pro implementaci a ne jen doprogramovávat funkcionalitu.

Atribut *Typ požadavku* sloužil k účelu vyplývajícimu ze svého názvu a v praxi nabýval tří hodnot: funkční požadavek, nefunkční požadavek, bezpečnostní požadavek. Atribut *Zdroj / autor* sloužil k evidenci informací o zdroji požadavku a datu jeho změny. Atribut *Stav požadavku* nabýval hodnot Z (požadavek zaznamenan), A (požadavek analyzován), R (rozhodnuto o realizaci), P (realizace provedena), ČR (částečně realizován), U (požadavek uzavřen) a zachycoval stav požadavku. Datum poslední změny (zadání, stavu, ...) požadavku bylo evidováno v rámci atributu *Poslední aktualizace*. Posledním z evidovaných byl atribut *Vazby na ostatní požadavky KP*, ve kterém byly výčtem atributů *Identifikační číslo* zachycovány logické vztahy mezi jednotlivými požadavky.

Katalog dále původně obsahoval atribut *Druh požadavku* zachycující informaci o původu požadavku, tedy byl-li definován v zadání, nebo je-li rozšířením zadání. Vzhledem k zvolenému agilnímu přístupu však tento atribut pozbyl významu a byl odstraněn. Změny byly při jednáních zaznamenávány do dílčích dokumentů a poté jimi byl pro dokumentační potřeby aktualizován i katalog požadavků.

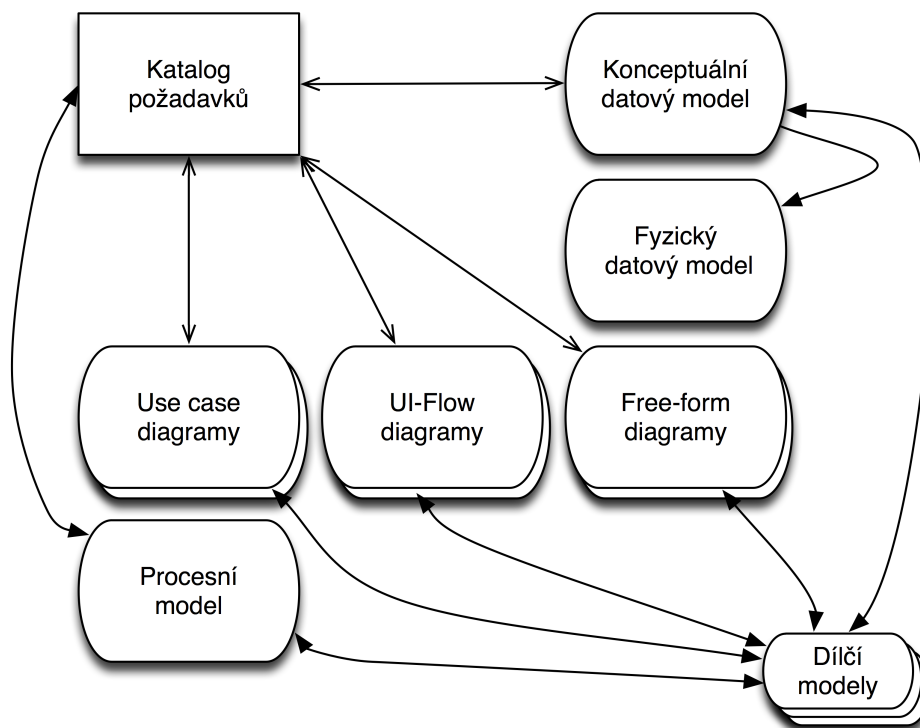
Katalog se úzce doplňoval s modely vytvářenými při zpracování projektu, ať už celkovými, nebo dílčími vytvářenými dle potřeby při jednání se zákazníkem (Obr. 7). Všechny dokumenty uvedené na obrázku byly součástí množiny Deliverables metodického předpisu AUP Product a jsou k nahlédnutí v přílohách této bakalářské práce.

Zadavateli byly na požádání přístupné veškeré dokumenty, nicméně žádán byl (a to zejména v počátku projektu) na nepravděpodobné bázi pouze Katalog požadavků.

3.2 Rozpočet a přizpůsobení řešení projektu

Hlavním omezujícím faktorem při zpracování projektu a zároveň faktorem ovlivňujícím chování jak Zadavatele tak Řešitele, byl objem finančních prostředků vyčleněných Zadavatelem pro IS/ICT podporu projektu Sanace dluhů. Zadavatel na projekt nebyl schopen vyčlenit prakticky žádné finanční prostředky, z čehož vyplývají efekty popsané v následujících odstavcích.

Nepříznivý rozpočet ovlivnil v prvé řadě zejména hardwarové a softwarové vybavení, které bylo možné využít při zpracování projektu a které bylo možné nasadit do produkčního prostředí. Jedním z praktických efektů této situace byla nutnost využít co největší množství software dostupného zdarma se všemi důsledky, které z toho vyplývají jako například nemožnost dodat některé pokročilé funkce z důvodu omezení zdarma dostupné verze vývojového prostředí. Dalším efektem byla nutnost



Obrázek 7: Evidence informací v projektu | Zdroj: Autor

přizpůsobit se již existujícímu hardware a dostupnému TASW v provozovně Zadavatele a využít ho co nejvhodnějším způsobem.

Podrobnou analýzu a popis využitého hardware a software lze nalézt v příslušné kapitole této bakalářské práce.

3.2.1 Platforma aplikace

Již v naprostém počátku řešení analýzy projektu bylo rozhodnuto, že výsledná aplikace bude naprogramována v jazyce C# pro GUI API Windows Forms. Toto rozhodnutí padlo zejména z požadavku Zadavatele na aplikaci, jež měla být vytvořena právě pro GUI API Windows Forms a vzhledem k předchozím zkušenostem Řešitele s programováním v jazyce C# a zmíněném GUI API. Tato volba poté vedla, po zhodnocení alternativ, k rozhodnutí využít pro vývoj IDE Microsoft Visual C# 2008 Express Edition.

Druhou alternativou bylo vytvoření webové aplikace, která by byla hostována na serveru v sídle Zadavatele. Jazykem využitým pro vývoj této webové aplikace by pravděpodobně bylo opět C#. Tuto možnost, jak je uvedeno výše, vyloučil Zadavatel svým požadavkem.

3.3 Bezpečnostní rizika

Při analýze projektu byla zvažována bezpečnost navrženého řešení, a to z hlediska zabezpečení dat a přístupu k nim, přičemž byly identifikovány následující rizikové aspekty vyvíjeného software.

3 ANALÝZA POŽADAVKŮ A SOUČASNÉHO STAVU

- **Bezpečnost datové základny** — Jak jsou zabezpečena data ukládaná aplikací?
- **Bezpečnost systému uživatelů software** — Jak je vyřešen systém uživatelů aplikace a jejich oprávnění vzhledem k přístupu k datům?
- **Bezpečnost komunikace** — Jak je zabezpečena komunikace po síti (zejména po Internetu)?

Způsob vypořádání se s těmito riziky je popsán v další části bakalářské práce zabývající se vlastní implementací.

4 Návrh řešení

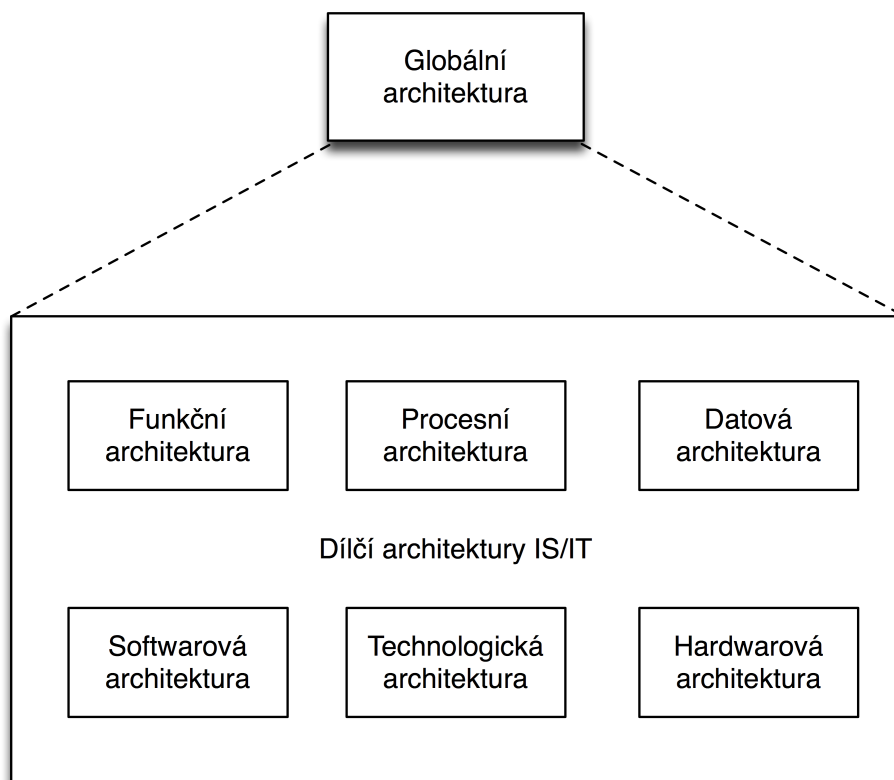
Tato kapitola zabývající se vlastním návrhem řešení ve své první části vychází z metodiky MDIS prof. Jiřího Voříška a popisuje navrženou globální architekturu IS/IT a dílčí architektury, tedy funkční, procesní, datovou, softwarovou, technologickou a hardwarovou architekturu.

V druhé části se pak tato kapitola stručně zabývá funkčními součástmi vyvinuté aplikace a následně velice podrobně popisuje hardwarové a softwarové požadavky jak vyvinutého řešení, tak i softwarové požadavky autora pro zpracování této bakalářské práce. Jako poslední je pak zmíněn návrh uživatelského rozhraní aplikace.

4.1 Architektura IS/IT

Způsob návrhu architektury byl převzat z metodiky MDIS popsané v knize Strategické řízení informačního systému systémová integrace od prof. Jiřího Voříška. Vychází z principu kdy architektura IS/IT je rozdělena do dvou úrovní (Obr. 8)) [Voříšek, 2006]:

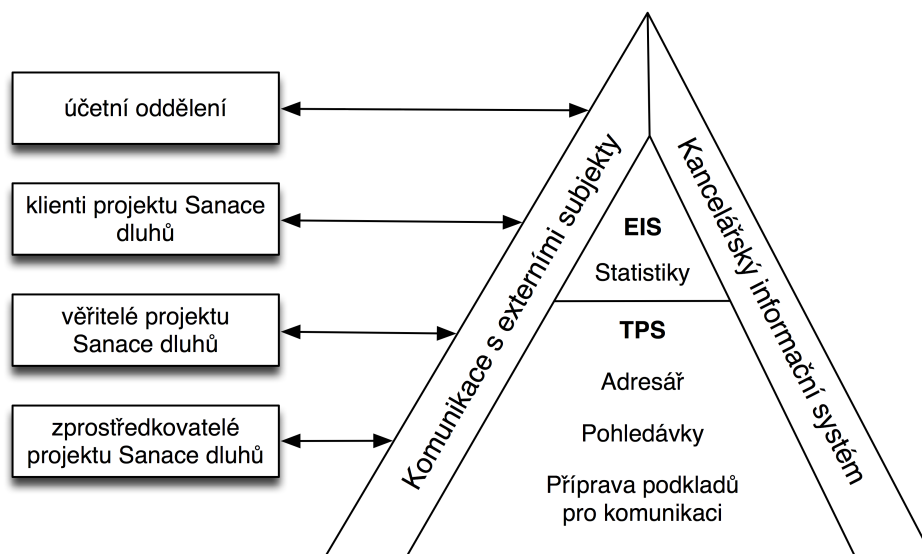
- **Globální architektura** zachycuje hrubý návrh IS/IT.
- **Dílčí architektury** rozvádějí návrh IS/IT z hlediska různých dimenzí IS.



Obrázek 8: Globální a dílčí architektury IS/IT | Zdroj: [Voříšek, 2006]

4.2 Globální architektura

Globální architektura vyvíjeného software je zachycena na Obr. 9. Vypovídá o základní funkcionalitě software v rámci EIS¹⁰, který podporuje strategické řízení projektu a TPS¹¹, který je zaměřen na podporu hlavní činnosti projektu na operativní úrovni [Voříšek, 2006].



Obrázek 9: Globální architektura vyvíjeného software | Zdroj: Autor

4.3 Dílčí architektury

4.3.1 Funkční architektura

Funkční architektura vychází z jednotlivých bloků globální architektury a postupně je rozkládá do menších skupin funkcí [Voříšek, 2006]. Vzhledem k charakteru řešeného projektu nebylo funkční architekturu možné finálně vytvořit před započítím programátorských prací. Funkční architektura sice v úvodní analýze byla navržena do vysokého stupně podrobnosti, vzhledem k neadoptování agilní metodiky ještě před započítím řešení projektu, byla však také v průběhu projektu často měněna a její finální verze byla dokončena až prakticky těsně před třetí produkční verzí software.

Funkční architekturu lze názorně popsat tabulkou, jak ukazuje prof. Voříšek [Voříšek, 2006]. Pro zachování stručnosti této bakalářské však tato tabulka není uvedena. I přes to lze informace o funkční architektuře velmi dobře získat z příložených výstupů projektu, zejména pak z diagramu funkčních součástí aplikace, katalogu požadavků, use case diagramů a UI flow diagramu.

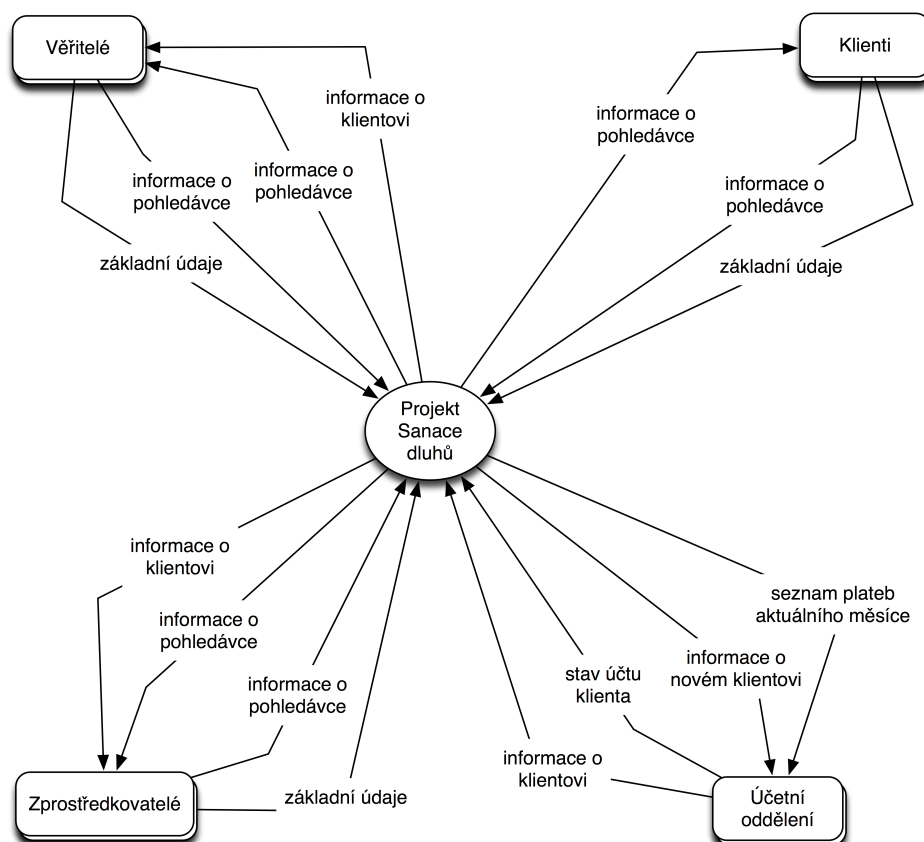
4.3.2 Procesní architektura

Procesní architektura zachycuje stav procesů v projektu Sanace dluhů po dokončení implementace navrženého systému. Při její tvorbě byl jako první vytvořen Data Flow Diagram, který popisuje způsob jakým projekt Sanace dluhů interaguje s relevantními vnějšími subjekty. Tento diagram pomohl nalézt důležité události, pro které byly vypracovány odpovídající procesy, přičemž jednotlivé události byly při modelování využity jako startovní body těchto procesů. DFD ve své konečné po-

¹⁰Executive Information System

¹¹Transaction Processing System

době je znázorněn na obrázku č. 10.



Obrázek 10: Data Flow Diagram | Zdroj: Autor

Ukázku procesního modelu lze v této bakalářské práci nalézt v kapitole 1.3. Ostatní modely nejsou v zájmu zachování stručnosti uvedeny.

4.3.3 Datová architektura

Při návrhu datové architektury je zkoumán svět v oblasti odpovídající rozměru řešeného projektu a je navrhována datová základna vytvářeného software tak, aby co nejlépe, v relevantní hloubce, odpovídala zkoumané oblasti reality. V první fázi návrhu datové architektury je vytvořen logický návrh datové základny (v tomto projektu *konceptuální datový model*) a na jeho základě je poté vytvořen fyzický návrh, kterému v řešeném projektu odpovídá *fyzický datový model* [Voříšek, 2006]. Informace o užitých modelech lze nalézt v oddílu 2.6, vlastní konceptuální datový model a fyzický datový model pak lze nalézt v přílohách této bakalářské práce.

4.3.4 Softwarová architektura

Softwarová architektura popisuje z jakých softwarových komponent se bude výsledný systém skládat a jak mezi sebou tyto komponenty budou interagovat. Hlavním bodem softwarové architektury je v řešeném projektu samozřejmě vyvíjená aplikace, která však pro svůj správný provoz potřebuje další podpůrný software (databázový server, kancelářské aplikace). Mimo hlavní aplikaci potřebují zaměstnanci projektu Sanace dluhů využívat i kancelářský software pro elektronickou ko-



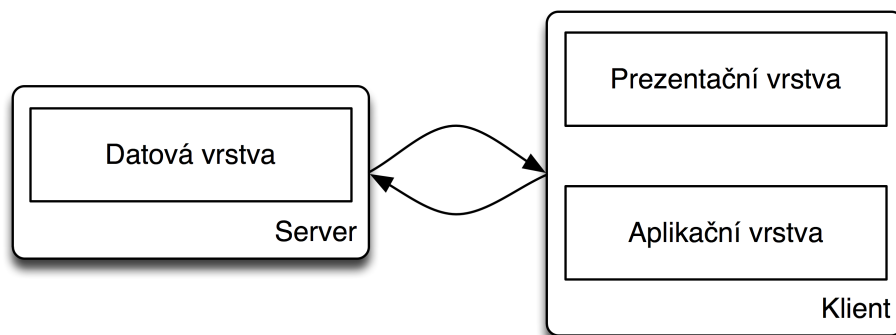
Jako technologická architektura výsledné aplikace byla zvolena Dvouvrstvá klient/server architektura typu Těžký klient s interaktivním zpracováním uživatelských požadavků. Dvouvrstvá architektura uvažuje tři základní vrstvy aplikace — datovou, aplikační a prezentační — které mohou být uspořádány do dvou vrstev následujícími dvěma způsoby.

- Způsob uspořádání dvouvrstvé architektury — Těžký klient je znázorněn na obrázku č. 12.

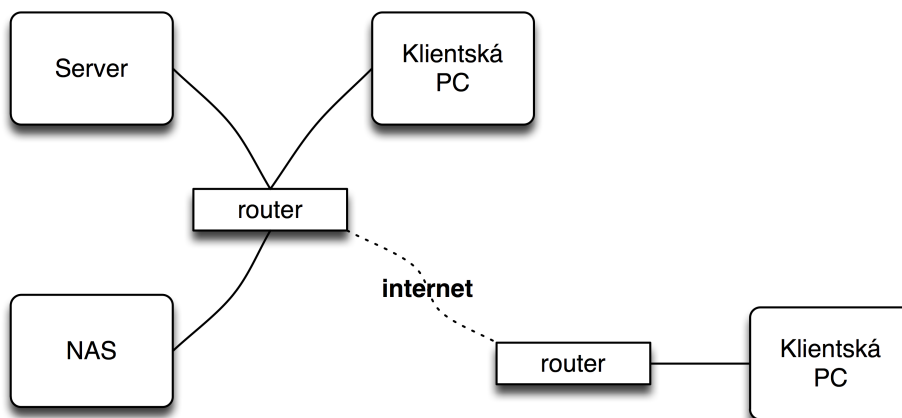
Úkolem návrhu hardwarové architektury je popsat veškerý hardware využitý v plánovaném systému a vazby mezi jeho jednotlivými prvky. Při návrhu hardwarové architektury bylo samozřejmě, vzhledem k rozpočtu projektu, třeba vyjít z již existujícího hardware, jež byl Zadavatelem zakoupen v minulosti, návrhem ideálních hardwarových komponent se tedy tato práce nezabývá.

4.4 Funkční součásti software

30



Obrázek 12: Těžký klient | Zdroj: Autor (podle [Voříšek, 2006])



Obrázek 13: Hardwarová architektura | Zdroj: Autor

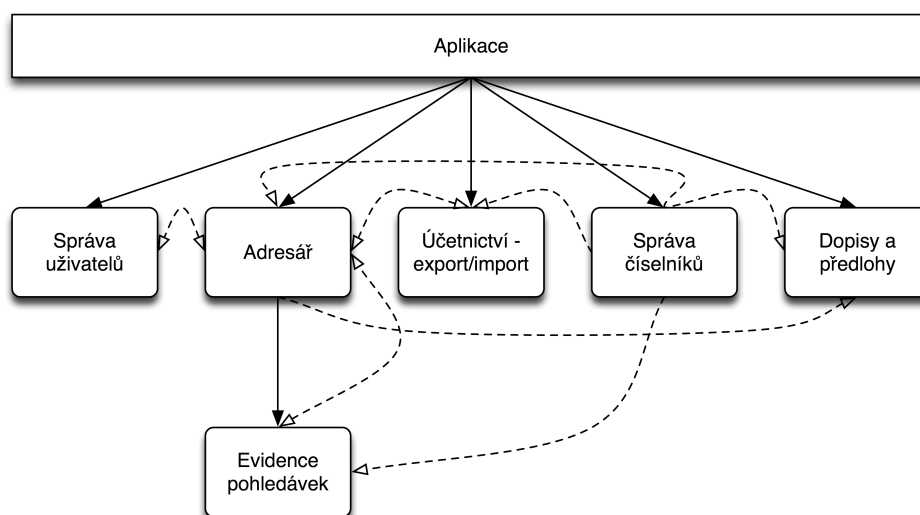
jsou spolu s jejich stručným popisem uvedeny v následujícím výčtu.

- **Adresář** — Tento modul je základním prvkem aplikace a poskytuje evidenci klientů a jejich případů, věřitelů a jejich kontaktních osob a zprostředkovatelů a jejich kontaktních osob.
- **Evidence pohledávek** — Tento modul umožňuje evidovat informace o pohledávkách a spojovat pomocí nich klienty, věřitele i zprostředkovatele, tedy subjekty definované v modulu Adresář. Jak již vyplývá z popisu, je tento modul úzce provázán právě s modulem Adresář.
- **Správa číselníků** — V modulu *Správa číselníků* je uživatelům poskytnuta možnost upravovat záznamy všech číselníků využívaných na různých místech aplikace.
- **Export/Import** — Tento modul zajišťuje obousměrnou komunikaci mezi účetním oddělením a sanačními pracovníky. Ve směru od účetního oddělení k sanačním pracovníkům je to zpracování MS Excel 2003 exportů z účetního oddělení a následná aktualizace stavů účtů klientů, v opačném směru je to pak generování seznamů plateb věřitelům. Modul dále umožňuje generovat seznam nesprávně splácejících klientů, které je třeba kontaktovat osobně.
- **Dopisy a předlohy** — Modul *Dopisy a předlohy* uživatelům umožňuje v prostředí aplikace

Microsoft Word 2007 vytvářet předlohy dopisů se zástupnými řetězci a následně na základě těchto předloh generovat dopisy libovolnému počtu klientů, věřitelů a zprostředkovatelů s informacemi specifickými pro ten který subjekt doplněnými na místo zmíněných zástupných řetězců.

- **Správa uživatelů** — Modul *Správa uživatelů* umožňuje, po přihlášení pod zvláštním administrátorským účtem, úpravy uživatelských účtů aplikace.

Obr. 14 ilustruje jednotlivé moduly aplikace a vztahy mezi nimi. Funkčně hierarchické vztahy jsou znázorněny nepřerušovanou čarou, vzájemný vliv modulů je reprezentován čarou přerušovanou.



Obrázek 14: Funkční součásti software | Zdroj: Autor

4.5 Softwarové a hardwarové požadavky projektu

4.5.1 Softwarové požadavky Řešitele pro zpracování projektu

Software využívaný Řešitelem při zpracování projektu lze rozdělit do tří hlavních skupin.

- **Software pro modelování** — Tato skupina zahrnuje software pro vytváření modelů a diagramů při analýze projektu
- **Software pro vývoj** — Do této skupiny spadá vývojové prostředí pro zvolený programovací jazyk
- **Podpůrný software** — Skupina podpůrného software zahrnuje veškerý software potřebný k tvorbě ostatních dokumentů vyžadovaných projektem

V následujících odstavcích je specificky vyjmenován a popsán software předchozích skupin spolu s problematikou licencí.

V případě software pro modelování a tvorbu diagramů byla situace jednoduše řešitelná, vzhledem k tomu, že použity byly pouze dvě aplikace a obě byly dostupné ve formě trial verze. Tyto aplikace jsou popsány v následujícím seznamu.

- **PowerDesigner** — Trial verze aplikace PowerDesigner byla využita pro tvorbu Konceptuálního datového modelu a Fyzického datového modelu.

- **OmniGraffle** — Trial verzi aplikace OmniGraffle využil Řešitel pro tvorbu různých modelů a diagramů, resp. pro jejich převedení do elektronického formátu.

Pro vývoj software řešeného projektu bylo vzhledem k volbě programovacího jazyka C# nutné zvolit jeden z odpovídajících produktů společnosti Microsoft (alternativami byly projekty Sharp-Develop a MonoDevelop, jejich využití se však zdálo nesmyslné vzhledem k možnosti použít IDE popsané v následujících větách). Z důvodů specifikovaných v licenčním ujednání software Microsoft Visual Studio 2008 dostupného studentům FIS VŠE přes MSDNAA¹² nebylo možno využít toto plnohodnotné vývojové prostředí. Jako vývojové prostředí byla tedy zvolena Express verze tohoto vývojového prostředí, kterou lze podle licenčního ujednání využít k jakémukoliv účelu a je distribuována zdarma.

Podpůrný software využívaný při zpracování projektu zahrnoval operační systémy pro provoz výše uvedeného software, kancelářské aplikace a nástroje pro sazbu dokumentů (včetně této bakalářské práce, jež je součástí dokumentace projektu). Zmíněný software zahrnuje následující aplikace.

- **Windows XP, Mac OS X 10.5** — Na oba operační systémy již Řešitel licenci vlastnil.
- **Apple iWork '08** — Apple iWork je balíkem kancelářských aplikací, jejíž součástí byly využívány pro tvorbu draftů dokumentů při zpracování projektu, pro tvorbu spreadsheetu s katalogem požadavků a pro tvorbu prezentace pro školení uživatelů aplikace. Licenci tohoto balíku již Řešitel vlastnil.
- **MacTex, TextMate** — MacTex je balík software pro sazbu dokumentů a je dostupný zdarma, TextMate je pokročilý plaintextový editor využíváný mimo jiné pro tvorbu zdrojového kódu jazyka LaTeX. Licenci aplikace TextMate již Řešitel vlastnil.

Všechny požadavky na hardware ze strany Řešitele — tedy PC pro vývoj aplikace a počítač Apple Mac pro tvorbu dokumentů — byly splněny.

4.5.2 Softwarové požadavky navrženého řešení

Softwarové požadavky výsledné aplikace vyplývají z funkčních i nefunkčních požadavků Zadavatele, z volby programovacího jazyka resp. frameworku pro vývoj aplikace a z volby databázového serveru. Prvním z požadavků byla možnost provozovat aplikaci pod systémem Windows XP, z čehož vyplývá i první požadavek na software nainstalovaný na klientských PC Zadavatele. Dalším požadavkem byla implementace tvorby šablon dopisů a doplňování definovaných polí z databáze ideálně v prostředí MS Word. Z tohoto důvodu vyvinutý software vyžaduje na klientském PC nainstalovanou aplikaci MS Word 2007. Jak již bylo zmíněno, pro naprogramování aplikace byl zvolen jazyk C#, resp. framework .NET, a proto je dalším z požadavků na nainstalovaný software i .NET Framework. Jako databázový server byl zvolen Microsoft SQL Server 2005 Express Edition, jehož instalace je proto požadavkem pro serverové PC.

Instalace Apache HTTP Serveru vychází ze zvláštního požadavku¹³ Zadavatele na dostupnost centrálního úložiště souborů nejen uvnitř sítě v sídle Zadavatele (jak tomu bylo původně), ale i přes Internet. Pro splnění tohoto požadavku byl na serverové PC nainstalován výše zmíněný HTTP server, který byl následně nakonfigurován tak, aby umožnil práci se soubory umístěnými na NAS ve vnitřní síti pomocí protokolu WebDAV s SSL.

Požadavky výsledné aplikace na software jsou shrnuty v Tab. 3)¹⁴.

¹²Microsoft Developer Network Academic Alliance

¹³zvláštním požadavkem rozumíme požadavek netýkající se vyvíjené aplikace

¹⁴tabulka neobsahuje všechny druhotné požadavky v ní uvedeného software

software	zdroj požadavku	ovlivněná PC
MS Windows XP nebo vyšší ¹⁵	nefunkční požadavek	server i klientská PC
MS Word 2007	funkční požadavek	klientská PC
MS SQL Server 2005 Express Edition	výsledná aplikace	server
Apache HTTP Server 2.2.10	zvláštní požadavek	server
MS .NET Framework 3.5	výsledná aplikace	klientská PC
MS .NET Framework 2.0	MS SQL Server 2005	server

Tabulka 3: Softwarové požadavky vyvinutého řešení

Licence Windows XP pro server a klientská PC zajistil Zadavatel, stejně tak jako licence pro balík Microsoft Office Standard 2007. Aplikace Microsoft SQL Server 2005 Express Edition a výše uvedené verze .NET Frameworku jsou distribuovány zdarma.

4.5.3 Hardwarové požadavky vyvinutého řešení

Minimální a doporučené hardwarové požadavky vyplývají ze softwarových požadavků uvedených v předchozím odstavci. Tabulka 4¹⁶ uvádí minimální a doporučené hardwarové konfigurace pro server i pro klientská PC.

PC	typ HW	minimální konfigurace	doporučená konfigurace
server	RAM	192 MB	512 MB
	CPU	Pentium 3 Compatible 500 MHz	1,5 GHz
	HDD	20 GB	> 20 GB
klientská PC	RAM	256 MB	1 GB
	CPU	500 Mhz	1,5 Ghz
	HDD	20 GB	> 20 GB

Tabulka 4: Hardwarové požadavky vyvinutého řešení

Výše uvedené požadavky na hardware byly Zadavatelem splněny v doporučené konfiguraci pro server i všechna klientská PC. Veškerý hardware musel být, vzhledem k rozpočtu, vybrán z již zakoupeného hardware ve společnosti Zadavatele.

4.6 Uživatelské rozhraní aplikace

Návrh uživatelského rozhraní probíhal po celou dobu řešení projektu a za častých konzultací se zástupcem Zadavatele, tedy s jedním z budoucích uživatelů aplikace. Při vlastním návrhu GUI vycházel autor z dokumentu Microsoft Inductive User Interface Guidelines [Microsoft Corporation, 2001] a z vlastních zkušeností s aplikacemi pro Microsoft Windows a Apple Mac OS X. Cílem této aktivity bylo vytvořit takové grafické uživatelské rozhraní, které by dokázal pochopit a užívat běžný uživatel i bez uživatelské příručky.

Přístup zajišťující co nejlepší pochopitelnost grafického uživatelského rozhraní zahrnoval zejména následující techniky:

¹⁶tabulka vychází z předpokladu nainstalovaného operačního systému Windows XP a HW požadavků poskytnutých společností Microsoft pro software uvedený v předchozích požadavcích a z doporučení Řešitele

- Využití standardních ovládacích prvků aplikací pro operační systém Microsoft Windows XP
- Koncentrace ovládacích prvků vyvolávajících akci vždy ve stejné části formuláře
- Využití názorných ikon pro ovládací prvky vyvolávající akci
- Aplikace K.I.S.S. principu na návrh obrazovek — všechny obrazovky byly tvořeny tak, aby byly co nejjednodušší na pochopení a vždy se zabývaly řešením pouze jednoho úkolu

5 Vývoj navrženého řešení

Vzhledem k tomu, že tato bakalářská práce nemá za cíl podrobně rozebírat vlastní programování vyvinuté aplikace, je tato kapitola poměrně stručná a popisuje pouze vybraná zajímavá témata. Nejprve se zabývá otázkou datové základny, přenesením jejího návrhu z modelů do prostředí databázového serveru a vrstvou uložených procedur, které byly na straně serveru naprogramovány.

V dalších odstavcích pak popisuje zvolené IDE a programovací jazyk využívané pro vývoj výsledné aplikace, zabývá se problematikou zabezpečení aplikace a v poslední podkapitole popisuje naplnění jednoho z požadavků Zadavatele — komunikaci klientské aplikace se serverem po síti.

5.1 Datová základna

5.1.1 Přenesení Fyzického datového modelu do prostředí MS SQL Server

Vzhledem k tomu, že při vývoji aplikace byla využita technika AMDD, tvorbě elementů databáze vždy předcházelo modelování. Díky tomuto faktu bylo možné využít funkcionality aplikace Power-Designer, ve které byly vytvářeny datové modely, která umožňuje vygenerování prvků databáze specifikovaných v modelu přímo do prostředí serveru. Díky modelování při návrhu datové základny bylo tedy možné ušetřit desítky hodin, které by jinak bylo třeba strávit ručním psaním DDL SQL příkazů.

5.1.2 Vrstva uložených procedur

Z důvodu eliminace dynamického SQL ve zdrojovém kódu aplikace bylo vytvořeno několik desítek uložených procedur v jazyce T-SQL zajišťující provádění veškerých SELECT, INSERT, UPDATE a DELETE příkazů. Výsledkem tohoto kroku tedy je, že veškerá komunikace mezi klientem a databázovým serverem je zajištěna striktně pouze pomocí volání uložených procedur s parametry. Využití uložených procedur však nese mnoho dalších výhod zejména z hlediska testování aplikace, náročnosti na množství dat přenášných po síti a zabezpečení¹⁷.

5.2 Zvolené IDE a programovací jazyk

Jazyk C# je objektově orientovaný programovací jazyk vyvíjený společností Microsoft a je jedním z programovacích jazyků pro tvorbu aplikací pro .NET framework. Framework .NET a tím i C# jsou původně orientovány pouze na platformu Microsoft Windows, existuje však projekt Mono, který umožňuje portování .NET aplikací na další platformy, nicméně vzhledem k faktu, že bude využívána pouze v prostředí Microsoft Windows, nebyla aplikace pod projektem Mono testována. Jazyk C# je specifikován v [ECMA-334, 2006].

Visual C# 2008 Express Edition je IDE pro jazyk C# vyvíjené a poskytované zdarma společností Microsoft. Toto IDE poskytuje prakticky veškeré nástroje¹⁸ vhodné pro vývoj aplikace v jazyce C#, zejména včetně Windows Forms Designeru pro snadný návrh grafického uživatelského rozhraní a Dataset Designeru pro návrh a úpravy datasetu využívaného pro komunikaci s databází.

5.3 Zabezpečení aplikace

5.3.1 Uživatelské účty aplikace

Pro co největší bezpečnost dat se kterými aplikace pracuje bylo třeba nalézt takový systém správy uživatelů, ve kterém jsou data co nejlépe chráněna jak před neoprávněnými zásahy oprávněných uživatelů aplikace (z hlediska odstraňování záznamů), tak před vnějšími útočníky snažícími se o vniknutí do aplikace za využití případné, při testování neodhalené chyby.

¹⁷při využití dynamického SQL se lze při odposlechu komunikace dozvědět informace o struktuře DB

¹⁸kromě Visual Studio Tools for Office, které bylo možné využít pro zlepšení uživatelského komfortu

Vzhledem k tomu, že data jsou ukládána v bezpečném a stabilním prostředí databázového serveru MS SQL Server 2005 Express Edition, bylo břemeno správy uživatelů přesunuto na databázový server způsobem, kdy se jeden uživatel aplikace rovná jednomu uživateli databázového serveru; modul správy uživatelů, který je součástí aplikace, tak poskytuje pouze grafické uživatelské rozhraní pro tuto funkcionalitu z prostředí aplikace. Každý z uživatelů¹⁹ má na objektech databáze práva pouze na příkazy SELECT, INSERT a UPDATE a nemůže tedy data ovlivnit jiným způsobem, než je mu povoleno ani pomocí jazyka SQL. Pouze administrátorský účet má dále přidělena práva na úpravy uživatelských účtů.

Veškerá funkcionalita, na kterou nemají běžné účty přidělena práva, je v GUI aplikace skryta.

5.3.2 Komunikace po síti

Jedním z požadavků Zadavatele byla možnost využívat aplikaci i mimo jeho sídlo, tedy komunikovat s databázovým serverem prostřednictvím sítě Internet. Obecné zajištění tohoto požadavku nebylo vzhledem k implementaci klient/server architektury problematické, bylo však také třeba zajistit bezpečnost dat při jejich přenosu. Pro zabezpečení dat na cestě mezi klientským počítačem a serverem bylo využito vestavěné funkcionality MS SQL Server 2005 Express Edition, která umožňuje šifrování kanálu pomocí SSL²⁰ a self-signed certifikátu [Lee, 2005].

¹⁹kromě fixního administrátorského účtu, který má i právo DELETE

²⁰Secure Sockets Layer — komunikační protokol

6 Předání řešení uživatelům

Tato kapitola stručně popisuje problematiku předání vyvinutého řešení uživatelům z hlediska kontroly jeho kvality a dále pak z hlediska vyškolení uživatelů k jeho používání. V rámci části zabývající se akceptačním řízením jsou popsány zejména způsoby a kritéria jeho ukončení a klasifikace vad.

6.1 Akceptační řízení

Akceptačním řízením rozumíme proces kdy je skutečná funkcionální hodnoceního systému porovnávána s požadovanou funkcionalitou. Jeho výstupem je *protokol o akceptaci*, který obsahuje seznam vad (vadou rozumíme nesoulad mezi skutečnou a požadovanou funkcionalitou), jejich klasifikaci, návrh na řešení a termín implementace uvedeného řešení.

Akceptační řízení bylo v řešeném projektu prováděno vždy na základě jednotlivých přírůstkových skupin a produkčních verzí software — pro první dvě přírůstkové skupiny zároveň a samostatně pro třetí přírůstkovou skupinu. Protokoly o akceptaci ze dvou provedených akceptačních řízení lze nalézt v přílohách této bakalářské práce.

6.1.1 Způsoby ukončení akceptačního řízení

Akceptační řízení bylo vždy ukončeno jedním ze stavů níže uvedeného výčtu. Jednotlivé stavy jsou odvozeny od počtu nalezených vad a jejich typů.

- **Akceptováno**
- **Akceptováno s výhradami** — Výstupy odpovídající dané fázi vykazují vady, které však nebrání zahájení následující fáze. Je proveden soupis výhrad s termíny jejich odstranění. Určí se termín opakované akceptace, do kterého budou všechny zjištěné chyby vyřešeny.
- **Neakceptováno** — Výstupy odpovídající dané fázi vykazují množství vad, které brání zahájení následující fáze. Je určen termín jednání, na kterém budou znovu podrobně diskutovány požadavky dané fáze, bude proveden soupis rozdílů mezi aktuálním požadovaným stavem s termíny jejich odstranění a určí se termín opakované akceptace, do kterého budou všechny zjištěné chyby vyřešeny.

6.1.2 Klasifikace vad

Každá z vad nalezených v průběhu akceptačního řízení byla klasifikována do jedné ze tří skupin uvedených v následujícím seznamu.

- **Vada kategorie A** — Aplikace není použitelná ve svých základních funkcích nebo se vyskytuje funkční závada znemožňující činnost aplikace. Tento stav může ohrozit běžný provoz Zadavatele, případně může způsobit větší finanční nebo jiné škody. Vada bude odstraněna v nejkratší možné lhůtě s ohledem na její povahu, dopad na činnost Zadavatele a způsob ukončení akceptačního řízení.
- **Vada kategorie B** — Funkčnost aplikace je ve svých funkcích degradována tak, že tento stav omezuje běžný provoz Zadavatele - Zhotovitel navrhne takové alternativní řešení nebo způsob obnovení funkčnosti aplikace, které umožní plné odstranění vady, nebo snížení její klasifikace na kategorii C. Vada bude odstraněna v nejkratší možné lhůtě s ohledem na její povahu, dopad na činnost Zadavatele a způsob ukončení akceptačního řízení.
- **Vada kategorie C** — Ostatní drobné vady, které nespádají do kategorií A a B — Vada může být odstraněna v další plánované verzi počítačového programu, nebo v jiném termínu, vždy však určeném v daném akceptačním řízení.

6.1.3 Kritéria ukončení akceptačního řízení

Následující výčet znázorňuje vliv kritérií klasifikace vad na způsob ukončení akceptačního řízení.

- **1 a více chyb typu A** — Presentovaný výstup *není akceptován*.
- **1 a více chyb typu B & 0 chyb typu A** — Presentovaný výstup je *akceptován s výhradami*.
- **0 chyb kategorie A i B** — Presentovaný výstup je *akceptován*.

6.2 Školení uživatelů

Pro úspěšné uvedení vyvinutého software do provozu je třeba dostatečně vyškolit uživatele pro práci s ním. V řešeném projektu proběhlo školení uživatelů pomocí prezentace a následného osobního předvedení funkcionality aplikace každému uživateli zvlášť²¹. Doplnkem znalostí, které uživatelé získali při školení je uživatelská příručka.

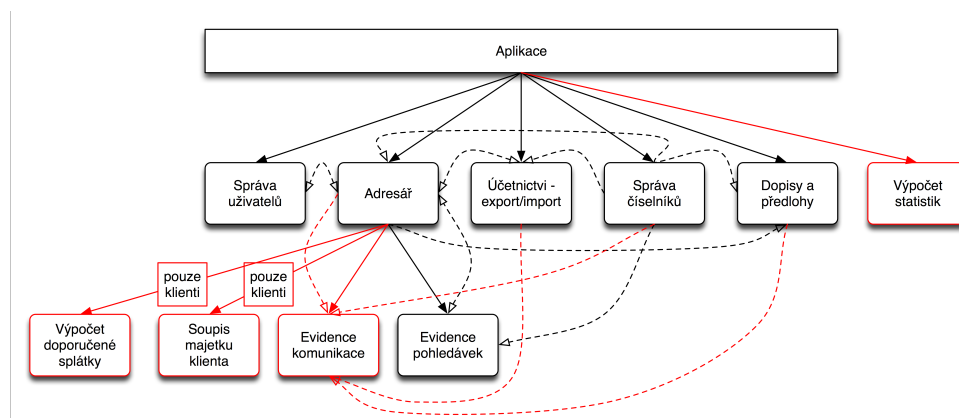
²¹tento přístup bylo samozřejmě možné využít pouze vzhledem k tomu, že počet uživatelů aplikace není vysoký

7 Rozšíření aplikace a další funkcionalita

Rozšíření aplikace mimo rámec této bakalářské práce jsou evidovány v katalogu požadavků jako požadavky čtvrté přírůstkové skupiny. Některé z těchto požadavků již byly analyzovány, některé z nich jsou pouze ve fázi zaevidování a budou tedy analyzovány později. K oddělení čtvrté přírůstkové skupiny od předchozích tří došlo zejména z důvodů časových. Časovými důvody rozumějme, že do data odevzdání této bakalářské práce nebylo možné při zachování stejné úrovně kvality dodat funkcionalitu obsaženou ve všech čtyřech přírůstkových skupinách. Na tomto místě je dobré zmínit, že od data přijetí zadání bakalářské práce až do data termínu odevzdání uběhl takřka přesně jeden rok.

Lze samozřejmě očekávat, že po nasazení aplikace do ostrého provozu bude Zadavatelem poptána nová funkcionalita, jejíž potřeba nebyla při analýze a testování prvních tří přírůstkových skupin známa. Tyto nově vzniklé požadavky budou řešeny stejně jako předchozí, tedy způsobem popsaným v této bakalářské práci.

Následující text stručně shrnuje a komentuje požadavky čtvrté přírůstkové skupiny.



Obrázek 15: Funkční součásti software včetně 4. přírůstkové skupiny | Zdroj: Autor

Požadavek *evidence komunikace s klienty a věřiteli* byl již analyzován a v katalogu požadavků je tedy podrobněji rozepsán. Cílem modulu, který z tohoto požadavku vznikne je umožnit uživatelům evidovat veškerou komunikaci s klienty a věřiteli, a to dvěma způsoby. Prvním způsobem je ruční vkládání záznamů o komunikaci, které je podpořeno příslušným číselníkem typů komunikace. Druhým způsobem je pak automatické vkládání záznamů o komunikaci, jehož záznamy má uživatel možnost pouze prohlížet (a odstraňovat je-li uživatel přihlášen pod administrátorským účtem). Automatické vložení záznamu o komunikaci nastane například v případě aktualizace stavu účtu klienta.

Požadavek *soupis majetku klienta* zatím nebyl analyzován a v době psaní tohoto odstavce (tedy před spuštěním projektu Sanace dluhů a ustálením jeho metodiky) nelze určit, bude-li tento požadavek implementován. Tento požadavek je tedy pouze nápadem vzešlým z brainstormingu se zástupcem Zadavatele.

Dalším, ještě neanalyzovaným požadavkem je *výpočet doporučené splátky*. Tento požadavek by do okna věnovaného klientovi měl přidat jednoduchou tabulku pro výpočet orientační doporučené splátky.

Následujícím požadavkem, který je evidován ve čtvrté přírůstkové skupině je *výpočet statistik*. Tento požadavek ještě také nebyl analyzován, o jeho implementaci je však již rozhodnuto. Pojem statistiky zde představuje sadu sum finančních částek, četnosti výskytů jednotlivých číselníkových

7 ROZŠÍŘENÍ APLIKACE A DALŠÍ FUNKCIONALITA

položek v záznamech a dalších údajů.

Poslední požadavek, který zde zmíníme je *rozšíření evidovaných informací o klientovi*. Tento požadavek byl přidán poměrně pozdě v průběhu zpracování této bakalářské práce a popisuje rozšíření datového modelu a příslušné části aplikace o informace specifikované v dodaném dokumentu MS Word.

Obrázek č. 15 prezentuje funkční součásti vyvinutého software po implementaci zatím známých požadavků 4. přírůstkové skupiny. Nové elementy oproti původnímu modelu jsou vyznačeny červeně, stejně tak i datové vazby (jsou-li známy, tedy byl-li požadavek analyzován).

8 Dokumentace systému

Podstatnou a neopomenutelnou součástí každého projektu vývoje softwarového řešení je tvorba dokumentace pro koncové uživatele a pro programátory, kteří by v budoucnu eventuelně mohli projekt převzít a dále ho rozvíjet.

Ve vztahu k úspěšnosti implementace nového softwarového řešení je tvorba dokumentace pro koncové uživatele, vedle jejich zaškolení do používání výsledného produktu, klíčová vzhledem k tomu, že po zaškolení uživatelů již školitel nemusí být ve chvíli potřeby konzultace dostupný on-site, nebo po některém z komunikačních kanálů a uživatelé tak musí mít zdroj informací, na který se mohou v případě problémů obrátit. I ve chvíli kdy je po určitém komunikačním kanálu k dispozici uživatelská podpora je však správné dokumentaci pro koncové uživatele vytvořit vzhledem k tomu, že je-li dosti podrobná, dokáže odpovědět na běžné dotazy, které by jinak uživatelskou podporu zbytečně vytěžovaly.

Dokumentaci pro programátory je třeba vytvořit zejména v ohledu na budoucí vývoj aplikace. Autor, který vytvořil danou verzi software se nijak nemusí podílet na vývoji jeho dalších verzí a je proto jeho povinností vytvořit dostatečný popis kódu, který vytvořil tak aby ho programátoři, kteří se na vývoji budou podílet v budoucnosti, mohli co nejsnadněji využít a ve vývoji pokračovat. Tvorba programátorské dokumentace je však samozřejmě důležitá i z hlediska udržitelnosti vývoje projektu v neměnném týmu. Pracuje-li na vývoji software více osob a/nebo naroste-li projekt do relativně větších rozměrů, je tvorba dokumentace klíčová pro orientaci v kódu.

8.1 Uživatelská příručka

Uživatelská příručka je druhem dokumentace pro koncové uživatele, která byla využita v řešeném projektu. Při její tvorbě byly jako hlavní zdroj informací využity Use Case scénáře vytvářené v průběhu řešení projektu a jejich rozvedení na User Interface Flow Diagramy vzhledem k tomu, že tyto dvě skupiny modelů nejlépe zachycují práci uživatelů s aplikací a je tak největší šance, že se pomocí nich podaří zachytit nejasnosti, na které uživatelé mohou narazit při jejím používání. Pro co největší názornost byly do příručky zahrnuty snímky obrazovek a již zmíněné diagramy. Její součástí jsou také *poznámky k vydání* pro aktuální verzi software.

8.2 Programátorská dokumentace

Programátorská dokumentace řešeného projektu se skládá z několika částí, které odrážejí postup vývoje software až do uzavření třetí přírůstkové skupiny. Následující výčet uvádí prvky dokumentace a důvod jejich zařazení.

- **Konceptuální datový model** — Z důvodu snadné orientace v entitách a jejich attributech uvažovaných při úpravách datové základny.
- **Fyzický datový model** — Pro názorný pohled na strukturu databáze bez nutnosti provádět reverse engineering.
- **Dokumentace C# kódu** — Z důvodu umožnění snadné orientace v třídách, metodách a dalších prvcích zdrojového kódu.
- **Dokumentace T-SQL procedur** — Pro orientaci v uložených procedurách a jejich attributech v prostředí MS SQL Server 2005 Express Edition.

Kromě prvků dokumentace z předchozího seznamu, které tvoří oficiální programátorskou dokumentaci, jsou budoucím vývojařům software samozřejmě k dispozici všechny ostatní modely přiložené k této bakalářské práci.

9 Závěr

Cílem projektu bylo navrhnout a realizovat IS/ICT podporu projektu Sanace dluhů řešeného Zadavatelem a měřítkem splnění cílů této bakalářské práce byla úspěšnost návrhu řešení IS/ICT a úspěšnost jeho následné implementace. Pojmy návrh a implementace zahrnují sadu činností, z nichž ty nejvýznamnější jsou uvedeny v následujícím výčtu:

- zhodnocení metodik využitelných při řešení projektu a výběr nejvhodnější z nich
- přizpůsobení zvolené metodiky
- analýza původního stavu IS/ICT Zadavatele
- shromáždění požadavků Zadavatele a jejich analýza
- návrh IS/ICT řešení
- vývoj IASW
- testování IASW
- akceptace IASW
- konfigurace hardware využívaného IASW a podpurným SW
- instalace a konfigurace IASW, podpurného SW a dalších aplikací
- zaškolení uživatelů

Vzhledem k tomu, že navržené řešení bylo úspěšně uvedeno do provozu a Zadavatel ho využívá bez významných připomínek, lze říci, že cílů stanovených před počátkem zpracování projektu bylo dosaženo.

Vedle míry dosažení stanovených cílů je důležitým ukazatelem úspěšnosti bakalářské práce i míra využitelnosti jejích výstupů. Mezi výstupy této bakalářské práce a projektu v jejím rámci řešeného převažují zejména výstupy praktického rázu, přičemž využitelnost těchto výstupů závisí na kvalitě jejich zpracování a zejména na míře, kterou podporují vizi a požadavky Zadavatele. V předchozím odstavci již bylo uvedeno, že úspěšnost implementace navrženého řešení byla vysoká a že Zadavatel je s tímto řešením bez větších připomínek spokojen — z tohoto faktu lze odvodit, že výstupy této bakalářské práce jsou velice dobře využitelné.

Toto tvrzení je dále podporováno faktem, že hlavní výstup, tedy vyvíjená aplikace úspěšně prošel dvěma akceptačními řízeními, která jej prověřovala z hlediska plnění požadavků definovaných Zadavatelem.

Závěrem bych rád uvedl, že pro mne jako autora bylo zpracování této bakalářské práce náročnou, avšak velice zajímavou zkouškou znalostí, které jsem získal při bakalářském studiu oboru Informatika na Fakultě informatiky a statistiky Vysoké školy ekonomické v Praze a následném nezbytném samostudiu. Znalosti potřebné k úspěšnému zpracování všech výstupů, které bylo nutné v rámci projektu vytvořit zasahovaly do prakticky všech informatických předmětů, které jsem měl možnost v posledních letech studovat od Vývoje informačních systémů, přes Databáze a Vývoj aplikací na platformě .NET až po Elektronickou sazbu a publikování. Přínos této práce je pro mne tedy nesporný — díky jejímu zpracování jsem měl možnost otestovat znalosti získané studiem na reálném projektu a zároveň je tím prohloubit. Rozhodně si také cením faktu, že výstupy této práce budou využívány v praxi a vzhledem k projektu, ke kterému se vztahují doufejme i pomohou změnit naši společnost k lepšímu.

Reference

- [Voříšek, 2006] VOŘÍŠEK, Jiří. Strategické řízení informačního systému a systémová integrace. 1. vyd. Praha : Management Press, s.r.o., 2006. 323 s. ISBN 80-85943-40-9.
- [Chlapek, Stanovská, Řepa, 2005] CHLAPEK, Dušan, STANOVSKÁ, Iva, ŘEPA, Václav. Vývoj informačních systémů : pracovní sešit ke cvičením. Praha : Oeconomica, 2005. ISBN 8024509776.
- [Buchalcevoová, Pavlíčková, Pavlíček, 2007] BUCHALCEVOVÁ, Alena, PAVLÍČKOVÁ, Jarmila, PAVLÍČEK, Luboš. Áklady softwarového inženýrství : materiály ke cvičení. Praha : Oeconomica, 2007. ISBN 9788024512709.
- [Voříšek, Pavelka, Vít, 2003] VOŘÍŠEK, Jiří, PAVELKA, Jan, VÍT, Miroslav. Aplikační služby IS/ICT formou ASP : proč a jak pronajímat infromatické služby. Praha : Grada, 2003. ISBN 8024706202.
- [Buchalcevoová, 2005] BUCHALCEVOVÁ, Alena. Metodiky vývoje a údržby informačních systémů. Praha : Grada, 2005. ISBN 8024710757.
- [Tu, Fr, Ru, 2002] TURK, Dan, FRANCE, Robert, RUMPE, Bernhard. Limitations of Agile Software Processes. [s.l.] : [s.n.], 2002. 1 s. Dostupný z WWW: <<http://www4.informatik.tu-muenchen.de/~rumpe/ps/XP02.Limitations.pdf>>.
- [Microsoft Corporation, 2001] Microsoft Corporation. Microsoft Inductive User Interface Guidelines [online]. [2001] , February 9, 2001 [cit. 2008-10-18]. Dostupný z WWW: <<http://msdn.microsoft.com/en-us/library/ms997506.aspx>>.
- [ECMA-334, 2006] Standard ECMA-334 : C# Language Specification [online]. Ecma International, [2006] , June 2006 [cit. 2008-10-17]. Dostupný z WWW: <<http://www.ecma-international.org/publications/files/ECMA-ST/ECma-334.pdf>>.
- [Jeffries, 2001] JEFFRIES, Ronald. What is Extreme Programming? [online]. c1999-2008 , 11/08/2001 [cit. 2008-10-12]. En. Dostupný z WWW: <<http://www.xprogramming.com/xpmag/whatisxp.htm>>.
- [Coffin, Lane, 2006] COFFIN, Rod, LANE, Derek. A Practical Guide to Seven Agile Methodologies, Part 1 [online]. c2008 , October 11, 2006 [cit. 2008-10-12]. En. Dostupný z WWW: <<http://www.devx.com/architect/Article/32761/0/page/1>>.
- [De Luca, 2003] DE LUCA, Jeff. Issue 4 - Q&A About Feature Milestones [online]. [2002-] , Fri, 05/09/2003 - 03:12 [cit. 2008-10-12]. En. Dostupný z WWW: <<http://www.featuredrivendevelopment.com/node/548>>.
- [Ambler, 2008] AMBLER, Scott W.. The Agile Unified Process (AUP) [online]. c2005-2008 , Last updated: August 16, 2008 [cit. 2008-10-12]. En. Dostupný z WWW: <<http://www.ambysoft.com/unifiedprocess/agileUP.html>>.
- [Ambler, 2007] AMBLER, Scott W.. User Interface Flow Diagrams (Storyboards) [online]. c2003-2007 [cit. 2008-10-11]. Dostupný z WWW: <<http://www.agilemodeling.com/artifacts/uiFlowDiagram.htm>>.
- [Lee, 2005] LEE, Il-Sung. SSL in SQL Server 2005 [online]. c2008 , Published Friday, August 05, 2005 8:48 PM [cit. 2008-10-18]. Dostupný z WWW: <<http://blogs.msdn.com/dataaccess/archive/2005/08/05/448401.aspx>>.

-
- [Robinson, 2003] ROBINSON, Simon, et al. C# Programujeme profesionálně. [s.l.] : Computer Press, 2003. 1160 s. ISBN 80-251-0085-5.
- [SPJ, 2008] Sdružení pro probaci a mediaci v justici. Sdružení pro probaci a mediaci v justici, o.s. - Úvod [online]. c1996-2008 [cit. 2008-11-08]. Dostupný z WWW: <<http://www.spj.cz>>.
- [Zákon 182/2006 Sb., 2006] Zákon č. 182/2006 Sb. o úpadku a způsobech jeho řešení (insolvenční zákon) [online]. 2006 [cit. 2008-11-09]. Dostupný z WWW: <<http://web.mvcr.cz/archiv2008/sbirka/2006/sb062-06.pdf>>. ISSN 1211 - 1244.
- [Komínek, 2008] KOMÍNEK, Stanislav. Sanace dluhů [online]. 2008 [cit. 2008-11-09]. Prezentace Microsoft PowerPoint. Dostupný z WWW: <<http://equal.spj.cz/admin/get.file.php?src=doc.00363.ppt>>.
- [Komínek, 2008 (2)] KOMÍNEK, Stanislav. Sanace dluhů - švýcarský nástroj reintegrace pachatelů trestných činů. Via Iuris : Bulletin pro právo ve veřejném zájmu ONLINE [online]. 2008 [cit. 2008-11-22]. Dostupný z WWW: <<http://viaiuris.pilaw.cz/index.php?p=msg&id=221>>.
- [Pokorný, Halaška, 2004] POKORNÝ, Jaroslav, HALAŠKA, Ivan. Databázové systémy. 2. přeprac. vyd. Praha : Vydavatelství ČVUT, 2004. 148 s. ISBN 80-01-02789-9.
- [Encyclopædia Britannica, 2008] Encyclopædia Britannica. Information system [online]. 2008 [cit. 2008-12-07]. Dostupný z WWW: <<http://www.britannica.com/EBchecked/topic/287895/information-system>>.
- [Probační a mediační služba ČR, 2008] Probační a mediační služba ČR. PROBACE - dohled nad podmíněně odsouzenými [online]. [2008] [cit. 2008-12-09]. Dostupný z WWW: <http://www.pmscr.cz/scripts/index.php?id_nad=36>.
- [SPJ, o.s., 2008] Sdružení pro probaci a mediaci v justici, o.s.. Slovníček pojmů [online]. c1996-2008 [cit. 2008-12-09]. Dostupný z WWW: <<http://www.spj.cz/index.php?page=63>>.

Seznam obrázků

1	Proces sanace dluhů Zdroj: Autor (podle informací od Mgr. Stanislava Komínka)	9
2	Životní cyklus AUP Zdroj: [Ambler, 2008]	16
3	Životní cyklus AMDD Zdroj: [Ambler, 2008]	18
4	Příklad Use Case Diagramu Zdroj: Autor	20
5	Příklad User Interface Flow Diagramu Zdroj: Autor	21
6	Katalog požadavků a verze software Zdroj: Autor	24
7	Evidence informací v projektu Zdroj: Autor	25
8	Globální a dílčí architektury IS/IT Zdroj: [Voříšek, 2006]	27
9	Globální architektura vyvíjeného software Zdroj: Autor	28
10	Data Flow Diagram Zdroj: Autor	29
11	Softwarová architektura Zdroj: Autor	30
12	Těžký klient Zdroj: Autor (podle [Voříšek, 2006])	31
13	Hardwarová architektura Zdroj: Autor	31
14	Funkční součásti software Zdroj: Autor	32
15	Funkční součásti software včetně 4. přírůstkové skupiny Zdroj: Autor	40
16	Use Case diagram — oblast Obecné činnosti Zdroj: Autor	54
17	Use Case diagram — oblast Uživatelé Zdroj: Autor	54
18	Use Case diagram — oblast Klienti Zdroj: Autor	55
19	Use Case diagram — oblast Věřitelé Zdroj: Autor	56
20	Use Case diagram — oblast Zprostředkovatelé Zdroj: Autor	57
21	Use Case diagram — oblast Pohledávky Zdroj: Autor	58
22	Use Case diagram — oblast Dopisy Zdroj: Autor	59
23	Use Case diagram — oblast Účetnictví Zdroj: Autor	59
24	Přihlašovací okno aplikace Zdroj: Autor	69
25	Hlavní okno aplikace — Adresář Zdroj: Autor	69
26	Okno aplikace — Editace číselníku Zdroj: Autor	70

27	Okno aplikace — Souhrn informací o klientovi Zdroj: Autor	70
28	Okno aplikace — Editace kontaktních osob zprostředkovatele Zdroj: Autor	71
29	Okno aplikace — Seznam pohledávek klienta Zdroj: Autor	71
30	Okno aplikace — Editace pohledávky Zdroj: Autor	72
31	Okno aplikace — Editace případů klienta Zdroj: Autor	72
32	Okno aplikace — Správa šablon dopisů Zdroj: Autor	73
33	Okno aplikace — Editace údajů zprostředkovatele Zdroj: Autor	73

Seznam tabulek

1	Vyhodnocení metodik	17
2	Artefakty projektu	17
3	Softwarové požadavky vyvinutého řešení	34
4	Hardwarové požadavky vyvinutého řešení	34

Slovník pojmů a zkratk

Termín	Zkratka	Význam [Zdroj]
Microsoft .NET Framework	.NET	Platforma pro tvorbu aplikací poskytovaná společností Microsoft. [Autor]
Architektura	-	Struktura (počítačového) systému. [Autor]
Computer-aided Software Engineering	CASE	Softwarové inženýrství (vývoj software) podporované počítačem (tzv. CASE nástroji) [Autor]
Databáze	DB	Strukturovaná kolekce dat v počítači. [The New Oxford American Dictionary ²²]
Datový model	-	Model popisující data v dané oblasti zájmu. [Autor]
Data Definition Language	DDL	Jazyk pro definici dat — podmnožina příkazů jazyka SQL [Autor]
Graphical User Interface	GUI	Grafické uživatelské rozhraní pomocí kterého pracuje uživatel s aplikací. [Autor]
Hardware	HW	Počítačové komponenty (technické vybavení počítače). [Autor]
Hypertext transfer protocol	HTTP	Klient/server komunikační protokol. [Autor]
Integrated development environment	IDE	Aplikace sloužící k vývoji software. [Autor]
Individuální aplikační software	IASW	Software vyvinutý na zelené louce podle požadavků objednatele. [Autor]
Jazyk C#	C#	Programovací jazyk vyvinutý společností Microsoft syntaxí podobný jazyku C++. [Autor]
Jazyk SQL	SQL	Structured Query Language — databázový dotazovací jazyk. [Autor]
Jazyk T-SQL	T-SQL	Programovací jazyk společnosti Microsoft pro databázový server Microsoft SQL Server. [Autor]
Man-day	MD	„Člověkodén“ — pracovní doba jednoho člověka za jeden den. [Autor]
Mediace	-	Zprostředkování řešení sporu, například mediace mezi obětí a pachatelem, mediace rodinných a pracovních sporů. [SPJ, o.s., 2008]
Metodika	-	Metodiku lze definovat jako sadu praktik a postupů. [Autor]
Platforma	-	Sada komponent počítače, ale také sada software nainstalovaného na počítači. [Autor]
Probace	-	Souhrnný termín pro služby zaměřené na pomoc, poradenství a kontrolu chování pachatelů trestných činů, kteří svůj trest nebo opatření vykonávají na svobodě. Z latinského termínu „probare“ — zkoušet, ověřovat. [SPJ, o.s., 2008]

Pokračování na další straně...

²²Vydání dostupné v Mac OS X Leopard

Termín	Zkratka	Význam [Zdroj]
Secure Socket Layers	SSL	Protokol pro zabezpečení komunikace přes TCP/IP. [Autor]
Server	-	Počítačový program (nebo také počítač) poskytující určité služby (funkcionalitu) dalším počítačovým programům (počítačům, uživatelům). [Autor]
Software	SW	Počítačový program. [Autor]
Typový aplikační software	TASW	Standardizovaný software pro širší skupinu zákazníků. [Autor]
Transmission control protocol / Internet protocol	TCP/IP	Sada protokolů pro komunikaci po síti. [Autor]
User interface	UI	Uživatelské rozhraní — rozhraní pomocí kterého uživatel interaguje s aplikací. [Autor]
Web-based distributed authoring and versioning	WebDAV	Rozšíření protokolu HTTP umožňující pohodlnou práci se soubory na serveru. [Autor]

A Konceptuální datový model

Tato příloha obsahuje Konceptuální datový model datové základny vyvinuté aplikace. Vzhledem k jeho rozsáhlosti a nárokům na velikost papíru ho nelze umístit přímo do svazku této bakalářské práce, a proto ho lze najít vložený v jejích deskách.

Notace tohoto modelu je popsána v kapitole 2.7.3.

B Fyzický datový model

Tato příloha obsahuje Fyzický datový model datové základny vyvinuté aplikace. Vzhledem k jeho rozsáhlosti a nárokům na velikost papíru ho nelze umístit přímo do svazku této bakalářské práce, a proto ho lze najít vložený v jejích deskách.

Notace tohoto modelu je popsána v kapitole 2.7.4.

C User Interface Flow Diagram

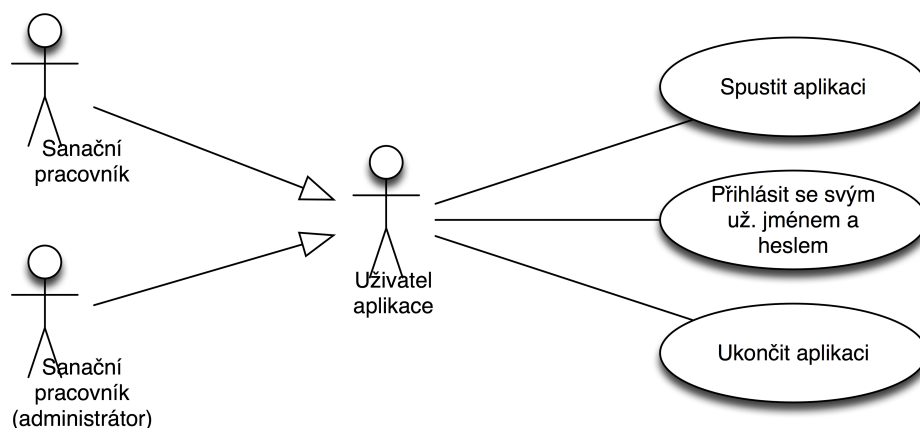
Tato příloha obsahuje User Interface Flow diagram reprezentující možnost pohybu uživatele po vyvinuté aplikaci. Vzhledem k jeho rozsáhlosti a nárokům na velikost papíru ho nelze umístit přímo do svazku této bakalářské práce, a proto ho lze najít vložený v jejích deskách.

Notace tohoto diagramu je popsána v kapitole 2.7.2.

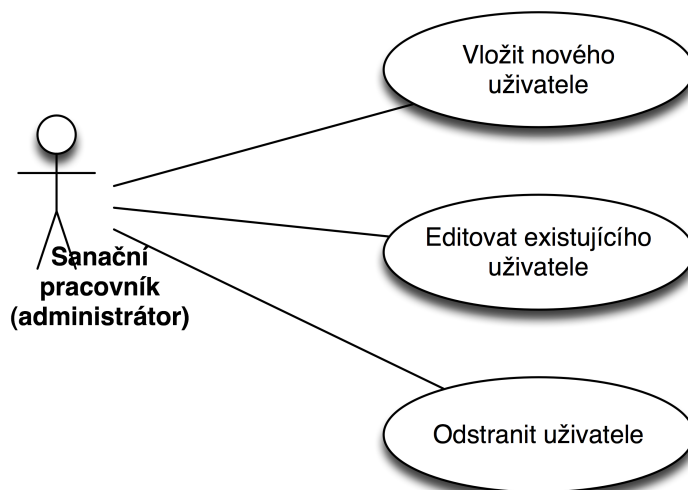
D Use Case diagramy

Tato příloha obsahuje sadu Use Case diagramů vytvořených při návrhu aplikace při a po konzultacích se Zadavatelem.

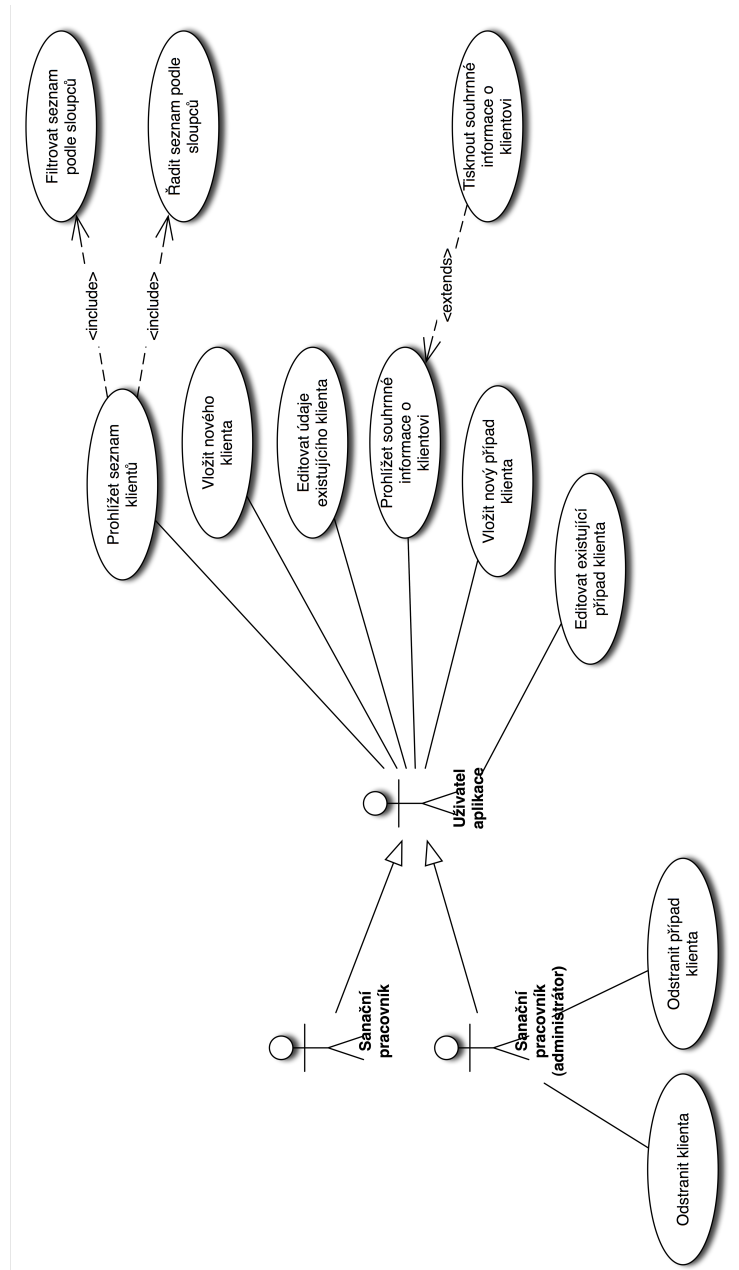
Notace těchto diagramů je popsána v kapitole 2.7.1.



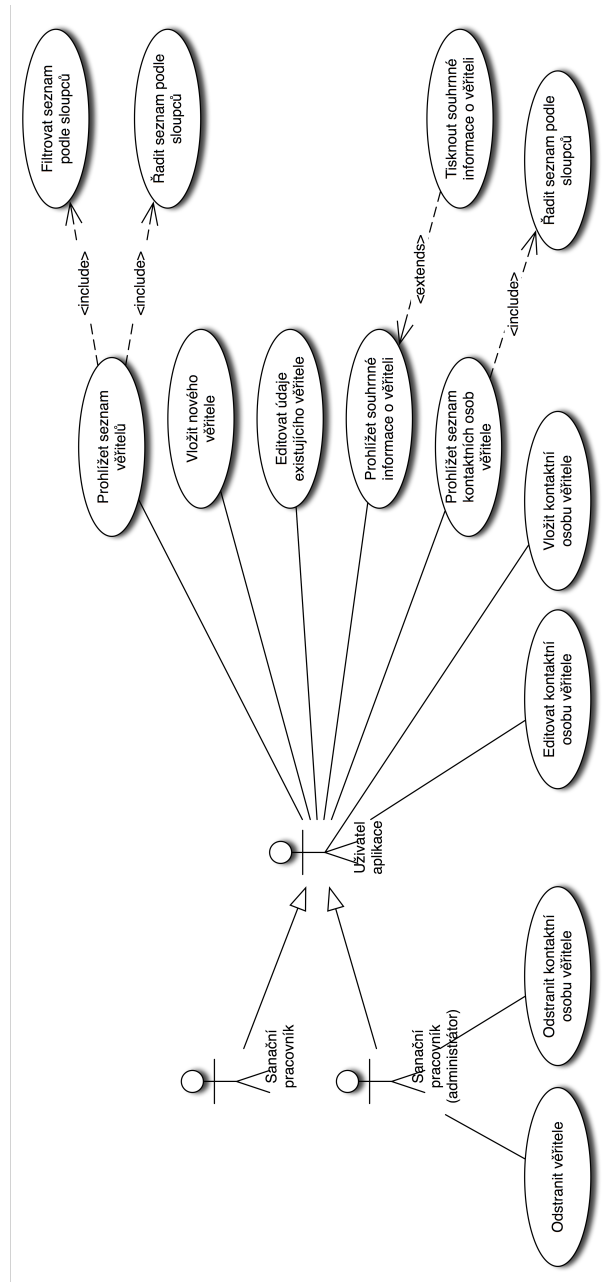
Obrázek 16: Use Case diagram — oblast Obecné činnosti | Zdroj: Autor



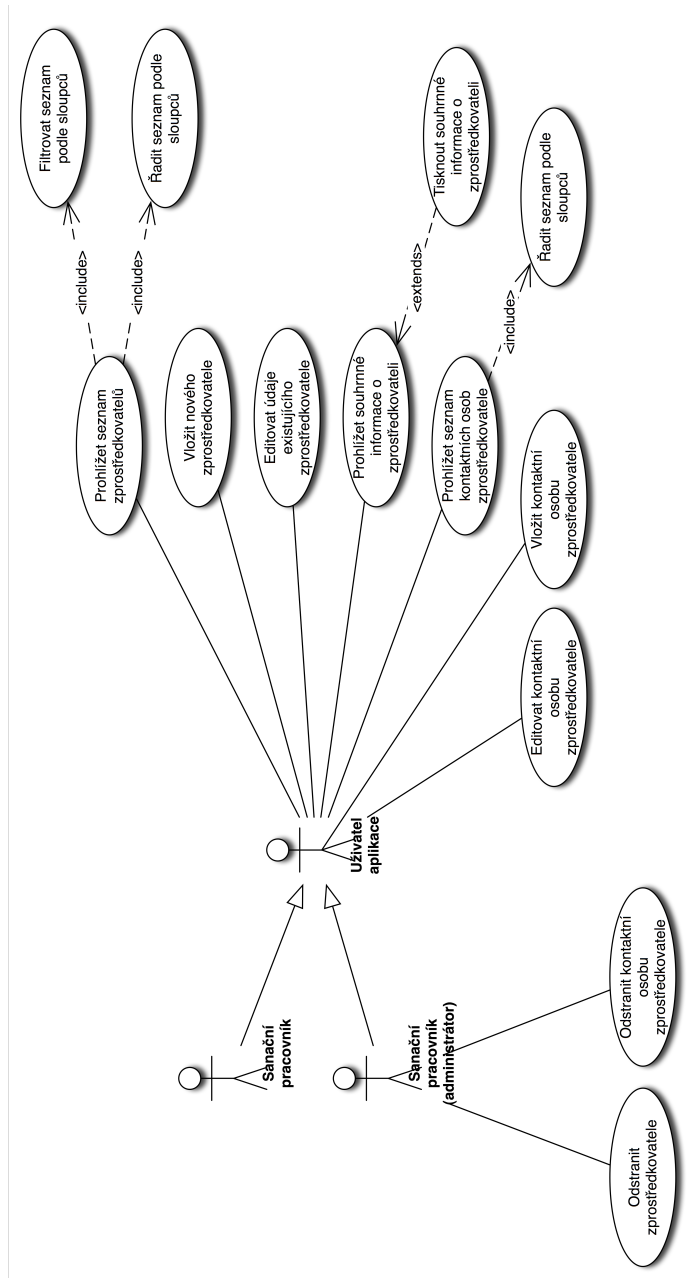
Obrázek 17: Use Case diagram — oblast Uživatelé | Zdroj: Autor



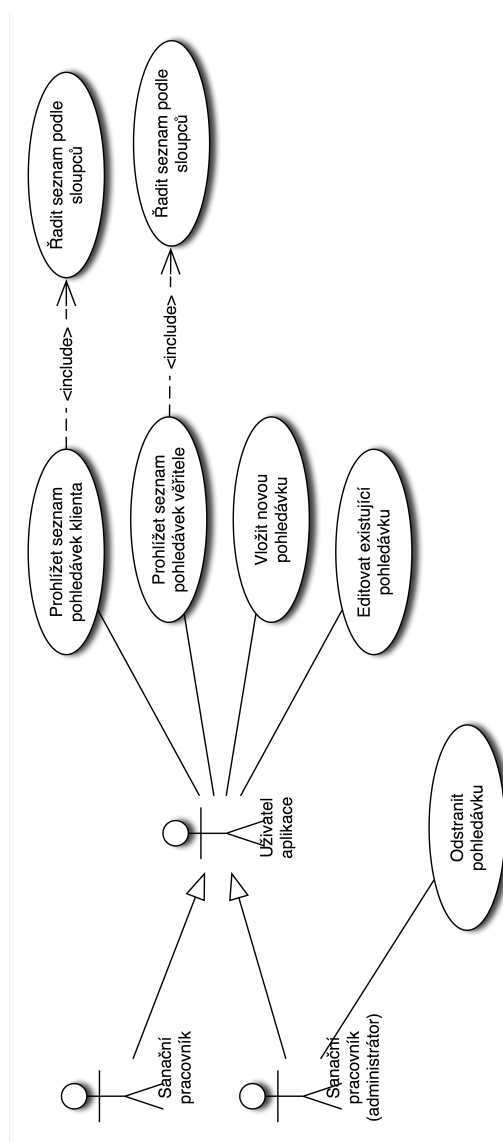
Obrázek 18: Use Case diagram — oblast Klienti | Zdroj: Autor



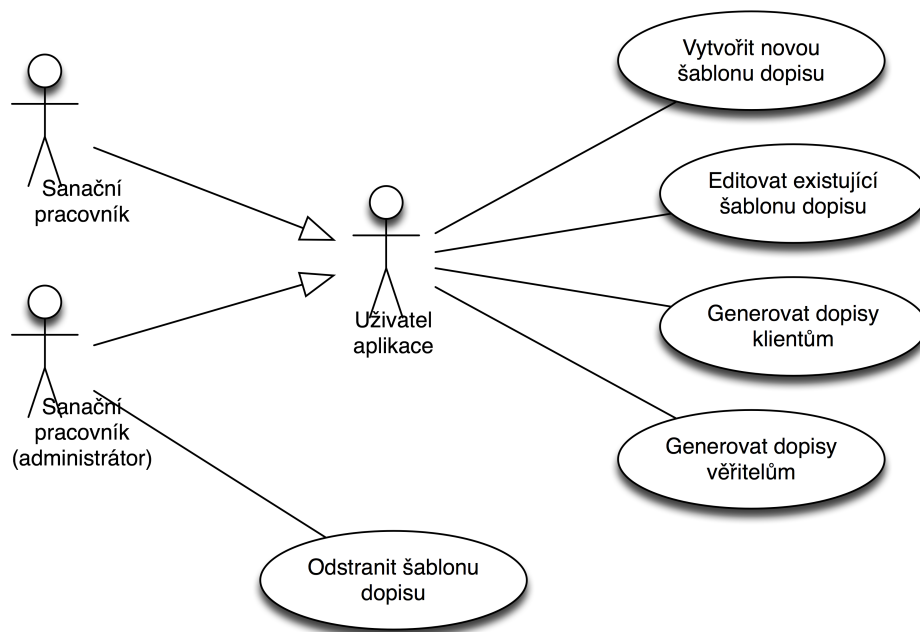
Obrázek 19: Use Case diagram — oblast Věřitelé | Zdroj: Autor



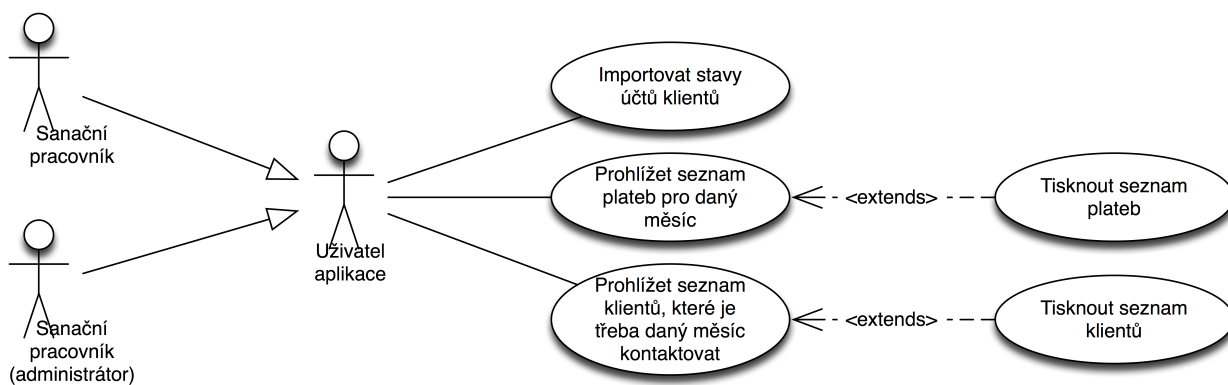
Obrázek 20: Use Case diagram — oblast Zprostředkovatelé | Zdroj: Autor



Obrázek 21: Use Case diagram — oblast Pohledávky | Zdroj: Autor



Obrázek 22: Use Case diagram — oblast Dopisy | Zdroj: Autor



Obrázek 23: Use Case diagram — oblast Účetnictví | Zdroj: Autor

E Protokol o akceptaci

Tato příloha obsahuje ukázkou protokolu akceptačního řízení, specificky přímo protokol prvního akceptačního řízení.

E.1 První akceptační řízení

První akceptační řízení proběhlo 22.9.2008 pro požadavky prvních dvou přírůstkových skupin, tak jak jsou definovány v katalogu požadavků. Výsledkem akceptačního řízení pro danou verzi aplikace bylo její *akceptování s výhradami*.

Chyby nalezené v akceptačním řízení byly zaznamenány a datum opakované akceptace bylo stanoveno na 17.10.2008.

Následující tabulka obsahuje seznam vad a navržených řešení.

	Klasifikace	Popis nedostatku	Akceptované řešení
1	B	Kliknutí na tlačítko detail pohledávky, není-li žádná zvolena, vede k pádu aplikace.	V daném případě zobrazit varovné hlášení.
2	B	Lze vyvolat uložení pohledávky pokud nejsou ve formulářových prvcích kontaktní osoby, věřitelé, nebo zprostředkovatelé.	Po pokusu o uložení při nezvolení kontaktní osoby zobrazit varovné hlášení.
3	B	Do vypočítávaných veličin jsou zařazeny nejen definitivně uznané pohledávky.	Do vypočítávaných veličin v seznamu klientů zařadit pouze definitivně uznané pohledávky.
4	B	Odstranění položky číselníku pokud je zároveň jiná rozepsána vyvolá chybu.	Odstranit chybu v dané situaci.
5	B	Po uložení případu zmizí záložka editace základních údajů.	Odstranit chybu v dané situaci.
6	C	Po uložení dat číselníku není uživatel nijak informován o úspěšnosti.	Po uložení číselníku zobrazit hlášení o případném úspěšném uložení, chybová hlášení jsou již implementována.
7	C	Je-li zadáno málo komplexní heslo uživatele, chybové hlášení není uživatelsky přívětivé.	V dané situaci zobrazit uživatelsky přívětivé hlášení.
8	C	U pole telefon při editaci dat klienta/věřitele/zprostředkovatele není uživatel informován, že číslo nesmí obsahovat mezery.	Zařadit danou informaci do hlášení při nesprávném formátu zadaného řetězce.
9	C	Pole číslo účtu má málo povolených znaků.	Zvýšit akceptovaný počet znaků na 30.
10	C	V detailu pohledávky není provizorní a definitivní dlužná částka formátována správně.	Změnit počet desetinných míst na 2.
11	C	Při přepnutí z jedné záložky adresáře do druhé nejsou aktualizována data v příslušné tabulce s daty.	Odstranit chybu v dané situaci.

Pokračování na další straně...

E PROTOKOL O AKCEPTACI

	Klasifikace	Popis nedostatku	Akceptované řešení
12	C	V seznamu pohledávek formuláře detailu věřitele nefunguje dvojklik na pohledávku.	Odstranit chybu v dané situaci.
13	C	Percentuelní údaje v celé aplikaci nejsou dostatečně přesné.	Percentuelní údaje uvádět s přesností na 2 desetinná místa.
14	C	Umístění tlačítek není v aplikaci jednotné.	Sjednotit veškeré formuláře tak, aby se tlačítka nacházela v panelu nástrojů každého okna.
15	C	Název pole “aktivní případ” v detailu případu klienta není dostatečně jednoznačný.	Pole “aktivní případ” přejmenovat na “aktuální případ”.
16	C	V panelu nástrojů seznamu kontaktních osob věřitele/zprostředkovatele není ikonka smazat.	Přidat ikonku smazat.
17	C	V seznamu věřitelů se nenachází pole ulice.	Přidat dané pole do seznamu.

F Testovací scénář

Následující obrázek je příkladem testovacího scénáře. Přesně je na obrázku uveden testovací scénář pro požadavek P010, jak je popsán v katalogu požadavků.

Test pro požadavek P010

Požadavek:	P010
Název požadavku:	Uživatelské účty

Detailní popis požadavku
<p>Aplikace umožní přidávat, editovat a odstraňovat uživatele. Každý uživatel bude mít své vlastní uživatelské jméno a heslo.</p> <p>Aplikace dále bude obsahovat administrátorský účet s následujícími právy (exkluzivními pro tento účet):</p> <ul style="list-style-type: none"> - veškeré operace s uživatelskými účty - veškeré operace odstranění záznamu

Testovací kritéria		
č.	název	popis
1	Vložení uživatele	Uživatel přihlášený pod administrátorským účtem vloží nového uživatele (jméno, už. jméno a heslo). Nový uživatel se poté do aplikace přihlásí.
2	Editace uživatele	Uživatel přihlášený pod administrátorským účtem změní uživatelské jméno a heslo.
3	Odstranění uživatele	Uživatel přihlášený pod administrátorským účtem odstraní uživatele z bodu TK 1.
4	Omezení práv pro práci s uživatelskými účty	Běžný uživatel (tedy ne administrátor) po přihlášení nevidí v hlavním menu položku "Uživatelské účty".
5	Odstraňování záznamů	Běžný uživatel (tedy ne administrátor) nemá nikde v aplikaci možnost odstranit záznam.

G Katalog požadavků

Tato příloha je katalog požadavků tak jak je popsán v kapitole 3.1.2. Uvedený katalog požadavků je ve verzi k datu odevzdání této bakalářské práce.

KATALOG POŽADAVKŮ							
Požadavky na aplikaci "Databáze dlužníků" pro Sdružení pro probaci a mediaci v justici, o.s.							
Autoři: Jan Kaitoun, Stanislav Komínek Datum: 9.12.2008 Verze: 5.00							
ID	Název	Popis	Přírůstková skupina	Typ	Poslední aktualizace	Zdroj / autor	Stav Vazby
P001	Evidence klientů	<p>Aplikace umožní vkládat, editovat a odstraňovat následující údaje o klientech.</p> <ul style="list-style-type: none">- titul- jméno- příjmení- bydliště klienta<ul style="list-style-type: none">- ulice- město (údaj z číselníku)- psč- kontaktní údaje klienta<ul style="list-style-type: none">- telefon- e-mail- středisko PMS (údaj z číselníku)- kontaktní osoba z probační služby (údaj z číselníku)- odpovědná osoba (uživatel aplikace)- bankovní spojení (číslo bankovního účtu)- poznámka- datum narození <p>Aplikace dále umožní vypsat základní informace o klientech ve formě tabulky. Tuto tabulku umožní aplikace seřadit sestupně i vzestupně podle vybraného sloupce a filtrovat vypsané záznamy podle všech uvedených údajů.</p> <p>Tabulka bude obsahovat následující údaje:</p> <ul style="list-style-type: none">- jméno- příjmení- město- středisko PMS- kontaktní osoba z PMS- odpovědná osoba z SPJ <p>Dodatečné (vypočítávané) údaje ve výpisu (údaje pro výpočet jsou vždy pro aktuální případ klienta):</p> <ul style="list-style-type: none">- celková definitivní dlužná částka- procento definitivní velikosti dluhu (podle následujícího vzorce) <p>Rozklik záznamu v tabulce povede na okno s detailním výpisem údajů o klientovi a jeho případech s možností editace.</p>	1	F	17.10.2008	jednání - Stanislav Komínek	U P008

P005	Evidence zprostředkovatelů	<p>Aplikace umožní vkládat, editovat a odstraňovat věřitelské zprostředkovatele s následujícími údaji:</p> <ul style="list-style-type: none"> - název subjektu - IČO - ulice - město - PSC - telefon - kontaktní osoby (0 až N pro každého zprostředkovatele) <p>Aplikace dále umožní vypsat základní informace o zprostředkovatelích ve formě tabulky. Tuto tabulku umožní aplikace seřadit sestupně i vzestupně podle vybraného sloupce a filtrovat vypsané záznamy podle všech uvedených údajů.</p> <p>Tabulka bude obsahovat následující údaje:</p> <ul style="list-style-type: none"> - název subjektu - IČO - ulice - město - telefon <p>Rozklik záznamu v tabulce povede na okno s detailním výpisem údajů o věřiteli a možnosti editace těchto údajů.</p> <p>Aplikace umožní pro každého zprostředkovatele vkládat, editovat a odstraňovat kontaktní osoby s následujícími údaji:</p> <ul style="list-style-type: none"> - titul - jméno - příjmení - telefon - e-mail <p>Tyto údaje umožní aplikaci vypsat, editovat i odstraňovat v tabulce v detailním pohledu na daného zprostředkovatele.</p> <p>V aplikaci bude vedena historie komunikace s klienty a věřiteli. Jednotlivé ručně vložené záznamy bude možné editovat a odstraňovat.</p> <p>Vedené údaje pro ručně vkládané záznamy:</p> <ul style="list-style-type: none"> - typ události (<i>údaj z číselníku</i>) - datum a čas - poznámka <p>Automaticky vkládané záznamy:</p> <ul style="list-style-type: none"> - aktualizace stavu účtu klienta - vygenerování opisu klientovi nebo věřiteli - změna definitivní dlužné částky pohledávky - změna stavu případu klienta <p>Vazba na P008 - mezi číselníky bude v rámci 4. přírůstkové skupiny přidán číselník druhů komunikace s klientem / věřitelem (<i>editovatelny</i>)</p>	2	F	17.10.2008	jednání - Stanislav Komínek	U	
P006	Evidence kontaktních osob zprostředkovatelů		2	F	17.10.2008	jednání - Stanislav Komínek	U	
P007	Evidence komunikace s klienty a věřitelem		4	F	18.6.2008	jednání - Stanislav Komínek	R	P001 P002 P003 P008

P008	Číselníky	V aplikaci budou vedeny následující číselníky: - číselník středisek Probační a mediální služby (<i>editovatelný</i>) - číselník kontaktních osob Probační a mediální služby (<i>editovatelný</i>) - číselník měst ČR (<i>editovatelný</i>) - číselník stavů případu dlužníka (<i>needitovatelný, s následujícími položkami</i>) <ul style="list-style-type: none">- aktivní (plati)- aktivní (neplati)- aktivní (pozastaveno)- poskytnutá půjčka (plati)- poskytnutá půjčka (plati se zpožděním)- poskytnutá půjčka (neplati)- poskytnutá půjčka (pozastaveno)- ukončeno úspěšně- ukončeno neúspěšně - číselník druhů komunikace s klientem / věřitelem (<i>editovatelný</i>) - číselník druhů pohledávek (<i>editovatelný</i>) - číselník stavů pohledávek (<i>needitovatelný, s následujícími položkami</i>) <ul style="list-style-type: none">- dočasně uznaná- dočasně neuzezaná- definitivně uznaná- definitivně neuzezaná- vyřízená- nerespektovaná	1	F	17.10.2008	Jednáni - Stanislav Komínek	U	
P009	Aplikace umožní paralelní práci více uživatelů		V	B	18.6.2008	Jednáni - Stanislav Komínek	U	P010
P010	Uživatelské účty	Aplikace umožní přidávat, editovat a odstraňovat uživatele. Každý uživatel bude mít evidováno své vlastní uživatelské jméno a heslo. Aplikace dále bude obsahovat administrátorský účet s následujícími právy (exkluzivními pro tento účet): <ul style="list-style-type: none">- veškeré operace s uživatelskými účty- veškeré operace odstranění záznamu	1	B	18.6.2008	Jednáni - Stanislav Komínek	U	
P011	Dopisová komunikace s klienty / věřiteli	Aplikace umožní vytváření šablon dopisů, kterých bude užíváno k generování dopisů pro klienty a věřitele. Šablona dopisu bude editovatelná v editoru MS Word 2007. Do šablony bude uživateli umožněno vložit text a některá z automatických polí, na jejichž místo budou v procesu generování dopisu doplněny informace vztahující se k danému klientovi / věřiteli. Dopisy budou rozděleny do dvou kategorií (dopisy klientům, dopisy věřitelům), kdy každá z nich bude mít nadefinována automatická pole, tak jak jsou popsána v dokumentu <i>Automatická pole pro generování dopisů</i> . Vygenerované dopisy budou zobrazeny v aplikaci MS Word 2007.	3	F	9.12.2008	Jednáni - Stanislav Komínek	U	P001 P002 P003 P004 P005 P006

P012	Evidence pohledávek	<p>Evidence pohledávek zaznamenává vztah mezi klientem a věřitelem kdy jedna pohledávka je vztahem mezi právě jedním klientem a právě jedním věřitelem. Klient může být několika pohledávkami vázán k několika různým věřitelům a stejně tak může být věřitel několika pohledávkami svázán s několika klienty.</p> <p>Applikace umožní v rámci detailního pohledu na údaje věřitele a klienta vkládat, editovat a odstraňovat pohledávky svázané s daným subjektem.</p> <p>Vedené údaje pohledávky:</p> <ul style="list-style-type: none"> - případ klienta - věřitel - referenční číslo - provizorní dlužná částka - definiivní dlužná částka - datum kdy byl věřitelům odeslán dopis s žádostí o informování o stavu a výši pohledávky - druh pohledávky (<i>údaj z číselníku</i>) - stav pohledávky (<i>údaj z číselníku</i>) - periodicta splácení věřitelům - datum první splátky pohledávky věřiteli - kontaktní osoba strany věřitele (buď kontaktní osoba věřitele, nebo jednoho z jeho zprostředkovatelů) - variabilní symbol - konstantní symbol - specifický symbol - poznámka <p>Výpis pohledávek bude zajištěn tabulkou s níže uvedenými údaji. Tuto tabulku umožní aplikace seřadit sestupně i vzestupně podle vybraného sloupce.</p> <ul style="list-style-type: none"> - stav pohledávky - druh pohledávky - provizorní dlužná částka - definiivní dlužná částka - referenční číslo - datum 1. splátky věřiteli - věřitel / jméno a příjmení klienta (v závislosti na pohledu - klient / věřitel) 	2	F	17.10.2008	Jednáni - Stanislav Komínek	ČR	P001 P002 P003 P004 P005 P006 P008
P013	Výpočet doporučené splátky		4	F	18.6.2008	Jednáni - Stanislav Komínek	Z	
P014	Výpočet statistik		4	F	18.6.2008	Jednáni - Stanislav Komínek	Z	
P015	Všechny akce odstraňující záznam musí být potvrzeny	Každé akci odstraňující záznam musí předcházet dialogové okno s požadavkem o potvrzení.	V	F	18.6.2008	Jednáni - Stanislav Komínek	ČR	
P016	Částky placené věřitelům	Všechny finanční částky směřující k věřiteli budou zaokrouhleny na koruny dolů.	V	F	9.12.2008	Jednáni - Stanislav Komínek	U	
P017	zaozkrouhlovány na koruny dolů Soupis majetku klienta		4	F	18.6.2008	Jednáni - Stanislav Komínek	Z	

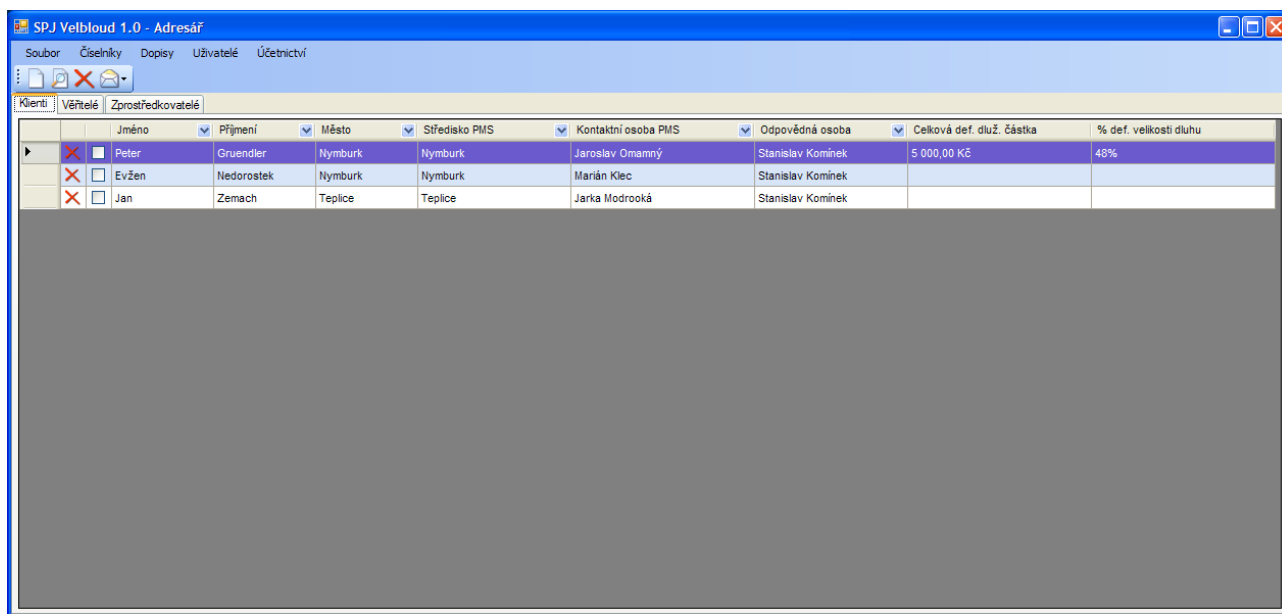
P018	Generování seznamu plateb věřitelům	<p>Aplikace umožní vygenerovat seznam plateb věřitelům za předcházející období.</p> <p>Klient je do tohoto seznamu zařazen, je-li jeho aktuální případ ve zkušební době, jeho stav je aktivní (nepozastavený) a stav jeho účtu je vyšší nebo roven částce, kterou platí do sanačního fondu.</p> <p>Údaje zobrazené v seznamu:</p> <ul style="list-style-type: none"> - účetní označení případu klienta - jméno klienta - název věřitele - referenční číslo pohledávky - variabilní symbol - konstantní symbol - specifický symbol <p>Tento seznam dále kromě zobrazení umožní aplikaci vytisknout.</p>	3	F	9.12.2008	jednání - Stanislav Komínek	U	P001 P002
P019	Generování seznamu klientů, které je třeba kontaktovat	<p>Aplikace umožní pro předcházející období vygenerovat seznam klientů, které je třeba kontaktovat.</p> <p>Klient je do tohoto seznamu zařazen, je-li jeho aktuální případ ve zkušební době, jeho stav je aktivní (nepozastavený) a je-li stav jeho účtu menší, než částka kterou platí do sanačního fondu.</p> <p>Údaje zobrazené v seznamu:</p> <ul style="list-style-type: none"> - účetní označení případu klienta - jméno klienta - kontaktní osoba z PMS - středisko PMS 	3	F	9.12.2008	jednání - Stanislav Komínek	U	P001 P002
P020	Aktualizace stavů účtů klientů importem výstupu z účetnictví	Aplikace umožní aktualizovat stavy účtů klientů pomocí výstupu z účetnictví ve formátu MS Excel 2003.	3	F	9.12.2008	jednání - Stanislav Komínek	U	
P021	Operační systém	Aplikaci bude možné provozovat pod operačním systémem MS Windows XP.	V	N	18.6.2008	jednání - Stanislav Komínek	R	
P022	Informace evidované o klientovi - rozšíření	Aplikace umožní vkládat, editovat a odstraňovat údaje o klientech vycházející z dokumentu <i>Vstupní formulář k Sanaci dluhů</i> .	4	F	30.9.2008	jednání - Stanislav Komínek	R	P001
P023	Komunikace po Internetu	Aplikace umožní komunikovat s databázovým serverem po Internetu.	V	N	9.12.2008	jednání - Stanislav Komínek	U	

H Ukázky obrazovek aplikace

Tato příloha obsahuje snímky vybraných obrazovek aplikace. Snímky obrazovek jsou přiloženy pro umožnění vytvoření si co nejlepší představy o nejvýznamnějším výstupu této bakalářské práce.

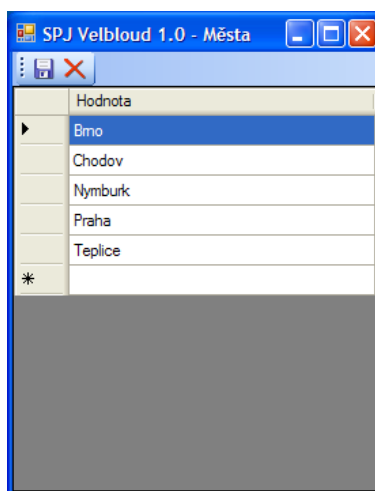


Obrázek 24: Přihlašovací okno aplikace | Zdroj: Autor



Obrázek 25: Hlavní okno aplikace — Adresář | Zdroj: Autor

H UKÁZKY OBRAZOVEK APLIKACE



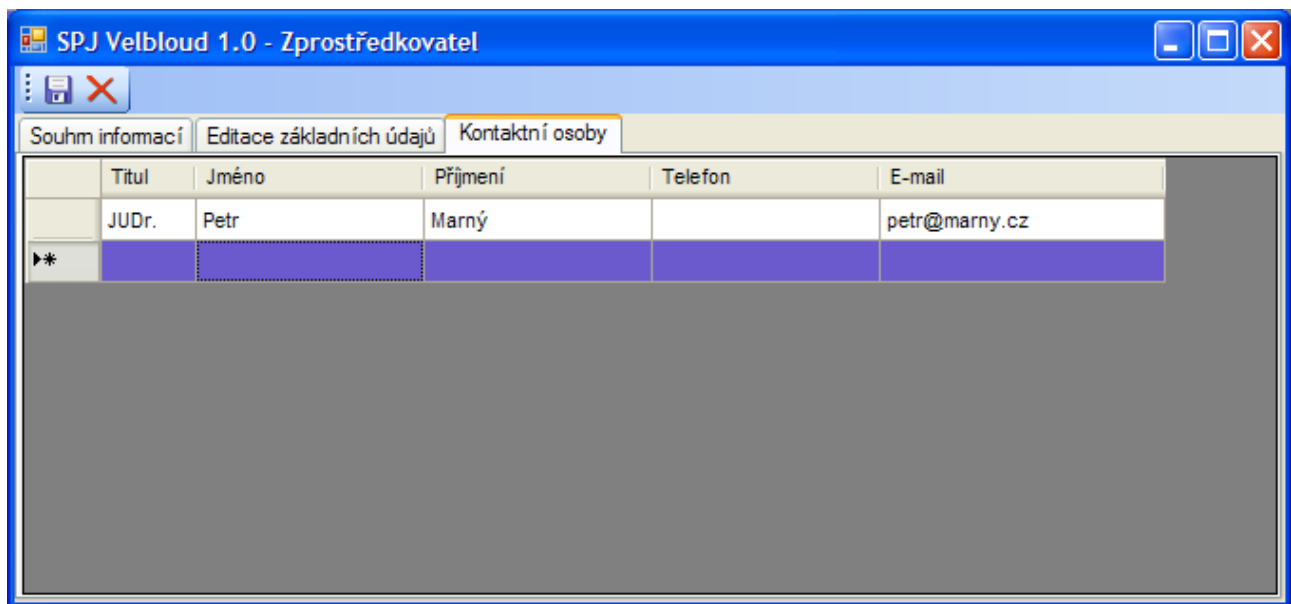
Obrázek 26: Okno aplikace — Editace číselníku | Zdroj: Autor

The screenshot shows a window titled "SPJ Velbloud 1.0 - Klient" with a tabbed interface. The "Souhrn informací" tab is active. The form is organized into several sections:

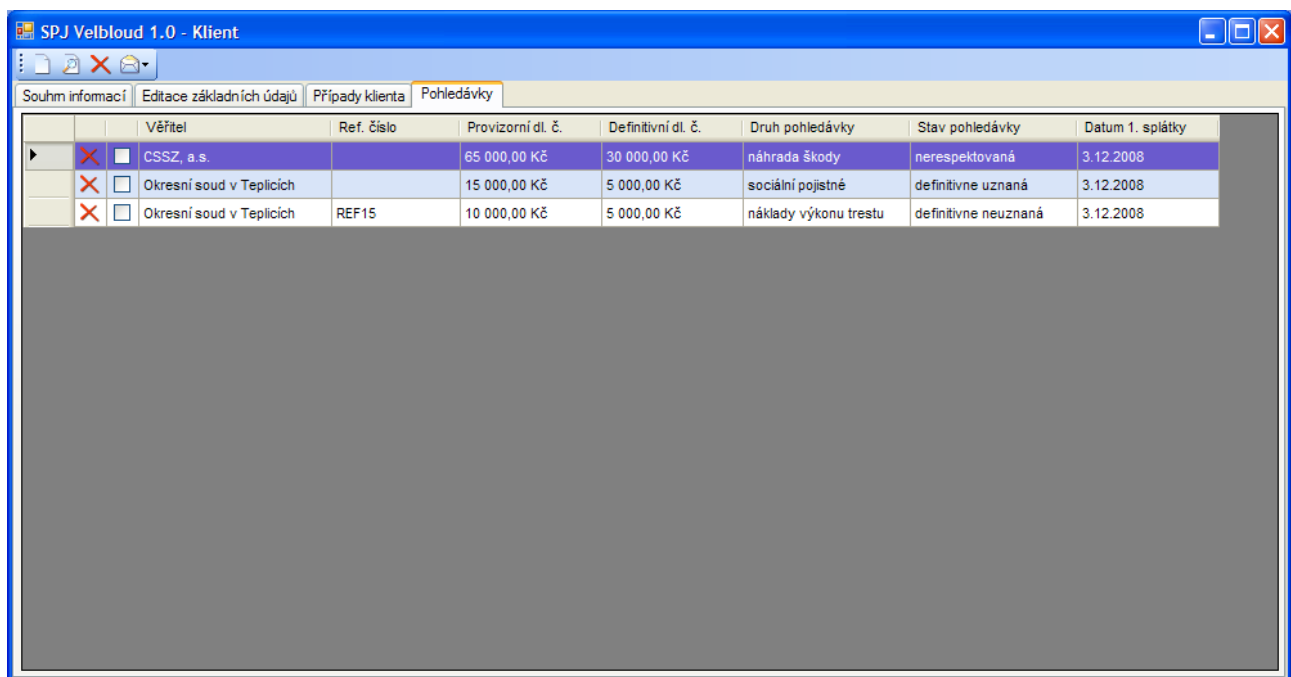
- Základní informace:** Jméno: Peter, Příjmení: Gruendler, Datum narození: 11. prosince 2008, Titul: (empty).
- Odpovědná osoba:** Stanislav Komínek.
- Účetnictví:** Celková splacená částka: (empty).
- Bydliště:** Ulice: Moskevská 334, Město: Nymburk, PSČ: 23521.
- Ostatní kontaktní informace:** Telefon: 60678943, E-mail: peter@gruendler.cz.
- Probační a mediační služba:** Středisko PMS: Nymburk, Kontaktní osoba: Jaroslav Omamný.
- Bankovní spojení:** Číslo účtu: (empty).
- Poznámka:** A large text area for notes.

Obrázek 27: Okno aplikace — Souhrn informací o klientovi | Zdroj: Autor

H UKÁZKY OBRAZOVEK APLIKACE



Obrázek 28: Okno aplikace — Editace kontaktních osob zprostředkovatele | Zdroj: Autor



Obrázek 29: Okno aplikace — Seznam pohledávek klienta | Zdroj: Autor

SPJ Velbloud 1.0 - Pohledávka

Základní informace

Druh pohledávky: náhrada škody
Stav pohledávky: nerespektovaná

Referenční číslo

Finanční informace

Provizorní dlužná částka: 65 000,00
Definitivní dlužná částka: 30 000,00
Periodicita splácní: 3
Datum odeslání dopisu věřiteli: 3.12.2008
Datum první splátky věřiteli: 3.12.2008

Subjekty

Klient: Gruendler
Věřitel: CSSZ, a.s.

Kontaktní osoba

☒ věřitele ☐ zprostředkovatele

Poznámka

Calendar Pop-up: prosinec 2008

po	út	st	čt	pá	so	ne
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Dnes: 11.12.2008

Obrázek 30: Okno aplikace — Editace pohledávky | Zdroj: Autor

SPJ Velbloud 1.0 - Klient

Případy klienta

2.12.2008 17:56

Základní informace

Účetní ID: 435344
Stav případu: aktivní (platí)

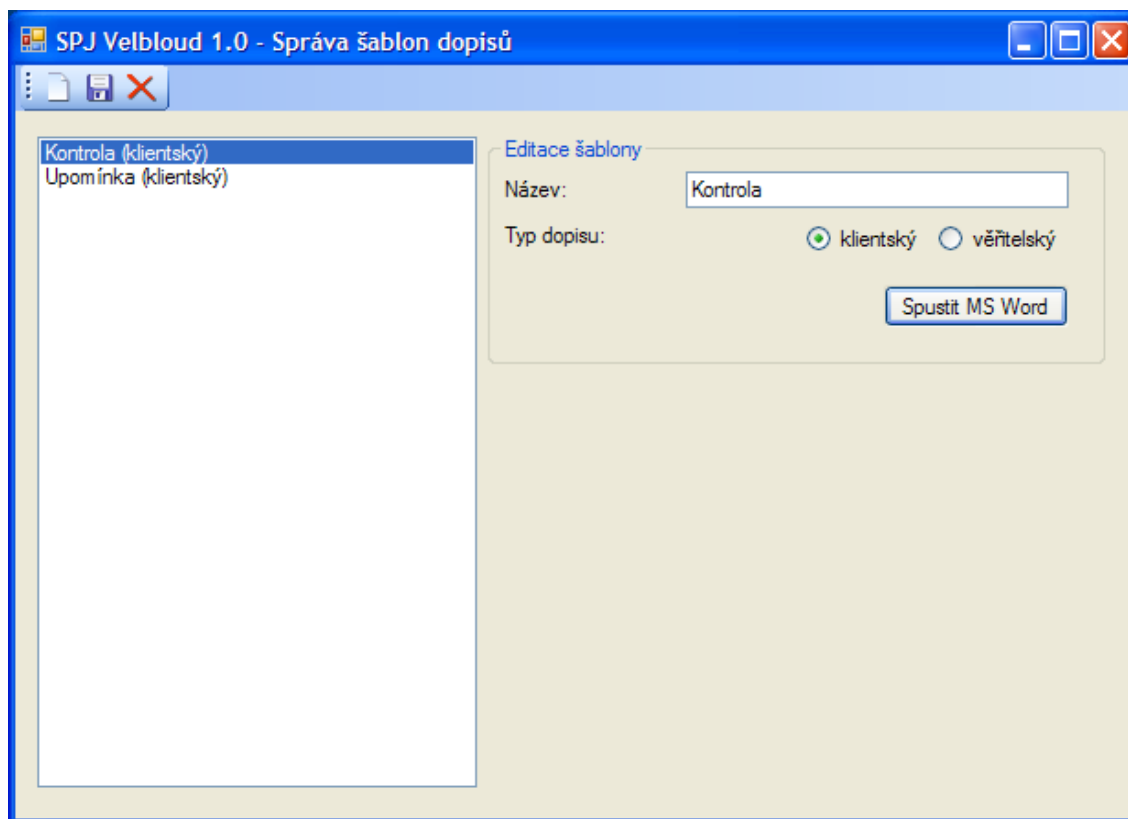
Atributy případu

☒ aktuální případ ☒ případ je ve zkušební době

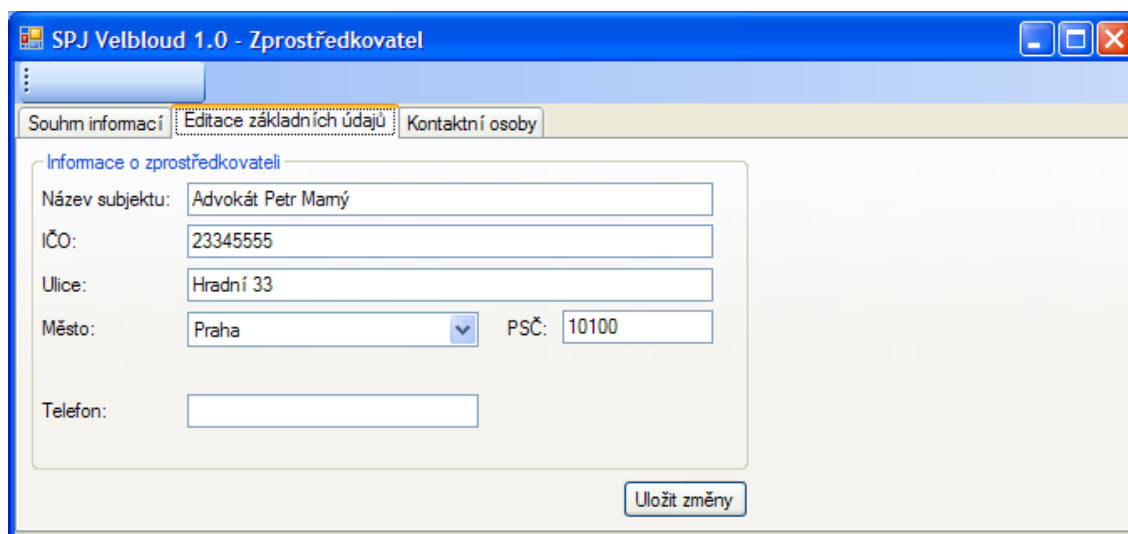
Finanční informace

Měsíční splátka: 1000,00
Počet měsíců splácní: 12
Platí od: 1.9.2008
Stav účtu: 2000,00
Poskytnutá půjčka: 0,00

Obrázek 31: Okno aplikace — Editace případů klienta | Zdroj: Autor



Obrázek 32: Okno aplikace — Správa šablon dopisů | Zdroj: Autor



Obrázek 33: Okno aplikace — Editace údajů zprostředkovatele | Zdroj: Autor