

Informační systém VISitor pro agendu vězněných osob

Information System VISitor For Agenda Of Prisoners

Bc. Michal Plavec

Diplomová práce
2011



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2010/2011

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Michal PLAVEC**

Osobní číslo: **A09717**

Studijní program: **N 3902 Inženýrská informatika**

Studijní obor: **Informační technologie**

Téma práce: **Informační systém VISitor pro agendu vězňených osob**

Zásady pro vypracování:

1. Zpracujte software pomocí vhodného programovacího jazyka.
2. Vytvořte vývojovou dokumentaci pomocí jazyka UML.
3. Vytvořte uživatelskou příručku.
4. Zpracujte příručky pro správce aplikace.
5. Naprogramujte kompletní instalátor aplikace.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. KANISOVÁ, Hana; MÜLLER, Miroslav. UML srozumitelně. 1. vydání. Brno : Computer Press, 2004. 155 s. ISBN 80-251-0231-9.
2. PETROUTSOS, Evangelos. Myslíme v jazyku Visual Basic .NET : 1. díl. První vydání. Praha : Grada Publishing a.s., 2003. 676 s. ISBN 80-247-0371-8.
3. PETROUTSOS, Evangelos. Myslíme v jazyku Visual Basic .NET : 2. díl. První vydání. Praha : Grada Publishing a.s., 2003. 540 s. ISBN 80-247-0372-8.
4. VIESCAS, John; CONRAD, Jeff. Mistrovství v Microsoft Office Access 2007. 1. vydání. Brno : Computer Press, 2008. 1224 s. ISBN 978-80-251-2162-7.
5. DVOŘÁK, Miloš. Objektová analýza, návrh a programování [online]. 2003 [cit. 2011]. Návrhové vzory (design patterns). Dostupné z WWW: <http://objekty.vse.cz/Objekty/Vzory>.
6. FORD, Sara. 266 tipů a triků pro Microsoft Visual Studio. Brno : Computer Press, 2009. 224 s. ISBN 978-80-251-2554-0.
7. LACKO, Luboslav. SQL - Hotová řešení. Brno : Computer Press, 2003. 296 s. ISBN 80-7226-975-5.
8. HANÁK, Ján. Visual Basic 2010 - Hotové riešenia [online]. Praha : Microsoft, 2010 [cit. 2011]. Dostupné z WWW: http://download.microsoft.com/download/2/7/9/27983464-6245-47C3-A2A9-FC57E05581A4/Visual_Basic_2010_riesenia.pdf.

Vedoucí diplomové práce:

Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

24. února 2011

Termín odevzdání diplomové práce:

18. května 2011

Ve Zlíně dne 24. února 2011

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Tato diplomová práce popisuje vytvořený informační systém VISitor, který je určený pro vedení specifické správní agendy s daty vězňených osob dočasně umístěných v káznících - zařízeních Vězeňské služby České republiky. Hlavním cílem tohoto systému je usnadnit některé administrativní procesy pracovníkům Vězeňské služby České republiky, jejichž každodenní součástí pracovní činností je naplňování zákonných norem ve smyslu realizace práv a nároků vězňených osob, vedení potřebné dokumentace a generování formulářů nebo jiných tiskových sestav.

Klíčová slova: vězeň, informační systém, věznice, uživatel, návštěva

ABSTRACT

This diploma thesis describes the creation of visitor information system, which is designed for administration by keeping the specific dates with the inmates temporarily placed in a hard labor – Prison Service of the Czech Republic. The main objective of this system is to facilitate some administrative processes, staff of the Prison Service of the Czech Republic, part of whose daily task is the implementation of legal standards in terms of the implementation of rights and entitlements of prisoners, keeping the necessary documentation and generate forms and other reports.

Keywords: prisoner, information system, prisons, user, visit

#

PODĚKOVÁNÍ

Děkuji vedoucímu práce panu Ing. Petru Šilhavému, Ph.D., za připomínky k mé práci.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 17.05.2011

.....
podpis diplomanta

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 FUNKCE INFORMAČNÍHO SYSTÉMU	12
1.1 VÝKLAD POJMŮ Z OBLASTI NASAZENÍ APLIKACE.....	12
1.2 HLAVNÍ PRVKY FUNKCIONALITY APLIKACE.....	13
1.2.1 Evidence osob ve věznici	13
1.2.2 Agenda umístění osob ve věznici.....	13
1.2.3 Administrativa plánování návštěv.....	13
1.2.4 Víceuživatelské prostředí	13
1.3 BEZPEČNOST DAT	14
2 PROGRAMÁTORSKÁ ČÁST	16
2.1 POUŽITÁ TECHNOLOGIE.....	16
2.2 PROGRAMOVACÍ JAZYK A VÝVOJOVÉ PROSTŘEDÍ	16
2.3 DATABÁZE	16
2.4 SPRÁVA POŽADAVKŮ	17
2.4.1 Funkční požadavky	17
2.4.2 Nefunkční požadavky.....	18
2.5 MODELOVÁNÍ PROCESŮ.....	19
2.5.1 Model procesu realizace žádosti vězněné osoby o naplánování nové návštěvy.....	19
2.5.2 Model procesu Přihlášení uživatele do systému	20
2.6 PŘÍPADY UŽITÍ.....	21
2.6.1 Základní diagram případu užití aplikace VISitor	21
2.6.2 Diagram případu užití Plánování návštěv	22
2.6.3 Slovní popis případu Naplánování nové návštěvy	23
2.6.4 Diagram případu užití Zadání nové osoby ve vztahu s osobou vězně	26
2.6.5 Slovní popis případu Zadání nové osoby ve vztahu s osobou vězně	27
2.6.6 Diagram případu užití Zadání nového uživatele aplikace.....	28
2.6.7 Slovní popis případu Zadání nového uživatele	29
2.6.8 Diagram případu užití Zrušení naplánované návštěvy	31
2.6.9 Slovní popis případu Zrušení naplánované návštěvy	31
2.7 SEKVENČNÍ DIAGRAMY	32
2.7.1 Sekvenční diagram Naplánování nové návštěvy.....	33
2.7.2 Sekvenční diagram Zrušení naplánované návštěvy	34
2.8 DIAGRAMY TŘÍD.....	34
2.9 DATA DIAGRAMY	41
II PRAKTICKÁ ČÁST	49
3 UŽIVATELSKÁ PŘÍRUČKA PROGRAMU	50
3.1 PŘÍRUČKA PRO SPRÁVCE APLIKACE	50
3.1.1 Instalace a spuštění aplikace	50
3.1.2 Umístění databázového souboru	51
3.1.3 První spuštění aplikace.....	52
3.1.4 Správa zaměstnanců	52

3.1.5	Vytvoření nového zaměstnance	54
3.1.6	Správa uživatelských účtů	56
3.1.7	Vytvoření nového uživatele	57
3.1.8	Typy uživatelů	59
3.1.9	Nastavení vlastností aplikace	60
3.2	ZÁKLADNÍ PŘÍRUČKA PRO UŽIVATELE APLIKACE	62
3.2.1	Přihlášení uživatele	62
3.2.2	Hlavní okno aplikace	64
3.2.3	Nastavení vlastností uživatele	65
3.2.4	Změna uživatelského hesla	65
3.2.5	Formulář pro výběr modulu činností	66
3.2.6	Zobrazení nápovědy k programu	67
3.3	SPRÁVA NÁVŠTĚV	67
3.3.1	Vytvoření nového návštěvního dne	67
3.3.2	Blokace návštěvních dnů	68
3.3.3	Vytvoření nové návštěvy	68
3.3.4	Definice osoby ve vztahu s vězněnou osobou	70
3.3.5	Změna stavu a zrušení naplánované návštěvy	71
3.3.6	Tisk seznamu naplánovaných návštěv	72
3.3.7	Zobrazení naplánovaných návštěv	72
3.4	SPRÁVA VĚZNĚNÝCH OSOB	73
3.4.1	Vložení nové osoby jako vězně	73
3.4.2	Zobrazení informací o vězněné osobě	74
3.5	SPRÁVA UBYTOVÁNÍ VĚZNĚNÝCH OSOB	75
3.5.1	Změna umístění vězně	75
3.5.2	Přidání nového oddělení	77
3.5.3	Předání nové ložnice-cely	77
3.5.4	Změna názvu oddělení, změna názvu místnosti	78
3.5.5	Sestava ubytování oddělení	78
	ZÁVĚR	79
	CONCLUSION	81
	SEZNAM POUŽITÉ LITERATURY	83
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	84
	SEZNAM OBRÁZKŮ	85
	SEZNAM TABULEK	87

ÚVOD

Aplikace VISitor je databázová aplikace navržená pro správu dat a informací o osobách ve výkonu trestu odnětí svobody nebo osob ve výkonu vazby. Tyto data jsou pak využívána pro vedení právní agendy specifické pro vězeňské prostředí, zejména plánování návštěv vězňených osob a evidenci ubytování vězňů v jedné organizační jednotce. V rámci tohoto textu bude čtenář seznámen s návrhem samotné aplikace pomocí sofistikovaných postupů objektového programování. Jedná se zejména o sběr požadavků pomocí specifikace funkčních a nefunkčních požadavků, zachycení firemních procesů pomocí procesních modelů a funkcionality systému bude popsána pomocí Use Case diagramů a sestavením scénářů některých případů užití. Na těchto základech bude čtenáři dále představen model tříd představující sloučení funkcionality a datových toků do jednotlivých objektů a komunikace těchto objektů bude názorně představena pomocí sekvenčních diagramů na několika vybraných funkcionalitách aplikace VISitor.

V poslední teoretické části se seznámíme s návrhem struktury databáze, pro ukládání dat vzniklých činnostmi aplikace, pomocí konverze návrhu tříd pomocí datového modelování. Výsledek tohoto procesu bude vyjádřen na několika diagramech popisujících strukturu jednotlivých databázových tabulek společně s uvedením relací mezi jednotlivými atributy.

V praktické části této diplomové práce bude čtenáři představena příručka správce aplikace a uživatelská příručka, která přiblíží čitateli samotnou funkcionality vypracovaného softwarového produktu.

I. TEORETICKÁ ČÁST

1 FUNKCE INFORMAČNÍHO SYSTÉMU

V této části se seznámíme se základními pojmy z oblasti nasazení aplikace, které usnadní snazší orientaci ve výčtu hlavních prvků funkcionality popisované aplikace VISitor.

1.1 Výklad pojmů z oblasti nasazení aplikace

- *Blokace návštěvního dne* – označení dne konkrétního data, jako dne důvodně vyloučeného k naplánování návštěv vězňů
- *Cela* – uzamykatelná místnost určená pro ubytování vězně
- *Ložnice* – místnost určená k ubytování vězněné osoby s možností uzamknutí
- *Návštěva* – zákonem daný nárok vězněné osoby realizovat v prostorách určených vězni setkání s příbuznými osobami
- *Návštěvní den* – den, na který je možné plánovat jednotlivé návštěvy vězňů
- *Návštěvní místnost* – místnost s definovanou kapacitou pro realizaci naplánovaných návštěv
- *Navštívenka* – tiskopis, který je určen k odeslání těm osobám, jako potvrzení o jejich pozvání na konkrétní návštěvu odsouzeného - vězně ve výkonu trestu odnětí svobody
- *Obviněný* – osoba, která je dočasně zbavená svobody po dobu vyšetřování a umístěná ve věznici
- *Oddělení věznice* – uzavřený prostor věznice s místnostmi pro ubytování vězňů tvořící logický a administrativní celek
- *Oddělení výkonu vazby a trestu* – administrativní celek – skupina zaměstnanců, jejichž pracovní náplní je naplňování účelu výkonu trest odnětí svobody a výkonu vazby
- *Odsouzený* – osoba, která je soudem zbavena osobní svobody na dobu určitou a umístěna do věznice
- *Organizační jednotka* – budova, nebo systém budov, ve které jsou umístěny osoby ve výkonu trestu odnětí svobody, ne výkonu vazby
- *Příbuzná osoba* – osoby s definovaným vztahem s osobou vězně
- *Umístění vězně* – určení na jaké konkrétní cele nebo ložnici, v rámci jakého oddělení, je vězněná osoby umístěna, ubytována
- *Vězeň* – osoba z rozhodnutí soudu umístěná do věznice

- *Věznice* – viz organizační jednotka
- *VIS kód* – kombinace čísel a znaků konstantní délky 6 znaků, tvořící unikátní označení osoby vězně
- *Vyrozumění o návštěvě* – tiskopis pro vyrozumění o naplánované návštěvě obviněného – osoby ve výkonu vazby
- *Zaměstnanec věznice* – civilní pracovník nebo příslušník Vězeňské služby České republiky, jehož pracovní náplní je naplňování účelu výkonu trestu nebo výkonu vazby a zároveň naplňování práv vězněných osob

1.2 Hlavní prvky funkcionality aplikace

1.2.1 Evidence osob ve věznici

- Vedení osobních dat o osobách umístěných ve věznici
- Informace o aktuálním pobytu a fyzickém umístění
- Shromažďování údajů o osobách s definovaným vztahem k vězněné osobě
- Fotografie osoby

1.2.2 Agenda umístění osob ve věznici

- Správa ubytovací kapacity věznice – členění, kapacita
- Generování tiskových výstupů umístění osob ve věznici
- Zobrazení a změna pomocí grafických nástrojů aplikace

1.2.3 Administrativa plánování návštěv

- Vedení jednotlivých návštěvních dnů
- Evidence realizace naplánovaných návštěv
- Kontrola stanovených limitů pro eliminaci chyb v plánování
- Generování tiskopisů
- Generování tiskových sestav se seznamy naplánovaných návštěv na jednotlivé dny

1.2.4 Víceuživatelské prostředí

- Režim aplikace klient – server, sdílení dat v síti provozovatele
- Identifikace uživatelů a chráněný přístup k funkcím aplikace
- Definování uživatelských účtů se specifikací uživatelského oprávnění

- Volitelná povinná autentizace uživatelů při spuštění aplikace

1.3 Bezpečnost dat

Navrhovaný systém bude obsahovat osobní data vězněných osob a jejich příbuzných osob, informace charakteru vězeňského prostředí a dotčených právních norem, základní údaje o zaměstnancích, kteří s informačním systémem pracují, ale také figurují jako osoby v rámci vedené administrativy. Je zřejmé, že tento systém musí minimálně respektovat zásady týkající se ochrany osobních údajů a norem vycházejících z právních norem určených pro prostředí nasazení.

Z tohoto důvodu je nutná existence řešení zabezpečení aplikačních dat před neautorizovaným přístupem a řešení pro řízení uživatelských účtů se zaměřením na eliminaci zneužití dat ze strany uživatelů aplikace. Ve druhém případě se nemusí jednat vždy o úmyslné zneužití nebo poškození datové integrity uživatelem, ale může se jednat i o neúmyslný či neuvědomělý uživatelský zásah.

Pro popisovanou aplikaci VSitor bylo stanoveno toto řešení bezpečnosti dat:

Databázový soubor bude umístěn v samostatném adresáři na místním souborovém serveru, který má zajištěno řízení přístupu do jednotlivých sdílených složek pomocí autentizace uživatelů přihlášených na jednotlivých pracovních stanicích. Tímto způsobem lze definovat skupinu osob s přístupem do této složky, která má z pracovního zařazení oprávnění využívat data zpracovávaná aplikací.

Druhým pilířem bezpečnosti je autorizace uživatelů před samotným spuštěním aplikace VISitor, pomocí povinnosti uživatelů, zadat uživatelské heslo z principu odlišné než je heslo pro přihlášení uživatele do operačního systému Windows.

Dalším bezpečnostním prvkem je rozdělení pravomocí jednotlivých uživatelů do uživatelských rolí tak, aby se eliminovala existence velmi silných uživatelů, kteří by měli oprávnění k samotným datům vězněných osob, ale také by disponovali oprávněním řídit uživatelské účty ostatních uživatelů. Z tohoto důvodu je v popisované aplikaci pouze jedna role uživatele, která může spravovat profily uživatelů, dva profily pro zadávání a editaci dat a jeden typ uživatele disponuje oprávněním jen data pasivně číst bez oprávnění je měnit.

V případě použití aplikace jako aplikace v lokálním režimu – databázový soubor je umístěn přímo na pracovní stanici, odpadají položky síťového zabezpečení, a v případě, že

se předpokládá provoz aplikace na multi-uživatelské stanici, je možné databázový soubor umístit do zvláštního adresáře s řízeným přístupem uživatelů.

2 PROGRAMÁTORSKÁ ČÁST

V této části textu se seznámíme s typem technologie a programovacího jazyka, s architekturou programu popsanou pomocí UML diagramů

2.1 Použitá technologie

Pro vývoj této aplikace byla použita technologie firmy Microsoft aplikační platforma .NET Framework 4, která obsahuje všechny potřebné knihovny a prostředí pro běh aplikací. Z toho vyplývá fakt, že pro běh aplikace je podmíněn přítomností této platformy na cílové stanici, ale není to nijak omezující podmínka, neboť v dnešní době je tato platforma zdarma dostupná všem, a to buď je přímo obsažena v operačním systému Windows, anebo je možné její doinstalování pomocí standardního procesu aktualizací a záplat operačního systému.

2.2 Programovací jazyk a vývojové prostředí

Aplikace VISitor byla implementována pomocí objektového programovacího jazyka Visual Basic .NET, který je nástrojem pro vytváření aplikací pomocí technologie .NET Framework.

Jako vývojové prostředí byl použit nástroj firmy Microsoft – Microsoft Visual Studio 2010, které je dostupné i zdarma ve verzi Express, jenž svým rozsahem postačuje pro tvorbu malých aplikací a od placených verzí se liší jen absencí některých nástrojů programování a nástrojů pro práci v programátorských týmech.

2.3 Databáze

Jako databázové úložiště byl zvolen databázový nástroj firmy Microsoft – relační databáze *Microsoft Access*, jenž je součástí softwarového balíčku *Microsoft Office* ve verzi *Professional*. Důvodem této volby, na rozdíl například od *Microsoft SQL Server*, byla jednak nemožnost nasazení *SQL Serveru* v síti provozovatele, ale také možnost provozovat aplikaci pouze lokálně na jedné klientské stanici, bez využití připojení k lokální síti, ačkoliv prioritním architekturou je architektura klient-server, což znamená sdílení databázového úložiště na síťovém poli a připojování jednotlivých klientů spuštěných na stanicích uživatelů.

2.4 Správa požadavků

Nyní se seznámíme s požadavky na funkcionalitu systému. Tyto požadavky shromažďují a popisují, jakou funkcionalitu by měl vyvíjený systém mít. Samotné požadavky se shromažďují při zahájení prací na projektu, neboť neúspěšné zpracování požadavků vede k neúspěchu celého projektu.

2.4.1 Funkční požadavky

Základním významem funkčních požadavků je specifikovat funkčnost aplikace. To znamená specifikovat takové operace a činnosti, které má navrhovaný systém obsahovat.

Požadavek 001 Evidence návštěv vězňů

Systém eviduje naplánované návštěvy jednotlivých vězňů. Každá návštěva je charakterizována dnem, na který je naplánována, čas zahájení a ukončení, délkou návštěvy, konkrétní pozvané osoby a jejich příchod, samotnou realizaci návštěvy nebo její odvolání či nenaplnění.

Požadavek 002 Organizace návštěv

Systém umožní uživatelům naplánovat návštěvy jen na určené dny. Při plánování systém prověřuje, zda naplánováním návštěvy nedochází k porušení stanovených limitů – kapacita návštěvní místnosti nebo limitu počtu návštěv v rámci definovaného časového intervalu.

Požadavek 003 Evidence vězňů

Pro potřeby správy návštěv bude systém schopen evidovat vězně, aktuálně umístění v organizační jednotce. V rámci této evidence osob bude možné ke každé vězni evidovat základní osobní údaje a údaje o aktuálním pobytu ve výkonu trestu odnětí svobody nebo výkonu vazby.

Požadavek 004 Evidence umístění vězňů

V souvislosti s evidencí vězňů, bude systém schopen spravovat agendu ubytování vězňů vzhledem k organizační jednotce, s přihlédnutím na systém a strukturu jednotlivých oddělení věznice a jim přiřazených cel a ložnic

Požadavek 005 Evidence osob v příbuzenském poměru

Pro plánování návštěv bude systém schopen evidovat další osoby, kterým lze evidovat základní údaje, ale také specifikovat adresu místa bydliště a zejména definovat vztah s osobou vězni. Tyto osoby pak bude možno následně použít jako osoby k pozvání na

<i>návštěvu vězňené osoby.</i>	
Požadavek 006	Evidence zaměstnanců
<i>V rámci správy uživatelů aplikace, specifikace struktury věznice, autorizace naplánovaných návštěv a stanovení osob zodpovědných za průběh návštěvních dnů, bude systém schopen evidovat jednotlivé zaměstnance věznice a jejich pracovní zařazení, společně se základními osobními údaji.</i>	
Požadavek 007	Správa uživatelů
<i>Pro definici skupin uživatelů a uživatelských práv bude systém schopen vytvářet uživatelské profily jednotlivých zaměstnanců a definovat jejich oprávnění, provádět v rámci tohoto systému operace s daty.</i>	

Tabulka 1 - Funkční požadavky

2.4.2 Nefunkční požadavky

Nefunkční požadavky slouží ke specifikaci určitých vlastností systému, nastavují omezující podmínky fungování systému a jsou nezávislé na funkčních požadavcích.

Požadavek 008	Víceuživatelské prostředí
<i>Systém bude schopen využívat více uživatelů. Každý uživatel bude člen definované skupiny uživatelů, která bude mít specifikované oprávnění provádět určité skupiny operací</i>	
Požadavek 009	Řešení kolizí při plánování
<i>Při plánování návštěv bude systém schopen detekovat možné překročení stanovených limitů, a pokud bude taková skutečnost detekována, pak ji oznámí uživateli varováním o této skutečnosti.</i>	
Požadavek 010	Generování tiskových sestav
<i>V souvislosti s agendou ubytování a správou návštěv, bude systém schopen vytvářet tiskové předdefinované sestavy a formuláře, které pak uživateli nabídne k tisku s možností zvolit parametry tisku – počet kopií, cílová tiskárna.</i>	
Požadavek 011	Podepisování funkčních operací
<i>Systém bude schopen ukládat, který uživatel provedl operaci, a tato informace bude pak s touto operací spojena pro pozdější použití</i>	

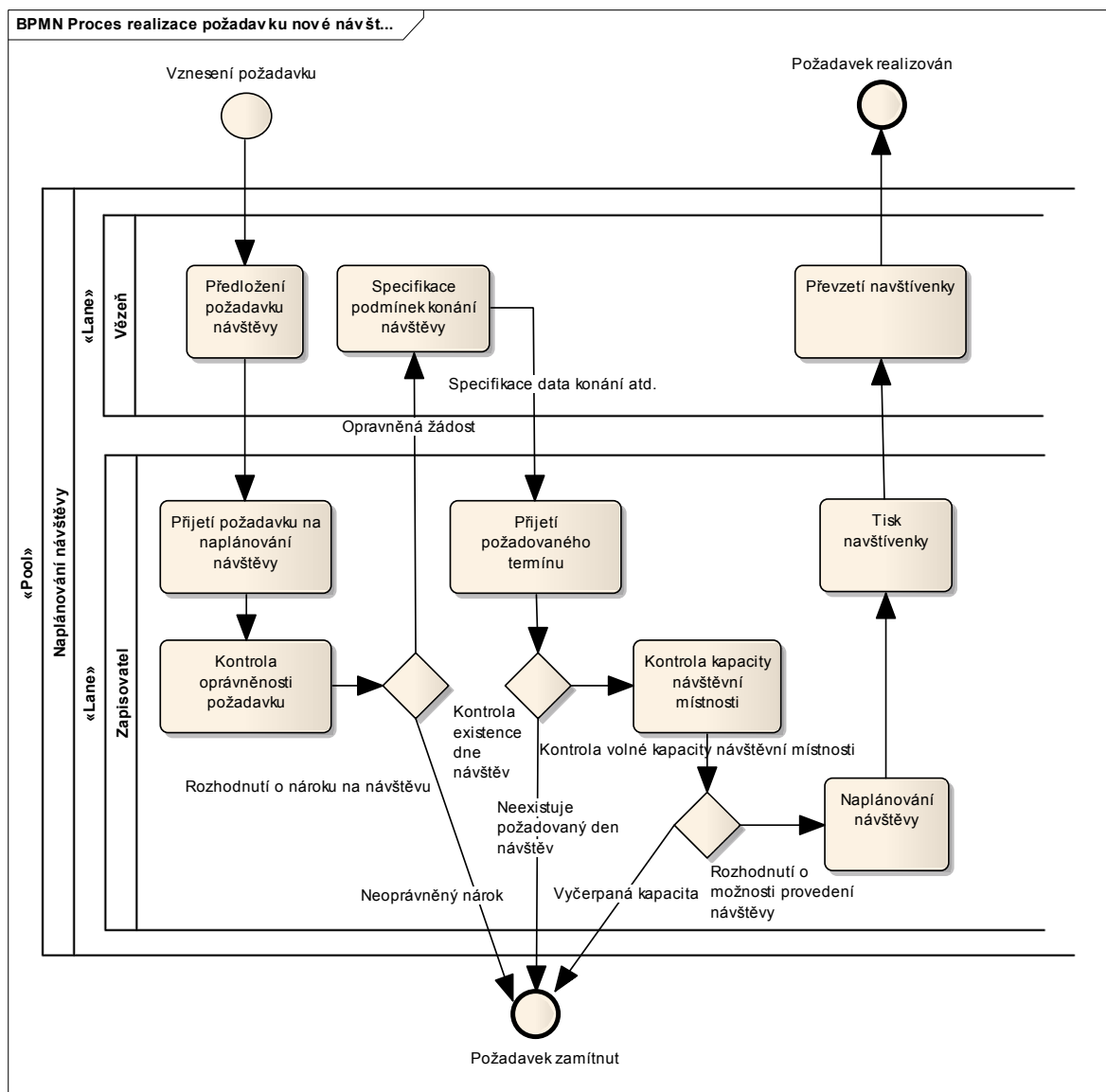
Tabulka 2 – Nefunkční požadavky

2.5 Modelování procesů

Při tvorbě úvodní studie tvorby informačního systému je pro pochopení procesů v dané oblasti nasazení aplikace výhodné v součinnosti se sběrem požadavků využít i diagramy popisující firemní procesy pro zachycení myšlenky jednotlivých operací.

2.5.1 Model procesu realizace žádosti vězněné osoby o naplánování nové návštěvy

Následující diagram popisuje proces realizace podání žádosti vězněné osoby o naplánování nové návštěvy.



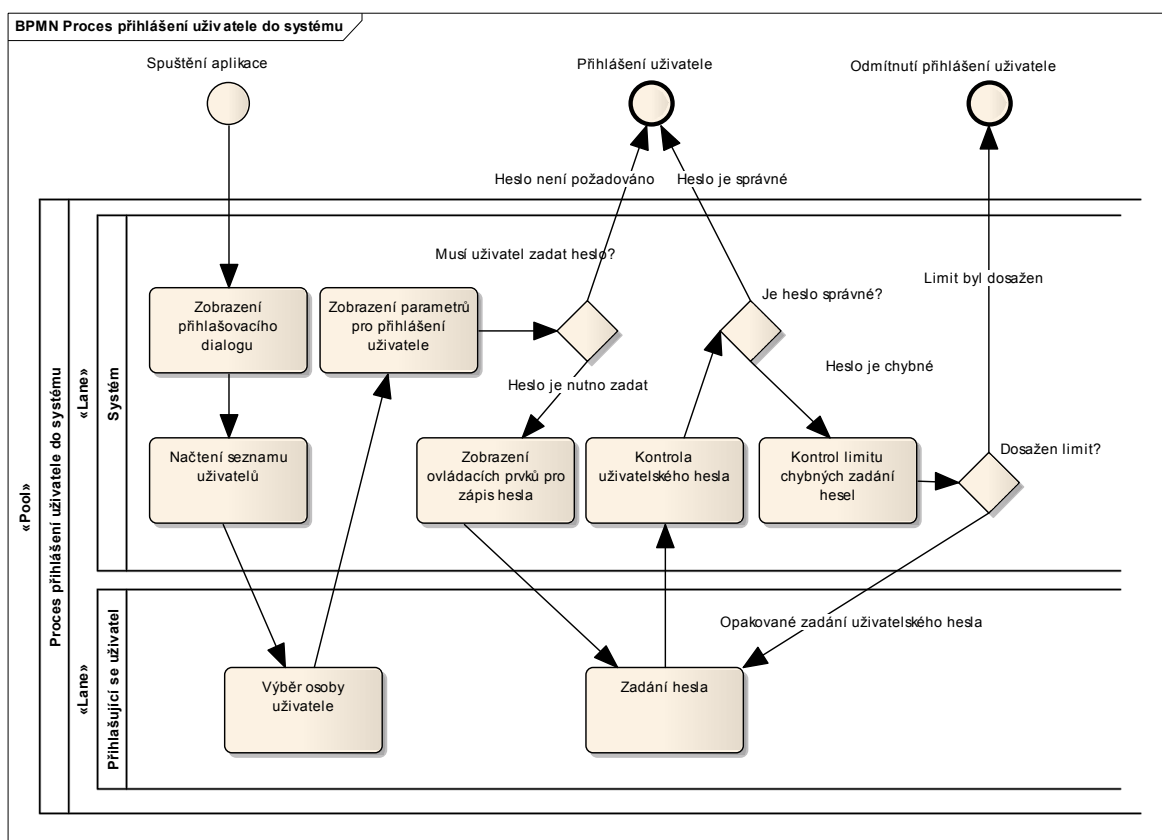
Obrázek 1 – Proces realizace požadavku naplánování návštěvy

Slovní popis procesu realizace požadavku naplánování nové návštěvy:

Vězněná osoba předkládá buď osobně, nebo pomocí předtištěného formuláře žádost o naplánování nárokové návštěvy. Pověřená osoba, zaměstnanec, nejprve posoudí, zda osoba žadatele má v požadovaném termínu nárok na realizaci návštěvy. Pokud ne, pak vyhodnotí žádost jako zamítnutou. Dále zaměstnanec posuzuje, zda existuje možnost realizovat návštěvu v termínu, který je osobou vězně požadován. Pokud není realizace v požadovaném termínu možná, pak je žádost zamítnuta, jinak proces pokračuje kontrolou volné kapacity návštěvní místnosti. Pokud je kapacita návštěvní místnosti v požadovaný termín vyčerpána, pak je žádost opět vyhodnocena jako zamítnuta, jinak dojde k naplánování nové návštěvy a následnému vygenerování tiskopisu navštívenky, která je pak signována oprávněným zaměstnancem (otiskem razítka) a následně je pak předána žadateli. Tím je proces realizace požadavku naplánování nové návštěvy úspěšně dokončen.

2.5.2 Model procesu Přihlášení uživatele do systému

Diagram popisuje proces přihlášení uživatele do systému VISitor. V rámci tohoto procesu se přihlašovaná osoba autentizuje výběrem jména uživatele a zadáním hesla, a následně se přihlašuje jako konkrétní uživatel s definovaným oborem uživatelských práv.



Obrázek 2 – Proces přihlášení uživatele do systému

Popis procesu přihlášení uživatele do systému:

Zaměstnanec spouští aplikaci. Aplikace zobrazí přihlašovací formulář, ve kterém je zobrazen výběrový seznam všech uživatelů a jako aktuálně vybraný uživatel je zvolen uživatel identifikovaný podle uživatele přihlášeného v operačním systému. Zaměstnanec vybere požadovaný profil uživatele. Systém nyní zjišťuje, zda je nutné pro zdárné přihlášení uživatele zadat uživatelské heslo. Pokud ano, pak zobrazí k tomu určené zadávací pole a čeká na zadání hesla, jinak zobrazí zprávu o možnosti přímého přihlášení. V případě, že je nutné zadat heslo, systém čeká na jeho zadání. Přihlašující se osoba zadává heslo a požaduje přihlášení. Systém provede kontrolu správnosti zadaného hesla. Pokud je heslo správné, uživatel je zdárně přihlášen, pokud je chybné, pak provede kontrolu možného počtu opakování zadání hesla, a v případě, že je možno heslo opětovně zadat, pak umožní přihlašující se osobě opětovné zadání hesla. Jinak proces končí neúspěšným přihlášením.

2.6 Případy užití

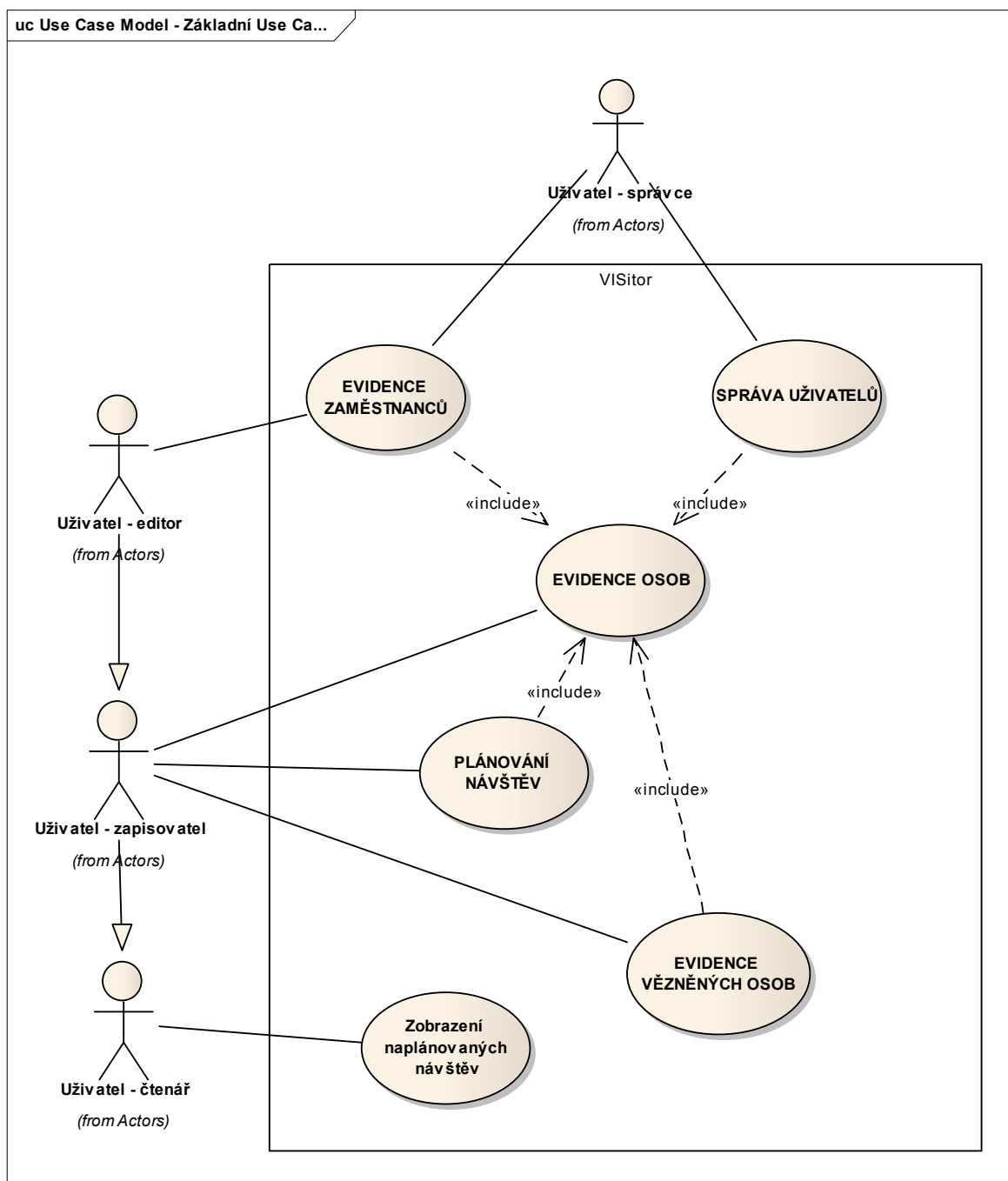
V této části se seznámíme s popisem funkčnosti systému VISitor. Pro tento účel bude použit slovní popis případu užití, což je popis vztahů mezi jednotlivými aktéry – uživateli, a případy užití v budovaném systému. Pro zobrazení bude použito takzvaných Use case diagramů, které zachycují přesnou funkčnost systému a způsob využívání systému.

2.6.1 Základní diagram případu užití aplikace VISitor

Následující diagram popisuje základní vztahy mezi uživateli systému – aktéry, a hlavními případy užití v rámci systému. Z diagramu je zřejmé, že v rámci systému bude vystupovat několik skupin uživatelů, které se budou lišit přidělenými uživatelskými rolemi, a několik skupin případů užití.

Za pozornost jistě stojí vazba mezi aktéry *editor*, *zapisovatel* a *čtenář*. Jedná se o vazbu generalizace neboli zobecnění, což v praxi znamená, že aktéři mají společné chování pro některé případy užití. V tomto případě lze vyvodit, že každý uživatele v roli *zapisovatele* nebo *editora* má oprávnění *Zobrazení naplánovaných návštěv*, ale uživatel v roli *čtenáře* nemá oprávnění jakýmkoliv způsobem zasahovat do *Plánování návštěv*.

A například uživatele v roli *správce* má pouze jako jediný přístup do agendy *Správa uživatelů*, ale nemá oprávnění ani *Zobrazit naplánované návštěvy*.

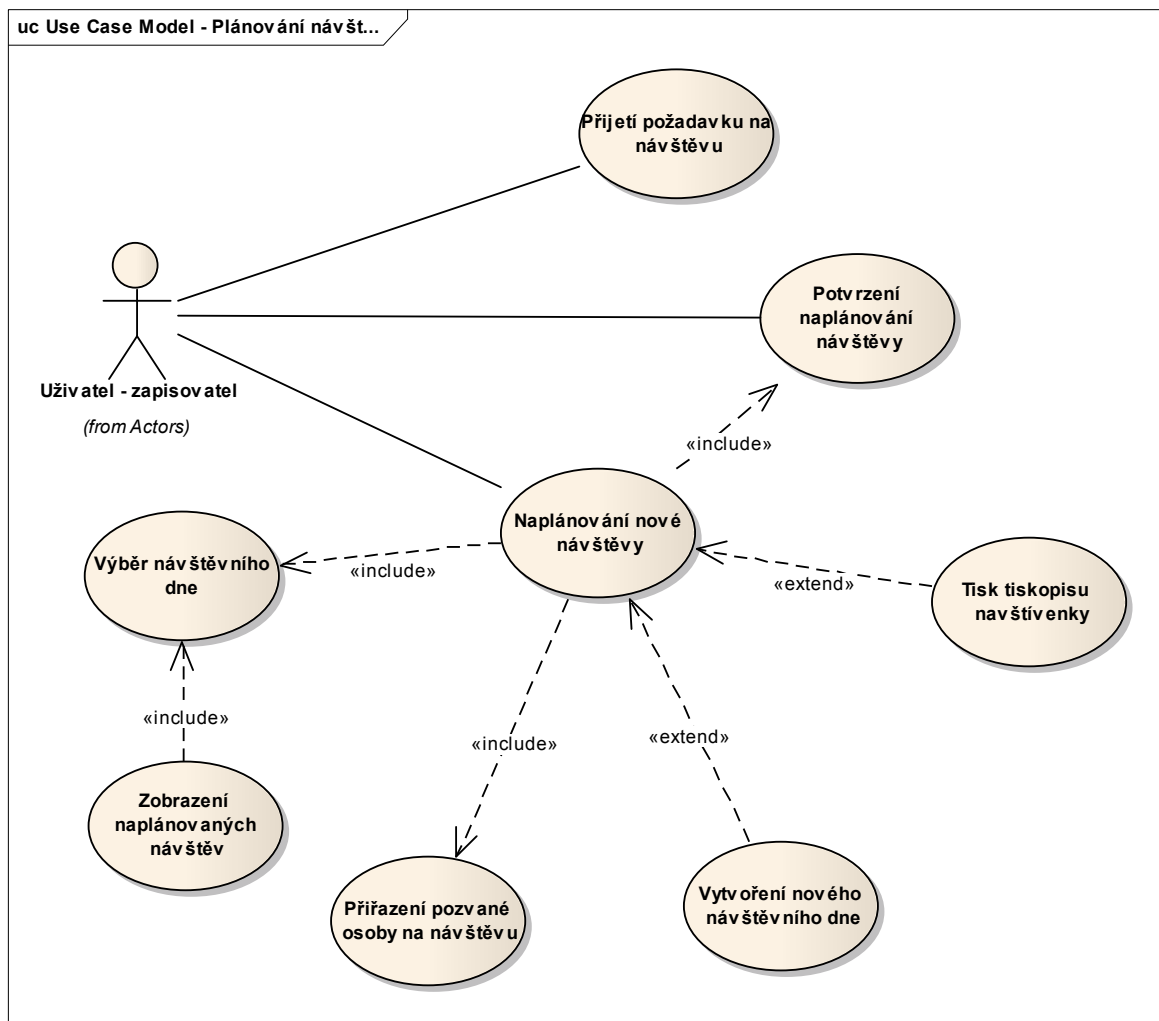


Obrázek 3 – Základní Use case diagram aplikace

2.6.2 Diagram případu užití Plánování návštěv

Nyní se podrobněji zaměříme na případ užití Plánování návštěv. Tento hlavní případ užití se skládá s dalších případů užití, které popisují konkrétněji některé interakce uživatelů se

systemem. Následující diagram popisuje vztah mezi uživatelem v roli zapisovatele a jednotlivými požadovanými funkcemi systému.



Obrázek 4 – Use Case diagram Plánování návštěv

2.6.3 Slovní popis případu Naplánování nové návštěvy

Uživatel s oprávněním vytvářet nové záznamy o návštěvách, zakládá do systému údaje o naplánování nové návštěvy vězněné osoby. Systém zobrazí dialogové okno, pomocí kterého lze zadat požadované údaje a následně vytvořit nový záznam s možností tisku tiskopisu o naplánované návštěvě. Při samotném případě užití je možné popsat hlavní scénář, ale i několik vedlejších scénářů.

Krok	Role	Akce
1	Uživatel	Vybírá v nabídce příkazů odpovídající příkaz pro zobrazení dialogu pro zadání nové návštěvy
2	Systém	Zobrazí dialog pro zadání dat, nastaví ovládací prvky do výchozího stavu a naplní výběrové ovládací prvky položkami –

		seznam budoucích návštěvních dnů, hodnoty času zahájení a konce návštěv a délky návštěvy.
3	Uživatel	Pomocí příkazového tlačítka zobrazí výběrový dialog pro výběr aktuálního vězně, kterému bude plánována návštěva
4	Systém	Zobrazí modální dialog s možností výběru vězněné osoby
5	Uživatel	Označí vhodnou osobu a zadá příkaz Vybrat
6	Systém	Převezme z výběrového dialogu odkaz na vybranou osobu vězně, dialog uzavře a nastaví formulář pro zadání nové návštěvy podle vybrané osoby
7	Uživatel	Vybírá ze seznamu budoucích návštěvních dnů požadovaný den návštěvy. <i>POKUD</i> není požadovaný den nalezen <i>PAK</i> uživatel požaduje vytvoření nového návštěvního dne => Alternativní scénář ALT101 <i>JINAK</i> vybere požadovaný den <i>KONEC POKUD</i>
8	Systém	Nastaví položky formuláře na vybraný návštěvní den a čeká na další uživatelský podnět
9	Uživatel	Vybírá ze seznamu požadovaný čas zahájení začátku návštěvy
10	Systém	Po výběru hodnoty času zahájení, nastavuje jako aktuální prvek pro výběr hodnoty představující délku návštěvy a rozevře jeho roletové menu
11	Uživatel	Vybírá hodnotu délky návštěvy
12	Systém	Dopočítává hodnotu konce návštěvy
13.1.	Uživatel	(Volitelně) zadává příznak, zda je povoleno během návštěvy přijetí nárokového balíčku
13.2.	Uživatel	Vybírá se seznamu konkrétní návštěvní místnost
14	Systém	Provádí výpočet, zda naplánováním této návštěvy nedochází k porušení stanovených limitů kapacity návštěvní místnosti nebo jiný pravidel <i>POKUD</i> je nalezeno porušení pravidel vrací se k bodu 8 <i>PAK</i> zobrazí informační zprávu a nedovolí uložení nové návštěvy <i>KONEC POKUD</i>
15	Systém	<i>POKUD</i> typ vězněné osoby vyžaduje definici osob pozvaných k návštěvě <i>PAK</i> zpřístupní pole pro výběr osob s definovaným vztahem k osobě vězně <i>JINAK</i> přechází na bod 17 <i>KONEC POKUD</i>
16.1.	Uživatel	Vybírá požadované osoby ze seznamu osob s definovaným vztahem k osobě vězně <i>POKUD</i> není v seznamu požadovaná osoba A uživatel požaduje vložení nové osoby s definovaným vztahem <i>PAK</i> pokračuje alternativní scénář ALT102 <i>KONEC POKUD</i>
16.2.	Systém	Kontroluje, zda při výběru nedochází k porušení zásad počtu pozvaných osob (maximum, minimum)

		POKUD došlo k porušení zásad PAK zobrazí upozornění A znepřístupní příkazy pro uložení JINAK zpřístupní příkazy pro uložení KONEC POKUD
17.1.1.	Uživatel	Požaduje uložení nové návštěvy
17.1.2.	Systém	Vytvoří nový záznam o naplánované návštěvě A zobrazí zprávu o výsledku uložení
17.1.3.	Uživatel	Bere zprávu na vědomí a kliká na tlačítko OK
17.2.1.	Uživatel	Požaduje uložení a tisk tiskopisu
17.2.2.	Systém	Uloží záznam o nové návštěvě Vygeneruje tiskopis Zobrazí dialog pro specifikaci tisku
17.2.3.	Uživatel	Volí parametry tisku Volí tisk
17.2.4.	Systém	Provede tisk na vybranou tiskárnu v počtu zvolených kopií
18	Systém	Uzavírá dialog pro zadání nové návštěvy

Tabulka 3 – Hlavní scénář Naplánování nové návštěvy

Alternativní scénář **ALT101** případu užití Naplánování nové návštěvy pro situaci, kdy v seznamu budoucích Návštěvních dnů schází den, který požadujeme. Hlavní scénář pro vytvoření nové návštěvy je přerušen v bodě 7 a uživatel požaduje založení nového Návštěvního dne.

Krok	Role	Akce
1	Systém	Zobrazí dialog pro vytvoření nového návštěvního dne Naplní položky výběrových nabídek ovládacích prvků pro výběr hodnot
Řízení vytvoření nového návštěvního dne obsahuje vlastní scénář		
2	Systém	Čeká, až uživatel pomocí modálního okna vytvoří nový návštěvní den ...
3	Uživatel	Ukládá nový návštěvní den
4	Systém	Uzavírá dialogové okno a testuje výsledek dialogového okna POKUD byl vytvořen nový den návštěv PAK provádí znovunačtení ovládacích prvků pro výběr návštěvního dne KONEC POKUD
Další kroky alternativního scénáře jsou totožné s hlavním scénářem od bodu 8		

Tabulka 4 – Alternativní scénář ALT101 Naplánování nové návštěvy

Dalším alternativním scénářem je scénář **ALT102**, který začíná v okamžiku, kdy uživatel požaduje zadání nové osoby do systému s definovaným vztahem s osobou vězně. Hlavní scénář je tedy přerušen v bodě 16.1. a pokračuje alternativním scénářem ALT102.

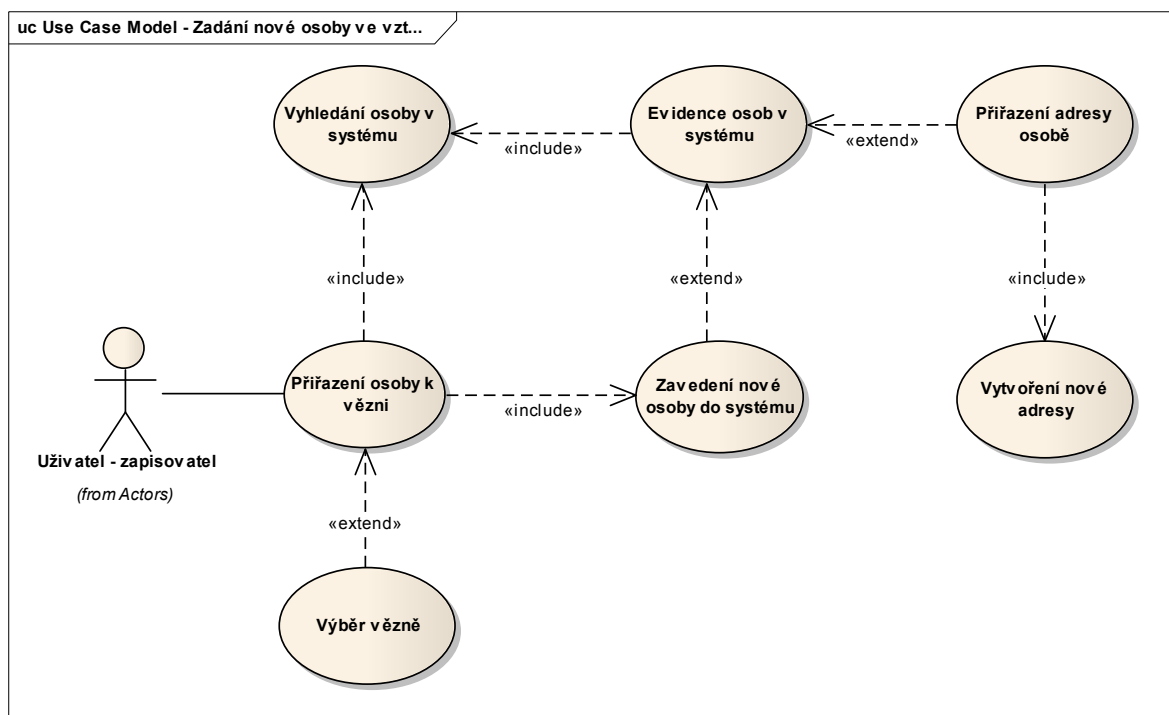
Krok	Role	Akce
------	------	------

1	Systém	Otevírá modální dialogové okno pro vytvoření definovaného vztahu mezi dvěma osobami – osobou vězně a další osobou. Načítá výchozí hodnoty do výběrových seznamů Nastavuje první osobu ve vztahu k osobě vězně
Řízení vytvoření nové osoby v definovaném vztahu s osobou vězně obsahuje vlastní scénář		
2	Systém	Čeká na uzavření dialogového okna...
3	Uživatel	Dokončil zadání nové definice osoby ve vztahu s vězněnou osobou
4	Systém	Uzavírá dialog a testuje návratovou hodnotu POKUD byla vytvořena nová osoba ve vztahu s osobou vězně PAK provede znovu načtení seznamu osob s definovaným vztahem s vězněnou osobou. KONEC POKUD
Další kroky alternativního scénáře jsou totožné s hlavním scénářem od bodu 15		

Tabulka 5 - Alternativní scénář ALT102 Naplánování nové návštěvy

2.6.4 Diagram případu užití Zadání nové osoby ve vztahu s osobou vězně

Dalším případem užití je funkcionality systému, kdy uživatele v roli zapisovatele, nejčastěji v průběhu zápisu nové návštěvy, požaduje zadání nové osoby do systému, která má specifikovaný vztah s osobou vězně. Pro potřeby plánování návštěv je také nutné takové osobě přiřadit i adresu trvalého bydliště.



Obrázek 5 – Use Case diagram případu užití Zadání nové osoby ve vztahu

2.6.5 Slovní popis případu Zadání nové osoby ve vztahu s osobou vězně

Uživatel s oprávněním minimálně zapisovatele, požaduje, aby byl do systému zapsán nový definovaný vztah mezi konkrétní osobou a osobou vězně. Může se například jednat o zápis příbuzenského vztahu mezi osobou vězně a jeho manželkou. Postup jakým lze tohoto docílit popisuje následující hlavní scénář případu užití.

Krok	Role	Akce
1	Uživatel	Příkazem z nabídky požaduje vytvoření nové definice vzájemného vztahu mezi osobou vězně a další osobou
2.1.1.	System	Zobrazí dialogové okno pro definici vztahu dvou osob POKUD je formulář volán s parametrem představujícím osobu vězně, PAK automaticky nastaví jako jednu osobu ve formuláři osobu vězně KONEC POKUD
2.2.1.	Uživatel	Tlačítkem pro výběr osoby může vybrat osobu vězně představující jednu stranu definovaného vztahu
2.2.2.	System	Zobrazí dialog pro výběr vězněné osoby Čeká na uzavření dialogu POKUD návratová hodnota dialogu představuje změnu PAK načte lokální údaje podle nově vybraného vězně KONEC POKUD
3.1.1.	Uživatel	Vybírá druhou osobu v definovaném vztahu s vězněnou osobou
3.1.2.	System	Zobrazí dialog pro výběr osoby pomocí vyhledání v systému uložených osobách
Řízení Vyhledávání osob v systému obsahuje vlastní scénář		
3.2.1.	Uživatel	Požaduje vložení nové osoby do systému
3.2.2.	System	Zobrazí modální dialog pro zavedení nové osoby do systému
Řízení Zadání nové osoby obsahuje vlastní scénáře		
4	System	Přebírá od uzavřeného dialogu parametr druhé osobě do definovaného vztahu s osobou vězně
5	System	POKUD má druhá osoba v systému přiřazenou nějakou adresu PAK ji zobrazí v ovládacím prvku JINAK zobrazí dialog pro zadání nové adresy => alternativní scénář ALT201 KONEC POKUD
6	Uživatel	Požaduje zadání nové adresy vybrané osobě
7	System	zobrazí dialog pro zadání nové adresy => alternativní scénář ALT201
8	Uživatel	Vybírá ze seznamu možností označení vztahu mezi osobou vězně a vybranou osobou
9	Uživatel	Příkazem požaduje uložení definice vztahu
10	System	Prověřuje, zda neexistuje nějaká dřívější definici vzájemného vztahu osob

		<i>POKUD</i> existuje <i>PAK</i> označíme všechny dřívější definice vztahu jako neplatné <i>KONEC POKUD</i>
11	Systém	Ukládá do systému novou definici vztahu osoby vězně a vybrané osoby Uzavírá dialog

Tabulka 6 - Zadání nové osoby ve vztahu s osobou vězně

Alternativní scénář ALT201 tohoto případu užití ošetřují stav, kdy systém zjistí, že vybraná osoba nemá definovanou žádnou adresu nebo uživatel požaduje zadání a následné přiřazení nové adresy vybrané osobě. Alternativní scénář navazuje na bod číslo 5 hlavního scénáře.

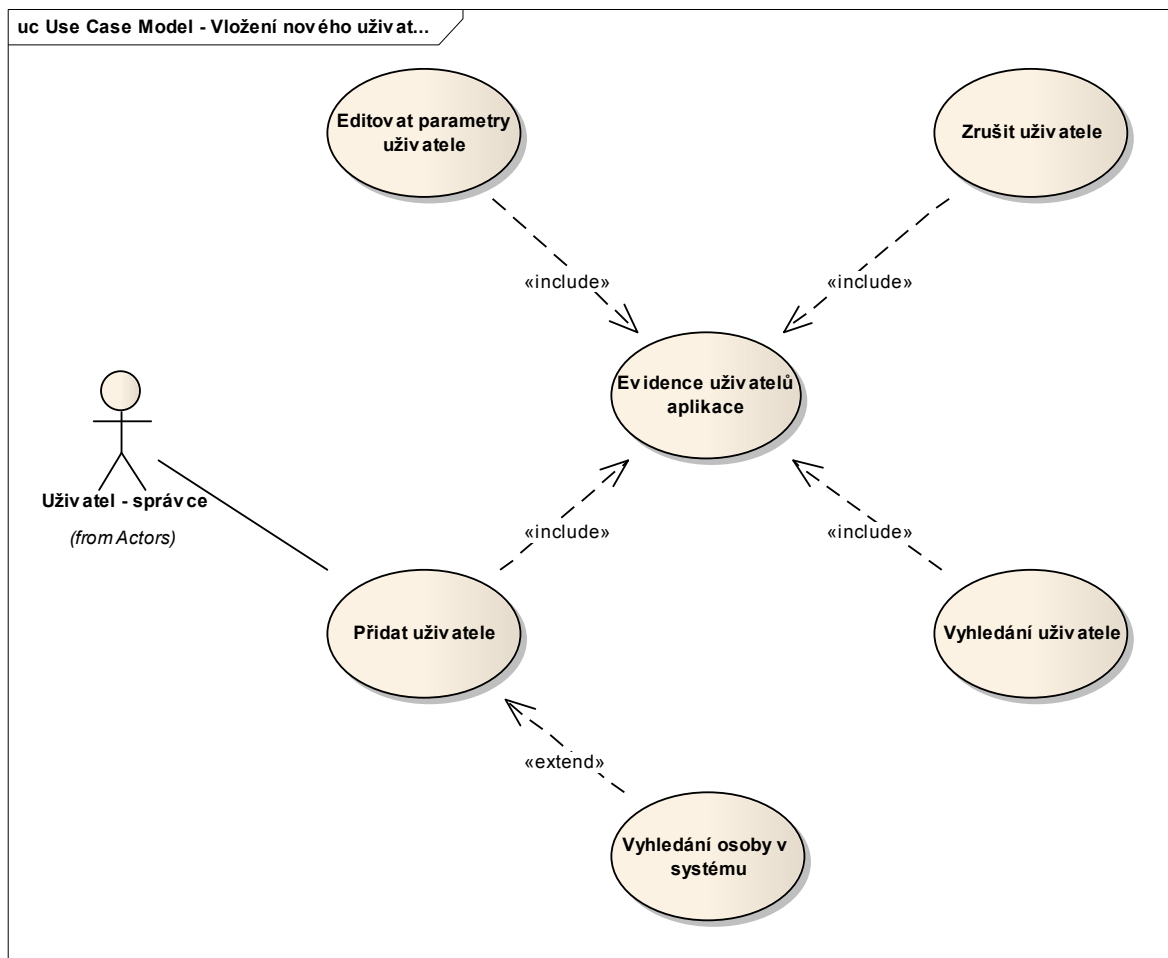
Krok	Role	Akce
1	Systém	Zobrazí dialogové okno pro zadání nové adresy a přiřazení této adresy vybrané osobě. <i>POKUD</i> je v parametru předána osoba, <i>PAK</i> načte informace o osobě, které se bude adresa přiřazovat <i>KONEC POKUD</i>
2	Uživatel	Příkazem z nabídky vybírá možnost zobrazení dialogu pro vytvoření nové adresy
3	Systém	Zobrazí dialog pro vytvoření nové adresy
4	Uživatel	Zadáva potřebné údaje pro vytvoření nové adresy (město, ulice, psč, číslo domu atd.) Ukládá zadané informace tlačítkem Uložit
5	Systém	Provádí kontrolu zadaných dat <i>POKUD</i> je nalezena chyba <i>PAK</i> zobrazí zprávu A návrat do stavu v bodě 4 <i>JINAK</i> ukládá data jako novou adresu <i>KONEC POKUD</i>
6	Systém	Po uzavření modálního dialogu pro vytvoření nové adresy, tuto adresu načte do ovládacího prvku Zobrazí roletové menu ovládacího prvku pro výběr typu adresy (trvalá, přechodné bydliště...)
7	Uživatel	Vybírá typ adresy
8	Uživatel	Ukončuje dialog tlačítkem Uložit
9	Systém	Přiřadí novou adresu vybrané osobě jako adresu vybraného typu
<i>Další kroky alternativního scénáře jsou totožné s hlavním scénářem od bodu 8</i>		

Tabulka 7- Alternativní scénář ALT201 definice osob ve vzájemném vztahu

2.6.6 Diagram případu užití Zadání nového uživatele aplikace

Uživatelé s oprávněním *Správce* mají možnost editovat uživatele systému VISitor. K editačním operacím patří také vytvoření nového uživatele. Uživatelem se může stát jen

osoba, která je evidovaná v systému, a které jsou přiřazeno odpovídající uživatelské oprávnění.



Obrázek 6 – Use Case diagram pro Vytvoření nového uživatele

2.6.7 Slovní popis případu Zadání nového uživatele

Uživatel s odpovídajícím oprávněním požaduje provedení operace vložení nového uživatele aplikace. Pro vytvoření nového uživatele je nutné definovat osobu uživatele a uživatelské oprávnění. K dalším údajům patří zadání hesla uživatele, specifikace povinného zadávání hesla při vstupu do systému a parametry pro identifikaci uživatele pomocí uživatelského jména.

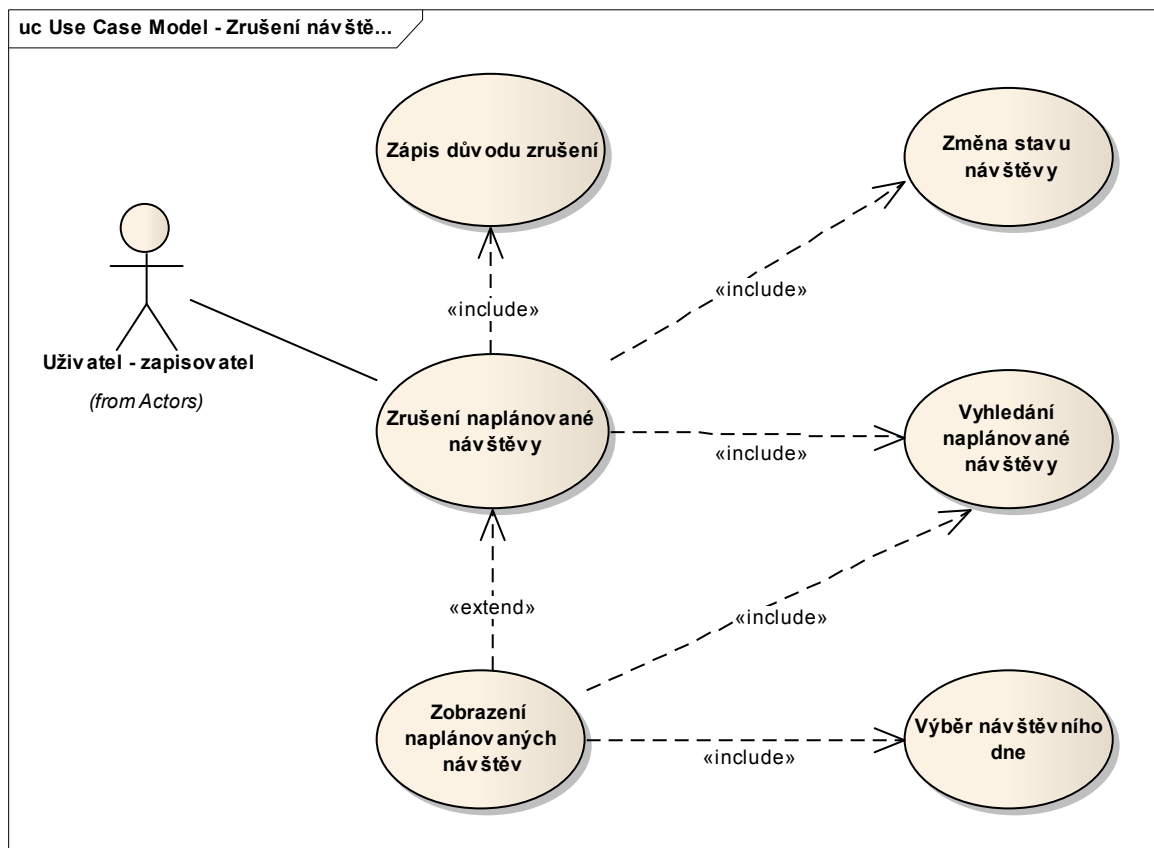
Krok	Role	Akce
1	Uživatel	Příkazem z nabídky požaduje otevření dialogu pro vytvoření nového uživatele
2	Systém	Otevírá dialog pro zadání parametrů nového uživatele Nastavuje všechny ovládací prvky do výchozího stavu
3.1.1.	Uživatel	Pomocí příkazu Vybrat otevírá dialog pro výběr osoby

3.1.2.	System	Zobrazí dialog pro výběr osoby v systému pomocí vyhledání podle zadaných kritérií
3.1.3.	Uživatel	Zadáva kritéria a požaduje vyhledání osob
3.1.4.	System	Vrací výsledek vyhledávání
3.1.5.	Uživatel	POKUD je nalezen požadovaný záznam PAK je označí jako vybraný záznam JINAK se vrací na bod 3.1.3. NEBO případně na bod 2 KONEC POKUD
3.1.6.	System	Uzavírá dialog pro výběr a předává vybranou osobu formuláři pro vytvoření nového uživatele
3.2.1.	Uživatel	Pomocí příkazu Nový požaduje vytvoření nové osoby
3.2.2.	System	Zobrazí modální dialog pro zadání nové osoby do systému
Řízení Zadání nové osoby obsahuje vlastní scénáře		
3.2.3.	System	Předává informace o nové osobě formuláři pro vytvoření nového uživatele
4	System	Zobrazí informace o vybrané osobě nového uživatele
5	Uživatel	Vybírá se seznamu typ uživatelského oprávnění
6	Uživatel	Definuje, zda uživatel musí při vstupu do systému zadat heslo
7	System	POKUD musí uživatele zadat heslo PAK zpřístupní pole pro zadání hesla A pokračuje bodem 8 JINAK znepřístupní pole pro zadání hesla KONEC POKUD
8.1.	Uživatel	Zadáva řetězec uživatelského hesla
8.2.	Uživatel	Zadáva řetězec kontrolního hesla
9	Uživatel	Definuje jméno uživatele v operačním systému Windows
10	Uživatel	Požaduje příkazem Uložit uložení dat nového uživatele
11	System	POKUD validace dat je v pořádku PAK uloží data A vrací zprávu s výsledkem A uzavírá dialog JINAK upozorní na zjištěné nedostatky KONEC POKUD

Tabulka 8 – Hlavní scénář pro Vytvoření nového uživatele

2.6.8 Diagram případu užití Zrušení naplánované návštěvy

Každou návštěvu, která nebyla v plánovaný čas realizována, je možné zrušit a umožnit tak osobě naplánování jiné návštěvy v blízkém náhradním termínu. Zrušení návštěvy patří k editačním operacím ve správě plánování návštěv, tudíž takovou operaci může provést jen osoba s odpovídajícím oprávněním – *Zapisovatel* nebo *Editor*.



Obrázek 7 – Use case diagram pro Zrušení naplánované návštěvy

2.6.9 Slovní popis případu Zrušení naplánované návštěvy

Oprávněný uživatel požaduje zrušení naplánované návštěvy. Pro tuto funkci je nutné nejprve najít v systému konkrétní naplánovanou návštěvu, zadat zdůvodnění zrušení a až poté ji systém nastaví příznak „zrušeno“.

Krok	Role	Akce
1	Uživatel	Požaduje zrušení konkrétní naplánované návštěvy
2	Systém	Zobrazí formulářové okno se seznamem návštěvních dnů se seznamem naplánovaných návštěv na tyto dny
3	Uživatel	Nalezne v seznamu návštěv Návštěvního dne požadovanou návštěvu pomocí výběru osoby vězně Uživatel požadovanou položky vybere (označí)

4.1.1.	Uživatel	Dvojklikem nebo výběrem příkazu Zobrazit podrobnosti v lokálním menu vybrané položky požaduje zobrazení detailu návštěvy
4.1.2.	System	Zobrazí dialogové okno a zobrazí v něm podrobnosti o naplánované návštěvě
4.1.3.	Uživatel	Pomocí příkazu Změnit stav nastavuje nový stav návštěvy na „Zrušena“
4.2.1.	Uživatel	Výběrem příkazu Zrušit návštěvu v lokálním menu vybrané položky požaduje zobrazení zrušení návštěvy
5	System	Zobrazí požadavek na zadání textu zdůvodnění zrušení návštěvy
6	Uživatel	Zadá text odůvodnění zrušení návštěvy
7	System	Ukládá zdůvodnění Nastavuje návštěvě nový stav – „Zrušena“
8	System	Provede znovu načtení seznamu návštěv aktuálního dne
9	Uživatel	Uzavírá dialogové okno nebo pokračuje v editaci návštěv

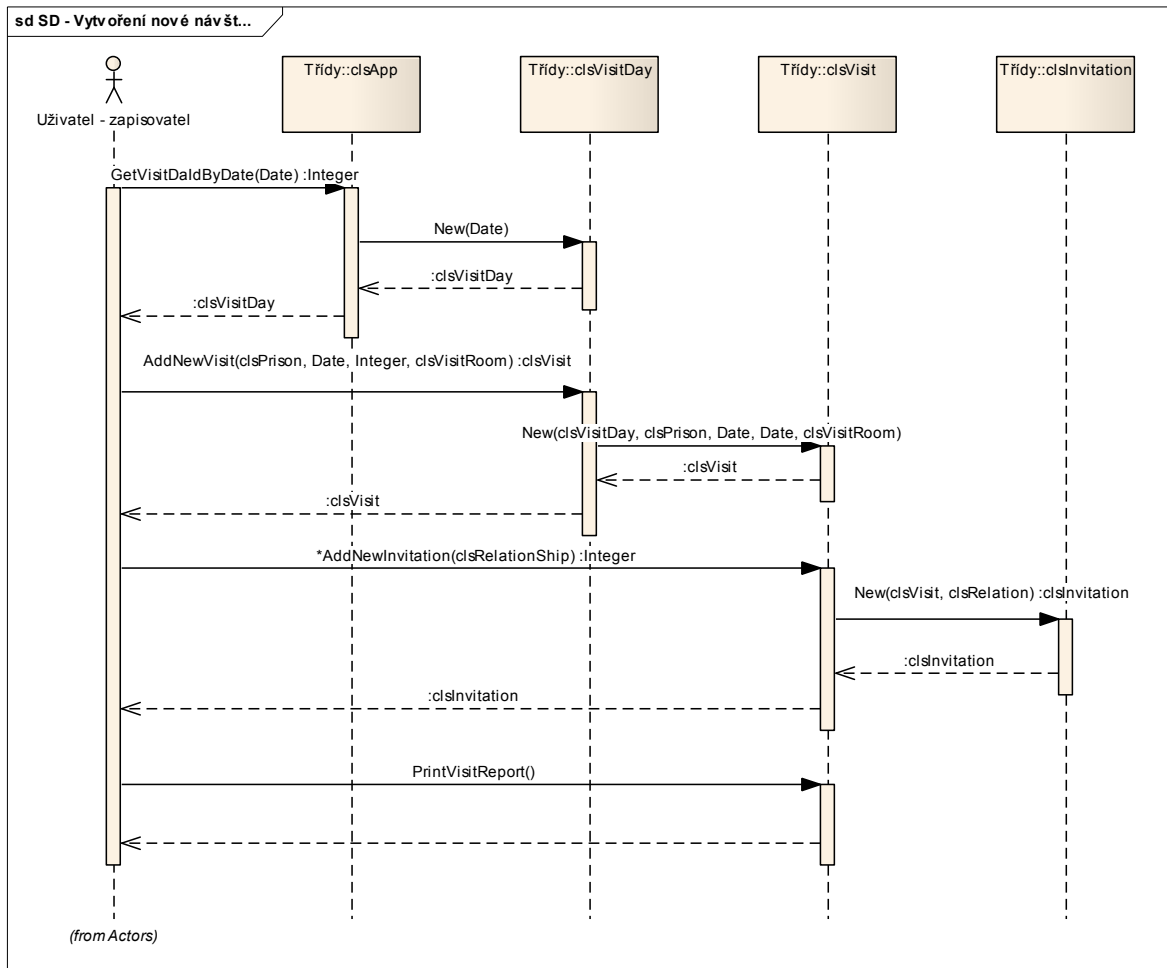
Tabulka 9 – Hlavní scénář Zrušení naplánované návštěvy

2.7 Sekvenční diagramy

Pro popis vzájemné interakce objektů v systému je výhodné použít diagram, který má vyjadřovací schopnost pro modelování spolupráce. Mezi takové patří sekvenční diagramy, které znázorňují jednotlivé objekty jako obdélníky v horní části diagramu. Z každého takového obdélníku vede směrem dolů svislá čára, čára života, mající za úkol reprezentovat život objektu. Zprávy mezi objekty jsou znázorněny vodorovnými šipkami mezi jednotlivými čarami života.

2.7.1 Sekvenční diagram Naplánování nové návštěvy

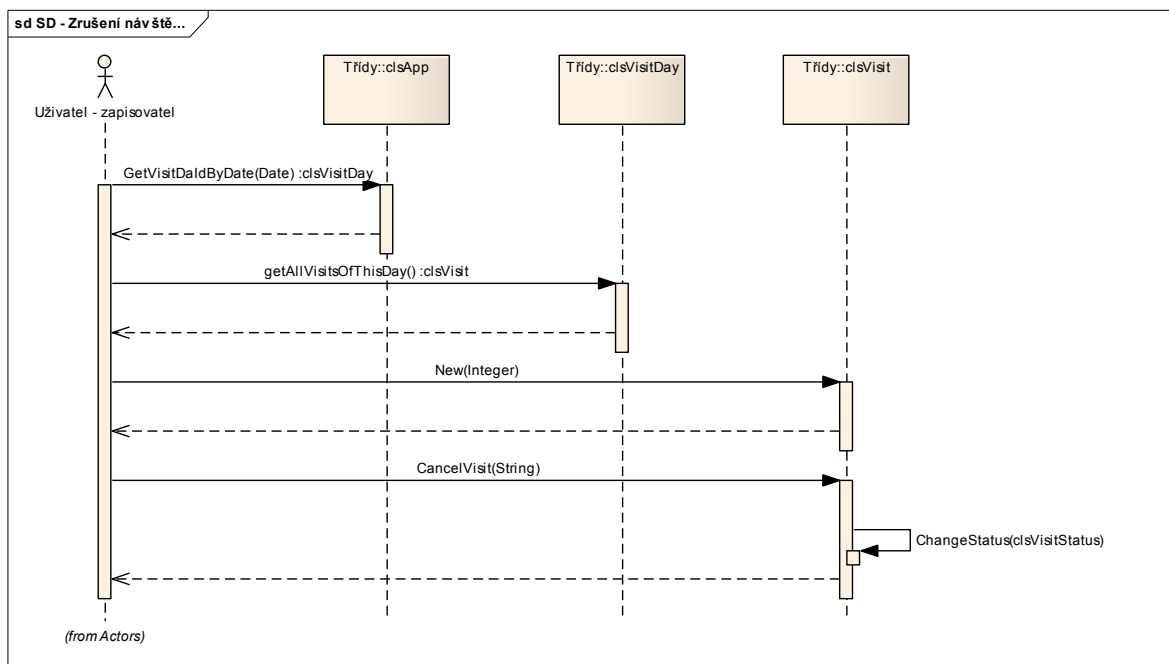
Na následujícím diagramu je znázorněna vzájemná spolupráce pomocí informačních toků mezi objekty během případu užití Zadání nové návštěvy.



Obrázek 8 - Sekvenční diagram Naplánování nové návštěvy

2.7.2 Sekvenční diagram Zrušení naplánované návštěvy

Další diagram vzájemné spolupráce zobrazuje spolupráci objektů v případě případu užití Zrušení naplánované návštěvy. Uživatel, který chce zrušit návštěvu, postupně komunikuje s objekty pro nalezení dne návštěv a následně pro nalezení konkrétního objektu naplánované návštěvy. Tomu pak zasílá požadavek na zrušení s vloženou zprávou obsahující odůvodnění takového kroku a objekt sám sebe označí jako zrušený a vrací uživateli zprávu o svém novém stavu.



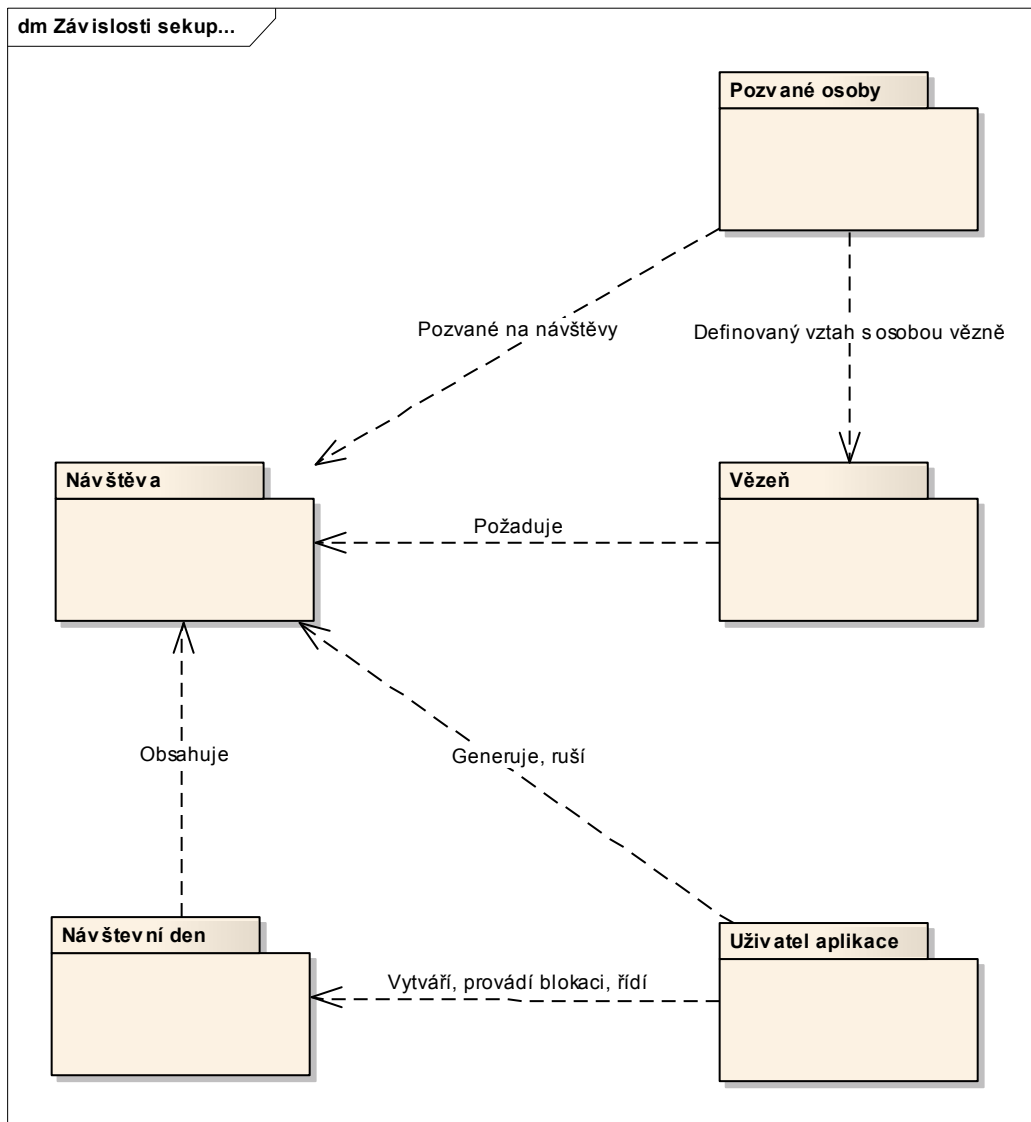
Obrázek 9 – Sekvenční diagram Zrušení naplánované návštěvy

2.8 Diagramy tříd

Nyní se seznámíme s diagramy tříd, které slouží jednak pro zachycení statické povahy systému, ale slouží také pro generování samotné struktury samotného programu nebo návrhu tabulek relační databáze. V případě popisu aplikace VISitor je použita forma zobrazení pomocí seskupení tříd, které dává pohled na popisovaný program jako na systém obsahující jednotlivé balíčky, které mají specifickou funkcionalitu, a mezi těmito balíčky probíhá také výměna služeb. Samotné balíčky obsahují objekty – třídy, které spolu logicky souvisí a komunikují mezi sebou.

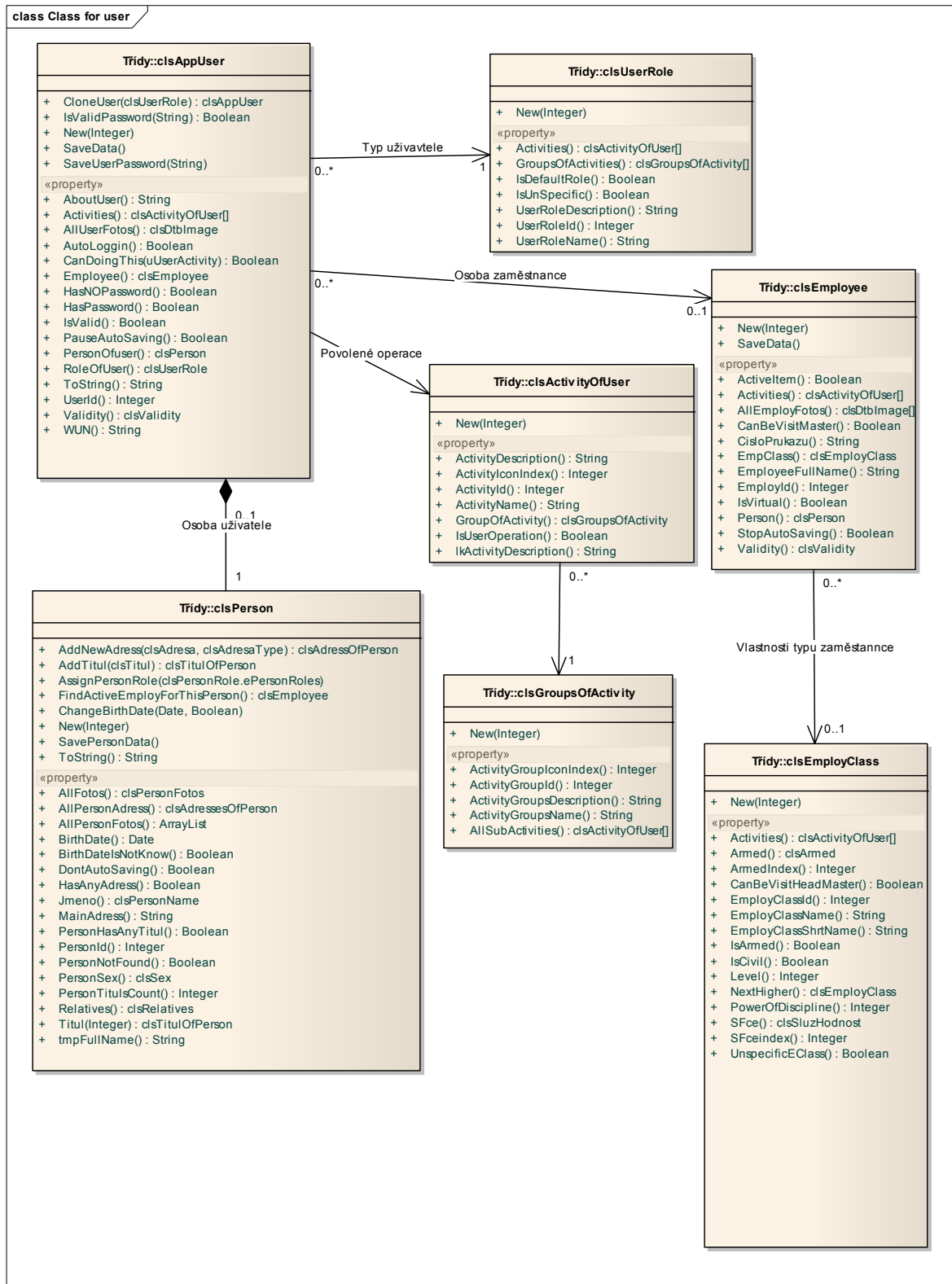
Následuje diagram, který popisuje hlavní myšlenku seskupení tříd pro funkcionalitu aplikace při Plánování návštěv vězněných osob. Dle tohoto diagramu lze vyvodit závislosti jednotlivých seskupení mezi sebou a z toho vyplývající asociace mezi jednotlivými

vnořenými třídami. Z následujících diagramů vnitřních schémat jednotlivých seskupení můžeme například usuzovat na existenci asociací mezi třídami *clsPrison* a *clsVisit*, *clsVisitDay* a *clsVisit* nebo asociací mezi třídami *clsInvitationPeople* a třídou *clsInvitation*.



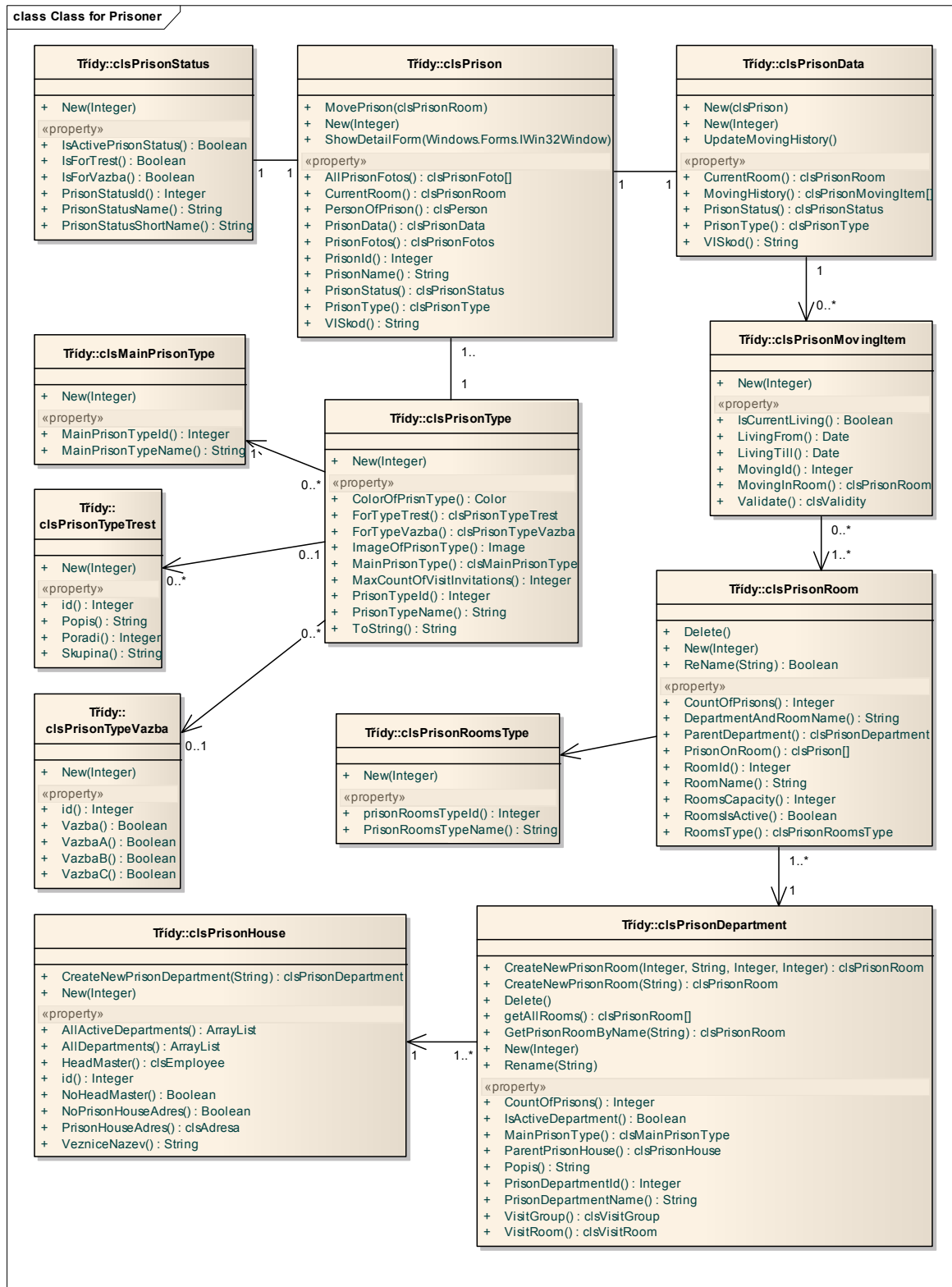
Obrázek 10 – Závislosti seskupení tříd

Následující diagram vyjadřuje vzájemné vztahy a rozhraní tříd pro zachycení modelu Uživatel aplikace. Z tohoto je zřejmé, že třída *clsAppUser*, která představuje objekt uživatele aplikace, je specializací objektu třídy *clsPeople*.



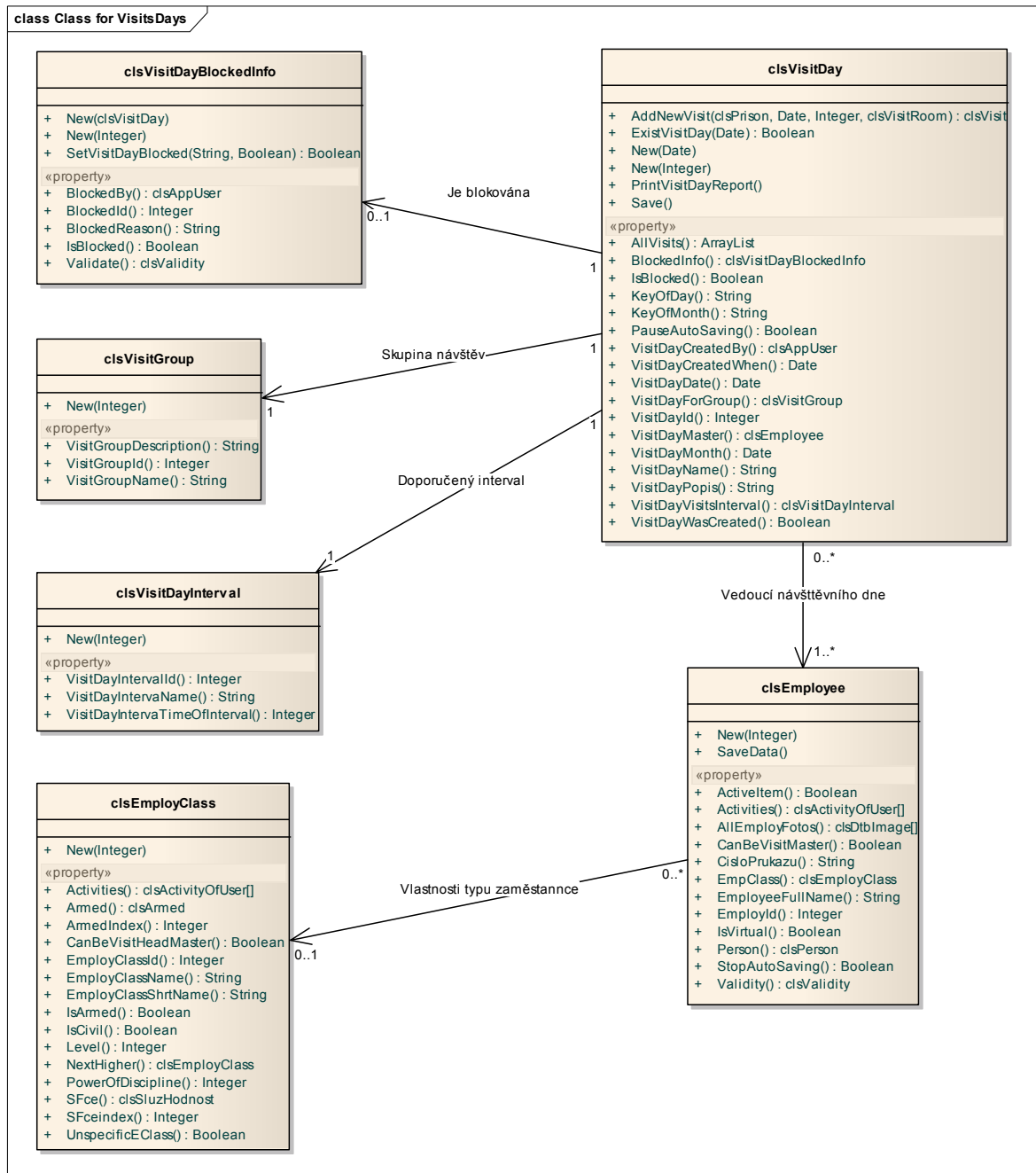
Obrázek 11 – Diagram třídy seskupení Uživatel aplikace

Dalším diagramem tříd je popis modelu tříd pro objekt osoby vězně. Ačkoliv to není na první pohled zřejmé, lze z tohoto modelu opět určit asociaci mezi třídami *clsPrison* a třídou *clsPerson*, a to pomocí veřejné vlastnosti objektu *PersonOfPrison*.



Obrázek 12 – Diagram třídy seskupení Vězeň

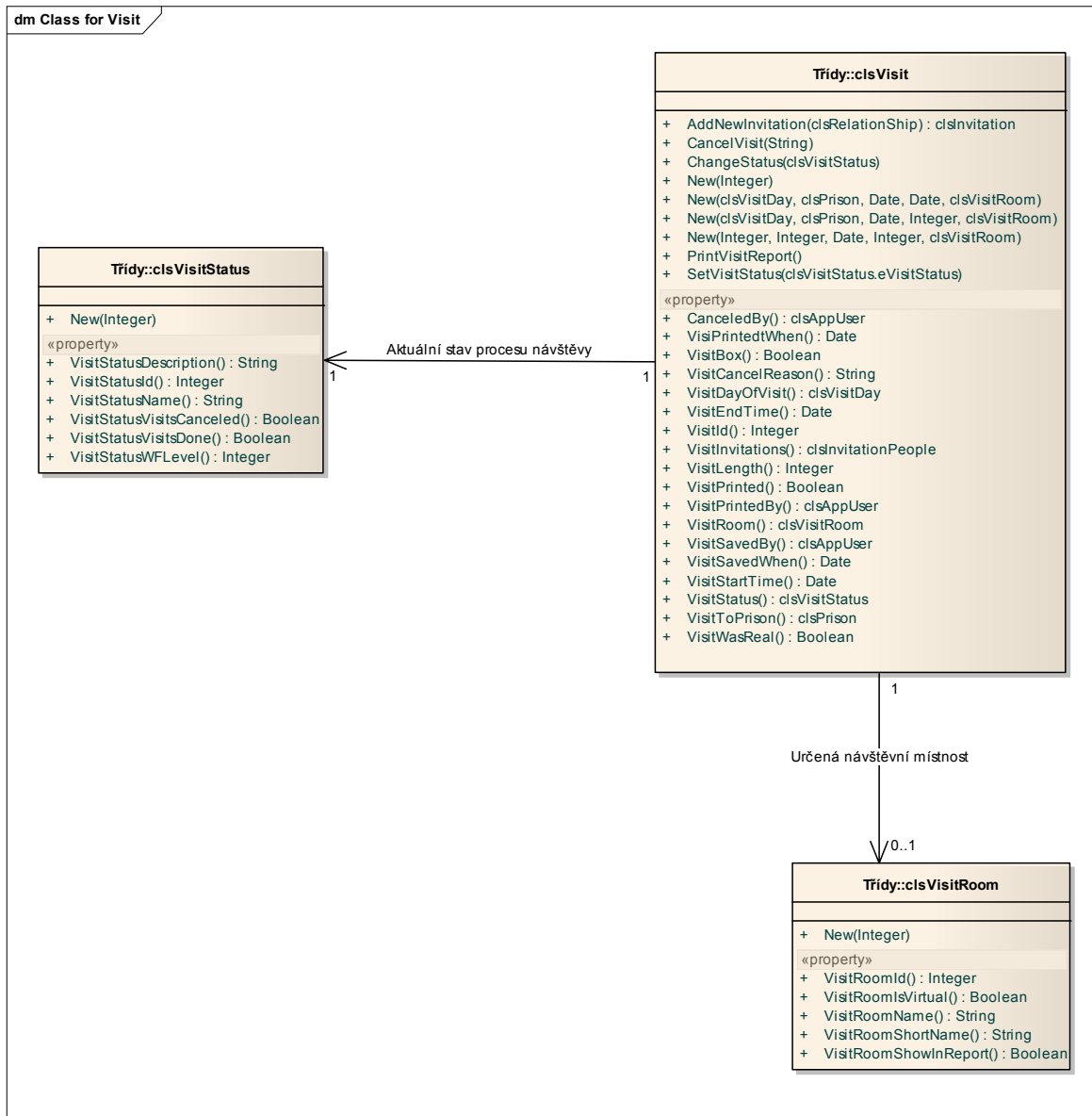
Následující diagram popisuje vnitřní schéma seskupení tříd Návštěvní den. Z toho zobrazení, můžeme usoudit, že zveřejněná operace *AddNewVisit* objektu *clsVisitDay* bude konstruktorem nových objektů třídy *clsVisit*.



Obrázek 13 – Diagram třídy seskupení Návštěvní den

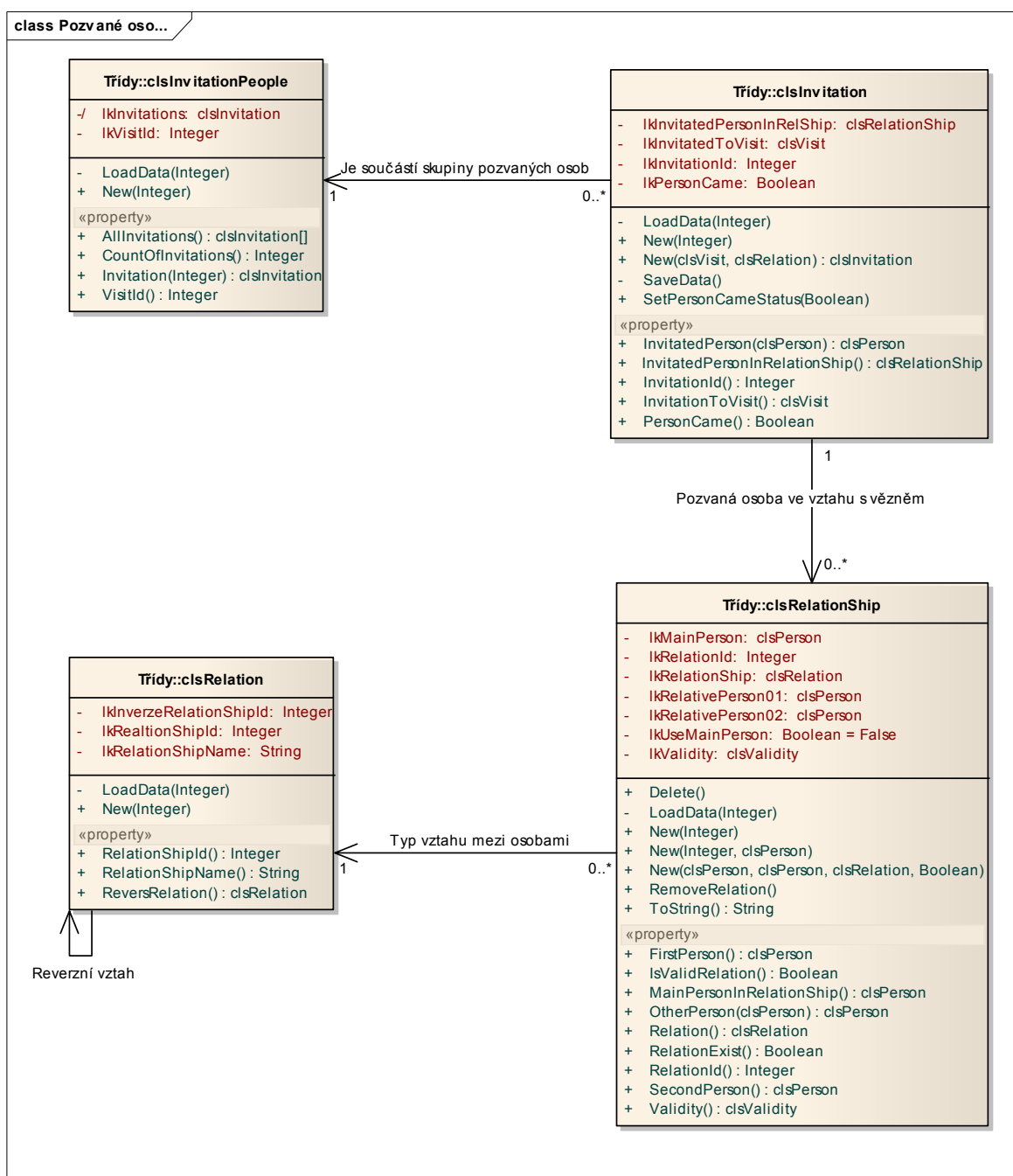
Další diagram představuje vnitřní schéma seskupení tříd pro objekt Návštěvy vězněné osoby. Opět z výčtu veřejných operací třídy *clsVisit* můžeme například vyvodit skutečnost, že veřejná operace *AddNewInvitation* bude konstruktorem pro nové objekty pozvaných osob k návštěvě, či zcela určitě bude existovat asociace s třídou *clsPrison* jako osobou

vězně, kterému je návštěva naplánována nebo asociace s třídou *clsAppuser*, jako uživatelem, který návštěvu naplánoval, vytisknul nebo změnil stav procesu realizace naplánované návštěvy.



Obrázek 14 – Diagram třídy seskupení Návštěva vězněné osoby

Následující diagram zobrazuje vnitřní vztahy jednotlivých komponent seskupení tříd pro objekt pozvaných osob na naplánovanou návštěvu. Z diagramu je zřejmá asociace na třídu clsVisit, která určuje, na kterou návštěvu byly osoby pozvané, nebo asociace třídy clsRelationship, která generuje objekt pro vztah mezi dvěma osobami, s třídou clsPerson. Další specifická asociace je asociace třídy clsRelation na sebe sama, a to z důvodu definice opačného (reverzního) vztahu mezi dvěma osobami. Například se jedná o definici vztahu mezi osobami rodiče a vlastního potomka.



Obrázek 15 - Diagram třídy seskupení Pozvané osoby

2.9 Data diagramy

Tato kapitola se zabývá návrhem ukládání a organizace dat informačního systému VISitor. Jak bylo uvedeno v úvodu této práce, z důvodu podmínky zadavatele a místních podmínek budou data uložena v relační souborové databázi *Microsoft Access*.

Struktura jednotlivých tabulek vychází z modelu tříd, kdy se konverzí jednotlivých atributů tříd, stanou sloupce tabulek a instance samotných objektů budou tvořit jednotlivé záznamy tabulek.

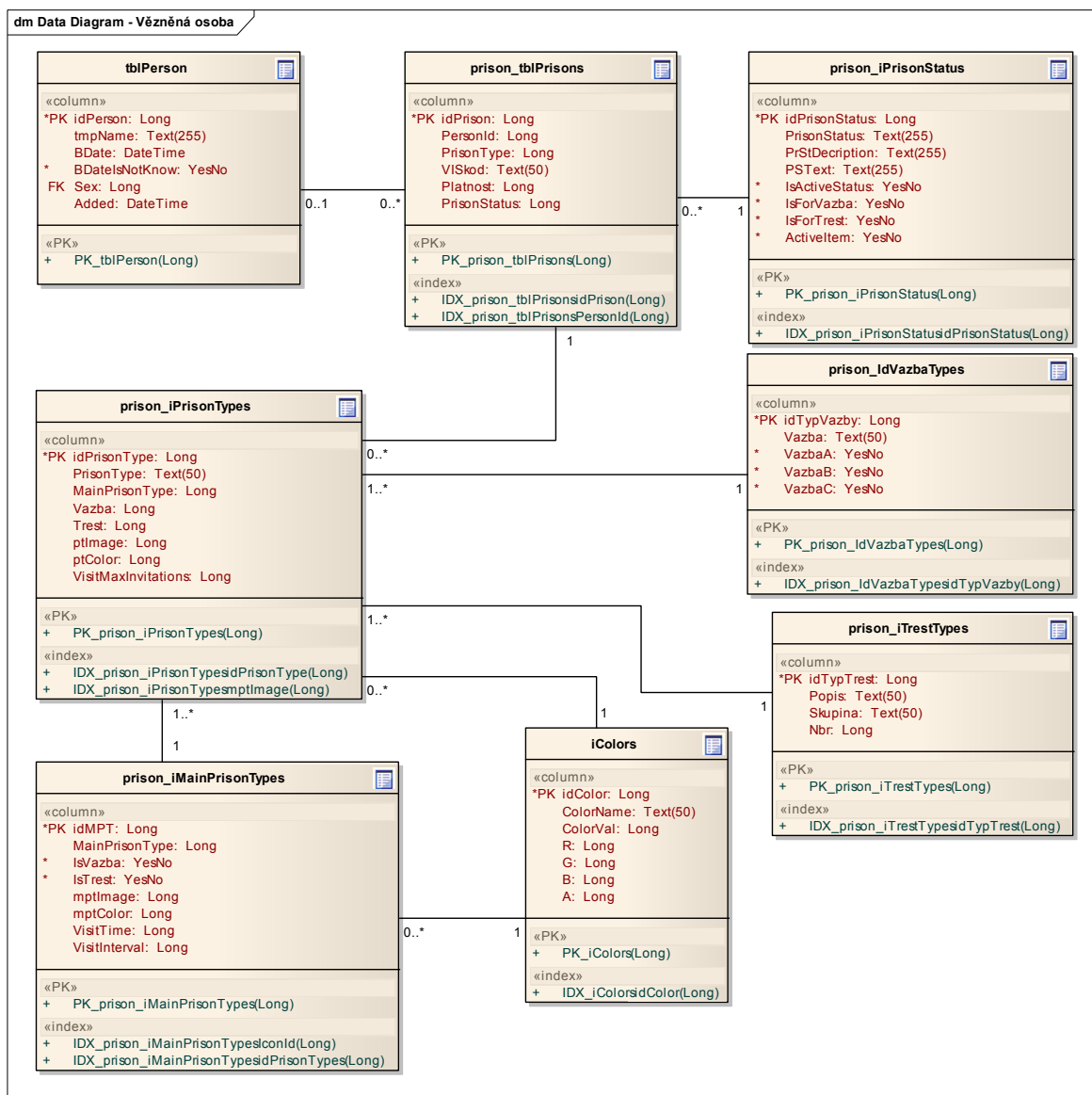
Jednotlivé typy asociací mezi třídami se do struktury databázových tabulek projeví jako typy vazby mezi primárními a cizími klíči v tabulkách. Například asociace s kardinalitou 1:1 vede většinou ke spojení do jedné tabulky. Nejčastěji používaná asociace s kardinalitou 1:N, znamená nutnost použití cizího klíče v pořízených tabulkách odkazující na záznam v hlavní tabulce. Asociační vazba N:M vyžaduje existenci takzvané asociační tabulky, která bude obsahovat, jak vlastní primární klíč, tak i oba cizí klíče z obou tabulek.

Následující data diagram zobrazuje jednotlivé databázové tabulky určené pro ukládání informací pro objekty uživatelů aplikace. V rámci tohoto procesu je možné ukládat data pro specifikaci vazby uživatele s existující osobou, definici jeho oprávnění provádět určité operace nebo uložení fotografií osoby uživatele.



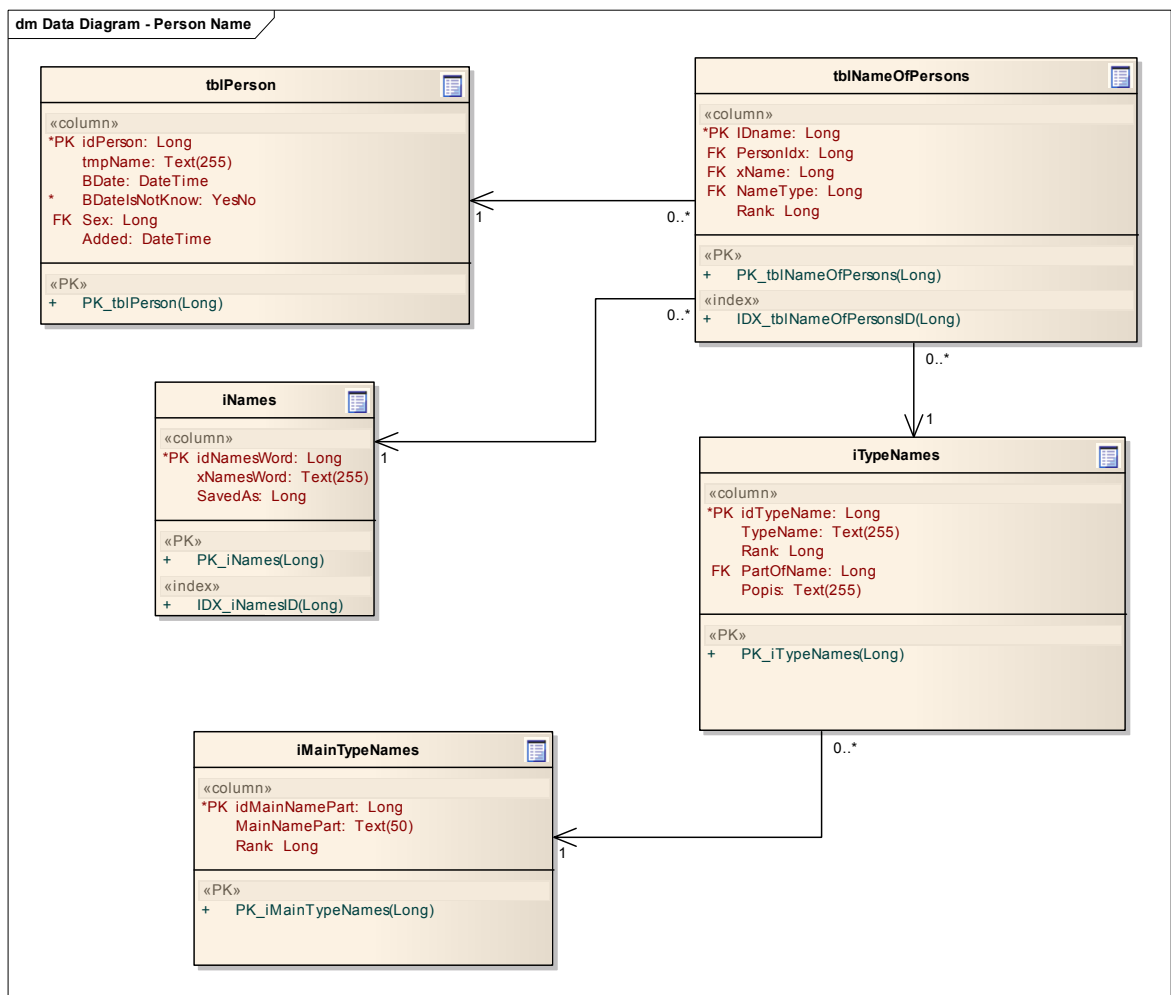
Obrázek 16 – Data diagram pro uložení objektu uživatele aplikace

Další obrázek zachycuje návrh databázových tabulek pro vytvoření struktury ukládání definice vlastností objektu vězněné osoby. Mezi ukládané vlastnosti patří propojení objektu vězně s osobou uloženou v tabulce *tblPerson*, specifikace typu věznění osoby a aktuální stav vězněné osoby.



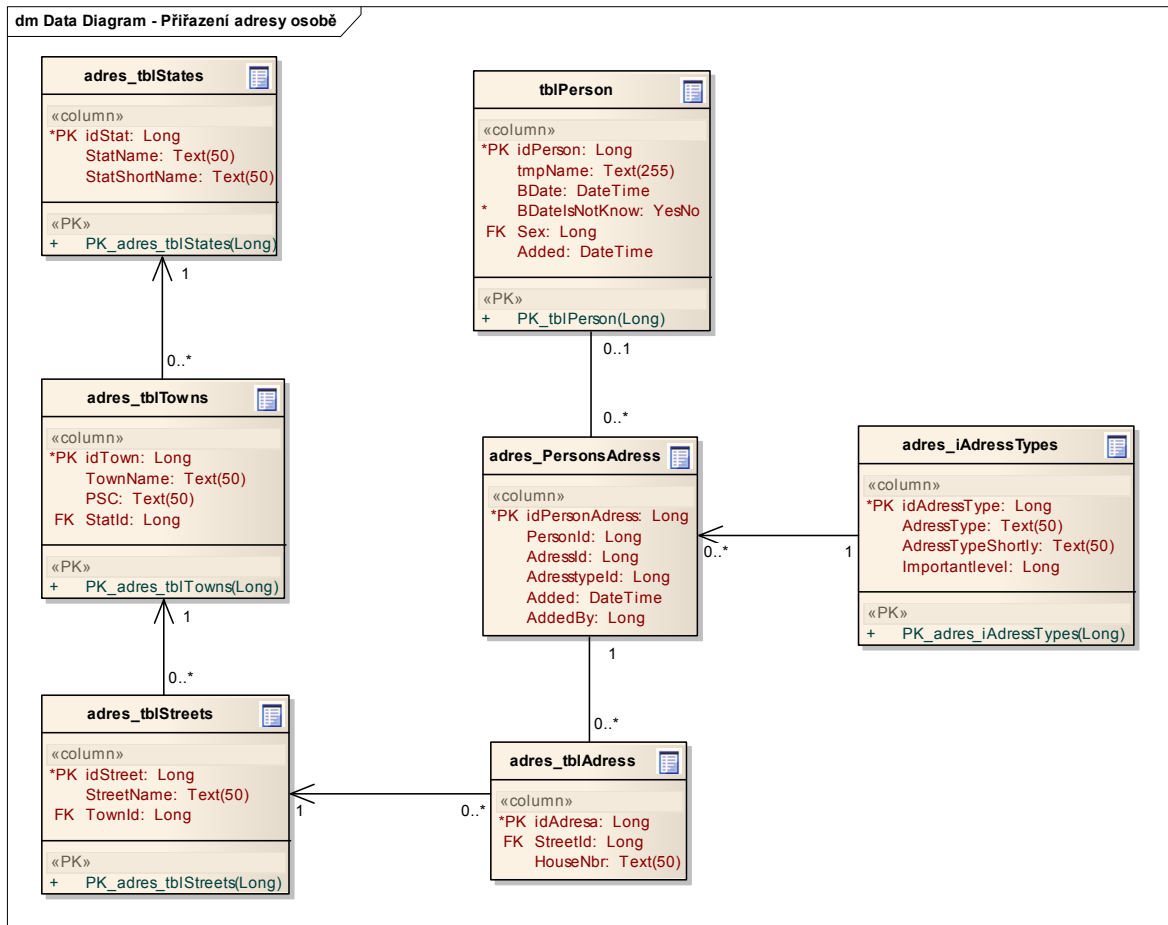
Obrázek 17 – Data diagram pro uložení dat definice vězněné osoby

Pro uložení jména osoby skládajícího se z několika slov jména a příjmení, byla vytvořena struktura tabulek, která umožní ukládat jednotlivá slova jména do jedné tabulky, přičemž každé takto uložené slovo má označení příslušnosti k části původního jména osoby. V praxi se jedná o převod asociační třídy do struktury relačních tabulek s kardinalitou N:M.



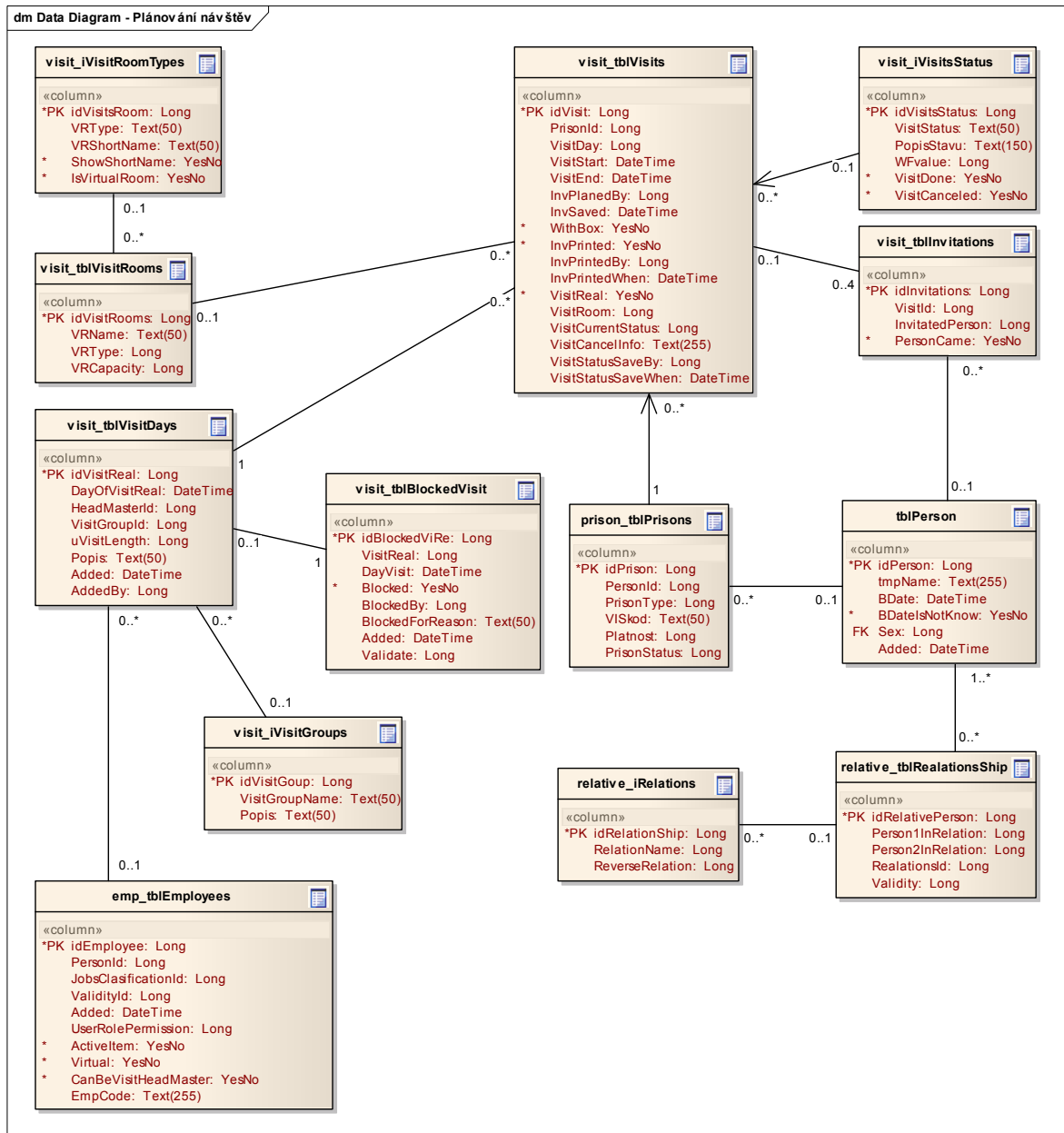
Obrázek 18 – Data diagram pro uložení struktury jména osoby

Následující diagram zobrazuje strukturu datového úložiště pro přiřazení uložené adresy k osobě s definicí jejího druhu.



Obrázek 19 – Data diagram pro přiřazení adresy osobě

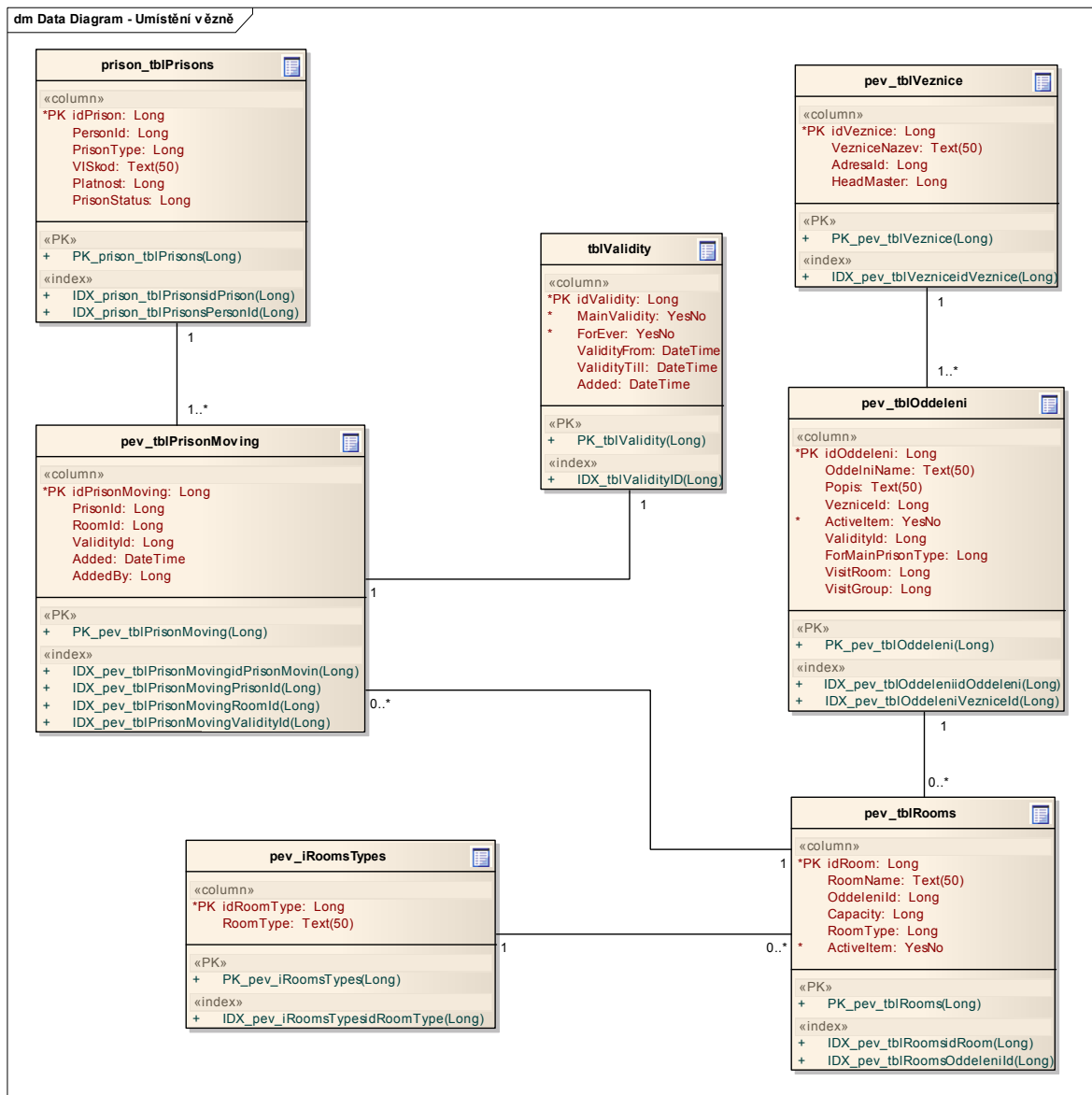
Další diagram vyjadřuje definici tabulek a jejich vzájemné propojení pomocí primárních a cizích klíčů pro uložení objektů představující jednotlivé naplánované návštěvy vězňených osob společně s parametry návštěvního dne a osob pozvaných k návštěvě jako návštěvníci.



Obrázek 20 – Data diagram pro Plánování návštěv

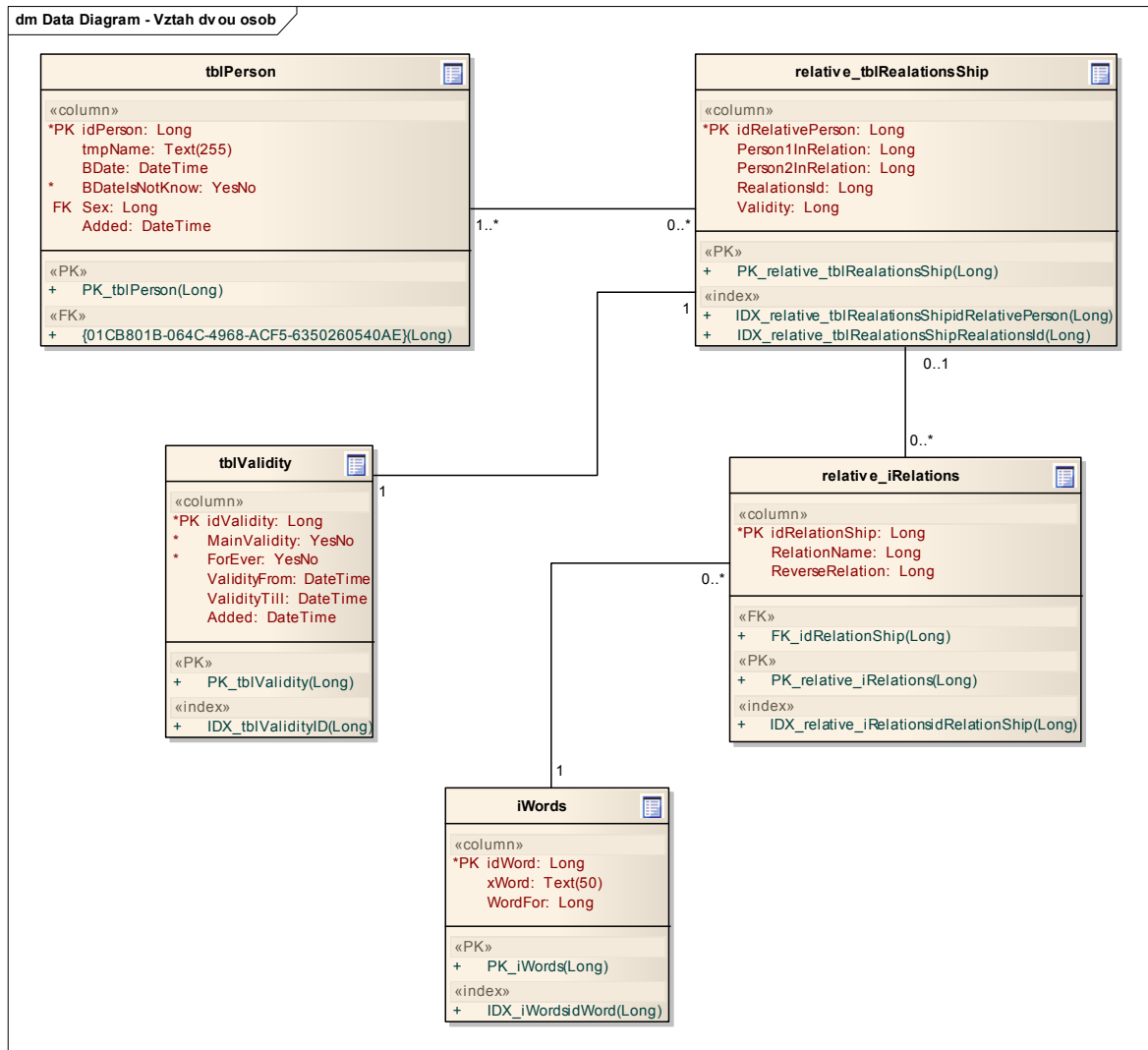
Následující data diagram popisuje strukturu a vzájemné relace mezi tabulkami pro uložení informací týkající se historie umístění vězňené osoby v rámci jedné organizační jednotky. Hlavní tabulkou v této funkcionalitě je tabulka *pev_tblPrisonMoving*, která obsahuje všechny záznamy o pohybu osoby. Platnost daného záznamu je dána hodnotami uloženými v propojené tabulce *tblValidate*, která slouží k definování platnosti údajů v rámci celé aplikace. Pokud se hledá aktuální umístění vězňené osoby, musí se projít platnosti všech

položek o umístění vybraného vězně a hledat takovou, která má definovanou platnost, platnou k aktuálnímu datu.



Obrázek 21 – Data diagram pro uložení historie ubytování vězně

Posledním data diagramem je struktura databázových tabulek pro uložení definice vzájemného vztahu dvou osob. Opět zde figuruje tabulka *tblValidate*, která obsahuje informace o platnosti definice vzájemného vztahu dvou osob. Dalším důležitým aspektem této funkcionality je dvojitá asociace tabulky *tblPerson* s tabulkou *relative_tblRealationsShip*, neboť ta definuje vazbu mezi dvěma odlišnými záznamy této tabulky.



Obrázek 22 – Data diagram pro uložení definice vzájemného vztahu osob

II. PRAKTICKÁ ČÁST

3 UŽIVATELSKÁ PŘÍRUČKA PROGRAMU

V této části textu diplomové práce bude čtenáři představena uživatelská příručka pro pochopení samotné funkcionality hotového programu, implementovaného na základě analytických kroků uvedených v předchozí teoretické části textu.

3.1 Příručka pro správce aplikace

Následující kapitola obsahuje informace pro uživatele s oprávněním správy uživatelských účtů, který je oprávněn provádět instalace nových programů na cílové stanici klienta.

3.1.1 Instalace a spuštění aplikace

Aplikace VISitor je dodána na instalačním nosiči, který obsahuje instalačního průvodce, který zajistí pohodlnou instalaci programu na cílový počítač. Průvodce se automaticky spustí po vložení instalačního média do diskové mechaniky. Pokud není povoleno automatické spouštění obsahu disků, pak je nutno průvodce spustit přímo aktivací programu *autorun.exe* umístěném v kořenovém adresáři instalačního media.

Po spuštění průvodce je provedena přítomnost prostředí *.NET Framework*¹ v potřebné verzi. Tato součást je potřebná pro samotný chod aplikace, neboť obsahuje potřebné knihovny. V případě, že není na cílovém počítači tento balíček detekován, bude instalace přerušena a uživateli bude nabídnuto stažení potřebného softwarového vybavení ze stránek výrobce, firmy Microsoft.

V další fázi instalace je volba umístění samotného programu. Jako výchozí umístění je instalátorem nabídnuta možnost vytvořit programovou složku ve standardním adresáři *Program Files*. Do tohoto adresáře se během instalace vytvoří podadresář VISitor a do této složky se umístí všechny potřebné soubory pro činnost aplikace. Jedná se zejména o samotný soubor s programem *visitor.exe* a soubor s textem nápovědy.

Při instalaci se na cílovou stanici klienta umístí i samotný databázový soubor *visitorddb.mdb*. tento databázový soubor se implicitně umísťuje do složky s uživatelskými dokumenty, kde je pro tento účel automaticky vytvořena složka *VisitorDtb*.

¹ Prostředí pro tvorbu a spuštění aplikací pomocí platformy .NET

Jako poslední krok je možnost umístění zástupce pro spuštění aplikace na plochu uživatele a zápis hodnot do registru *Windows* aktuálního uživatele o umístění databáze.

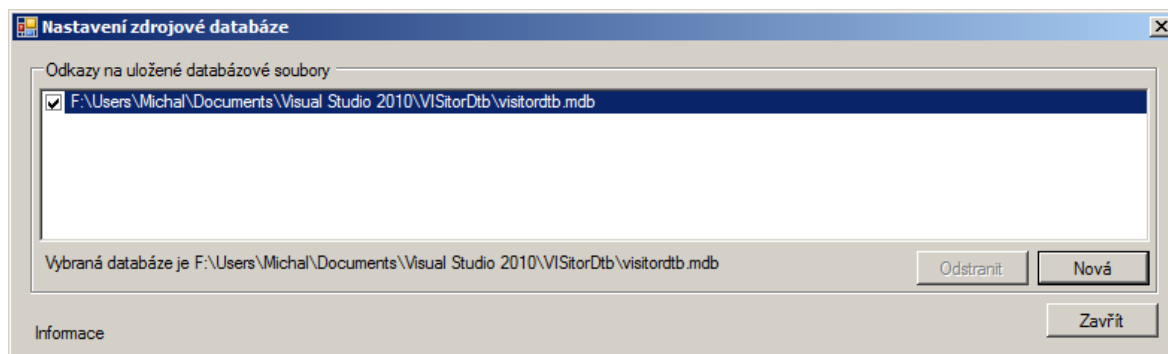
Tímto je instalace klienta aplikace VISitor úspěšně dokončena.

3.1.2 Umístění databázového souboru

Před tímto krokem je nutné rozhodnout, jakým způsobem bude aplikace VISitor provozována. V případě, že bude využívána v režimu klient-server, pak je nutné předem vytvořit sdílený adresář na místním diskovém serveru a omezit přístup do tohoto adresáře jen oprávněným uživatelům v místní doméně. Pak je možné umístit do tohoto adresáře databázový soubor *visitordtb.mdb*.

V případě, že není nutné využívat aplikaci ve víceuživatelském režimu, pak je možné soubor umístit do libovolné složky na lokálním disku, do které má uživatelé povolen přístup s možností zápisu nových dat.

Změnu užívaného databázového souboru je možné provést pomocí dialogového okna zobrazeného pomocí příkazu *Zdrojová databáze*, v položce *Nastavení*, v hlavní liště aplikace.



Obrázek 23 – Formulář pro nastavení zdrojové databáze

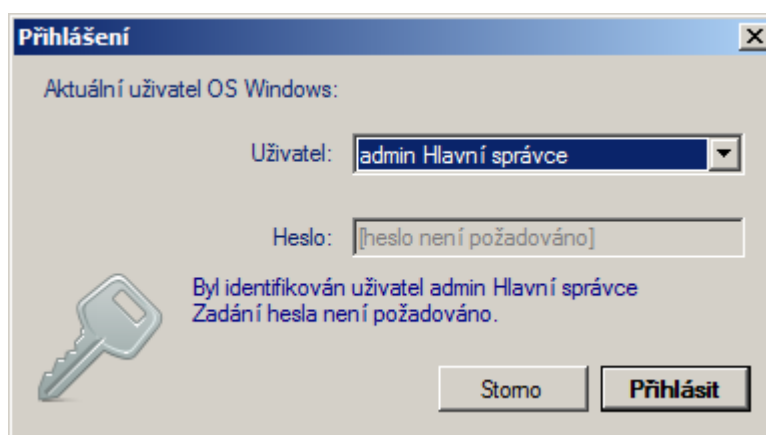
V tomto dialogovém okně je možné změnit používaný databázový soubor. Změna se provádí zaškrtnutím příslušného pole v seznamu odkazů databázových souborů.

Případně je možno přidat nový odkaz pomocí dialogu pro výběr databázového souboru nebo odstranit již neplatný nebo nepoužívaný odkaz. Všechny zobrazené informace jsou automaticky synchronizovány s uživatelským registrem Windows.

Změna používání databázového souboru se úplně projeví až po znovuspuštění aplikace, proto je uživateli v případě, že došlo ke změně zdrojové databáze nabídnuta možnost aplikaci restartovat.

3.1.3 První spuštění aplikace

V základním, počátečním stavu, je v databázovém souboru přítomen pouze jeden uživatelský účet. Je to účet *Hlavní správce*, který slouží jen pro první přihlášení, neboť tento uživatelský účet nemá nastaveno žádné heslo. Je doporučeno po prvním přihlášení do aplikace pomocí tohoto účtu, vytvořit minimálně jeden další nový uživatelský účet typu správce a původní účet zrušit, aby se předešlo přístupu do aplikace neautorizovaným osobám.

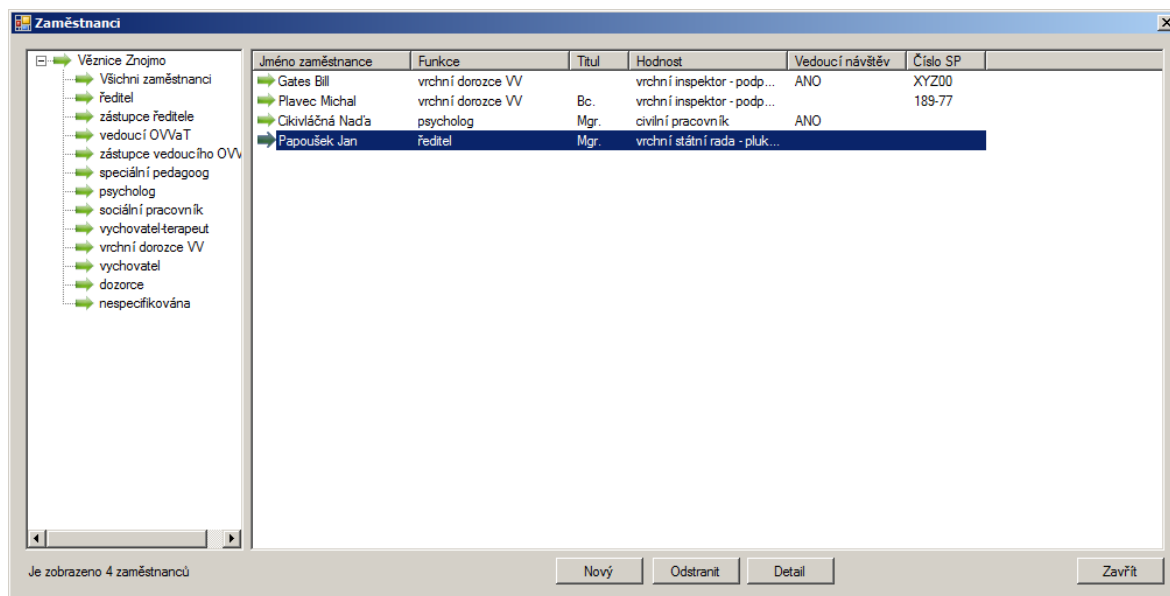


Obrázek 24 – Prvotní přihlášení do aplikace

3.1.4 Správa zaměstnanců

Pro potřeby plánování návštěv a evidenci uživatelů aplikace, je v programu implementována možnost evidence osob v pracovním poměru s organizační jednotkou – věznicí. Jedná se zejména o pracovníky Oddělení výkonu vazby a trestu, kteří pracují s daty vězňených osob a jiné oprávněné osoby, které mají nějaký význam pro administrativní činnosti agendy vězňených osob.

Editace zaměstnanců se provádí pomocí formuláře *Zaměstnanci*, který lze zobrazit pomocí příkazu v menu hlavního okna *Moduly - Správa zaměstnanců - Procházet zaměstnance* nebo aktivací příslušného odkazu ve formuláři *Výběr modulu činností*.



Obrázek 25 – Formulář pro přehled zaměstnanců

Formulář je tvořen dvěma dominantními ovládacími prvky – stromový prohlížeč se seznamem jednotlivých skupin zaměstnanců a listovým seznamem pro zobrazení zaměstnanců patřící do skupiny, která je vybrána ve stromovém prohlížeči. V případě, že je zvolen kořen stromu – název věznice, budou zobrazeni všichni platní zaměstnanci evidovaní v informačním systému VISitor.

K dalším ovládacím prvkům patří tlačítka *Nový*, *Odstranit* a *Detail*. Jak jejich název napovídá, jedná se o příkazy pro činnost se záznamy zaměstnanců. Tlačítkem *Nový*, zobrazíme modální dialogové okno pro zadání nové osoby zaměstnance, tlačítkem *Odstranit*, je možné vybraný profil zaměstnance odstranit a pomocí tlačítka *Detail* je možné zobrazit formulář pro zobrazení a změnu parametrů vybrané osoby zaměstnance.

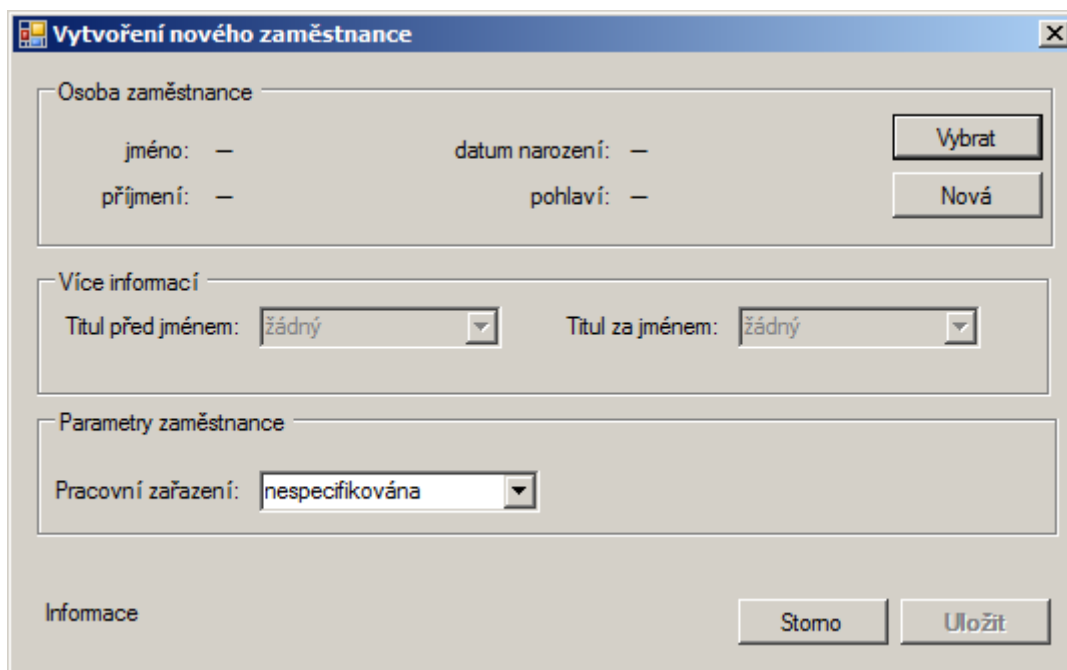
Obrázek 26 – Formulář Detail zaměstnance

V tomto dialogovém okně je možné provádět změny parametrů zaměstnance přiřazené osoby. Jedná se zejména o změnu pracovního zařazení, které se provádí pomocí výběru z možností uvedených ve výběrovém seznamu, evidenčního kódu zaměstnance a zaškrtnutím volbou parametru pro nastavení oprávnění zaměstnance pro roli vedoucího návštěvního dne. K dalším parametrům patří možnost nahrání fotografie nebo podobenky osoby zaměstnance, změnu akademického titulu, data narození nebo označení osoby jako virtuálního zaměstnance. Tohoto je využíváno zejména pro účely testování nových funkcionalit aplikace. Tlačítkem Zavřít se uloží všechny změny do databáze a dialogové okno se uzavře.

3.1.5 Vytvoření nového zaměstnance

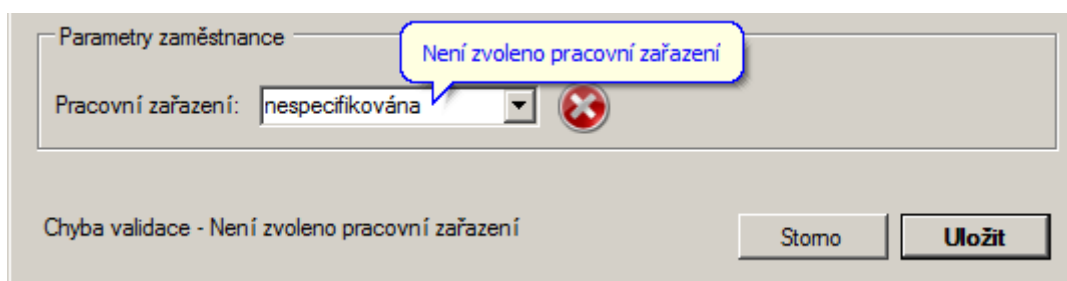
Dialogový formulář umožní uživateli vytvořit profil nového zaměstnance věznice, lze vyvolat pomocí nabídky v menu *Moduly - Správa zaměstnanců – Vytvořit zaměstnance*. Jako první je nutné specifikovat osobu, která se má stát evidovaným uživatelem. Na výběr jsou dvě možnosti, jak lze osobu definovat. První možností je pomocí tlačka Vybrat,

zobrazit vyhledávací formulář, pomocí kterého lze hledat konkrétní osobu podle zadaných kritérií a posléze ji označit jako vybranou a předat ji jako aktuální osobu zaměstnance do původního formuláře. Druhou možností je vytvoření nové osoby pomocí tlačítka *Nová*, které zobrazí dialogové okno pro vložení nové osoby do systému.



Obrázek 27 – Dialog pro vložení nového zaměstnance

K dalším krokům patří možnost definování akademických titulů, kterých je osoba nositelem a specifikace pracovního zařazení. Tento údaj je povinný, neboť vytváříme profil nového zaměstnance, a proto je nezbytné definovat tento parametr. Stiskem tlačítka *Uložit* je zahájen proces validace zadaných dat. Pokud je nalezena chyba v zadaných údajích, je uživatel o této skutečnosti informován pomocí textu zobrazeného v bublinkové nápovědě, prvku s chybovou hodnotou a tento prvek je označen ikonkou. Pokud během procesu validace není nalezena žádná chyba, pak je vytvořen nový objekt zaměstnance. Tlačítkem *Storno* je možné dialog uzavřít bez uložení nových dat.



Obrázek 28 – Zobrazení chyby validace

3.1.6 Správa uživatelských účtů

Přístup k funkcím aplikace VISitor mají jen registrované osob, kterým byl vytvořen uživatelský účet pro autorizaci vstupu. Pro účely správy uživatelských účtů je program vybaven vlastním dialogovým formulářem, který zobrazuje aktuální uživatelské účty s možností editace jejich parametrů. Tento formulář lze zobrazit pomocí nabídky menu hlavního okna *Moduly – Správa uživatelů – Procházet uživatele* nebo pomocí odpovídajícího příkazu ve formuláři *Výběr modulu činností*.

Obrázek 29 – Formulář pro editaci uživatelských účtů

Formulář uživatelé je tvořen listovým seznamem uživatelských profilů, ovládacích prvků pro zobrazení parametrů vybraného uživatele a několika příkazovými tlačítky pro editaci položek. Výběrem položky uživatele v listovém seznamu dojde k načtení hodnot vlastností vybraného profilu uživatele. Jedná se zejména o parametry definující oprávnění a platnost uživatelského účtu, parametry autentizace a heslo vybraného uživatele pro ověření uživatele při přihlašování do informačního systému VISitor. Je možné také nastavit vhodnou fotografii uživatele pro snadnější orientaci v seznamu uživatelů. Typ oprávnění

uživatele lze jednoduše změnit pomocí výběru požadované hodnoty z výběrové roletky ovládacího prvku. Pokud je provedena změna typu oprávnění, pak je na pozadí aplikace provedena sekvence několika kroků: Nejprve je vytvořena kopie aktuálního profilu uživatele s novým oprávněním. Této kopii jsou předány všechny vlastnosti původního objektu, kterému bude následně ukončena platnost, a tím tak bude zrušen. Tato operace je nutná pro zachování integrity operací provedených uživatelem s původním oprávněním.

Další skupinou ovládacích prvků jsou prvky pro definici platnosti uživatelského profilu. Vhodnou volbou těchto prvků lze nastavit uživatelskému profilu určitou „životnost“, po kterou bude uživatelský profil platný. Jednou z možností je nastavení stálé platnosti, druhou volbou je možnost definovat hraniční okamžiky intervalu platnosti. Pokud není uživatelský profil platný, pak není možno se pomocí tohoto profilu do aplikace přihlásit.

Následuje skupina ovládacích prvků, ve které lze povolit přihlašování uživatele bez nutnosti zadat heslo a prvky pro definici parametrů autentizace uživatele. Automatická autentizace uživatele probíhá pomocí kontroly uloženého jména uživatele operačního systému a jménem aktuálně přihlášeného uživatele. Pokud je při procesu spouštění aplikace nalezena shoda těchto jmen, pak je ve formuláři pro přihlášení nabídnut tento uživatelský profil jako výchozí. Pro nastavení je možné zvolit jméno aktuálního uživatele operačního systému nebo toto jméno lze zapsat ručně do textového pole.

Další volbou je editace hesla uživatele, kdy je možné uživatelské heslo smazat nebo změnit. Hodnotu hesla není možné nějakým způsobem zjistit, neboť se nikde neukládá a pro ověření správnosti hesla se využívá pouze kontrolní hashová hodnota hesla, ze které není možno původní heslo zjistit.

Na spodní části formuláře jsou příkazová tlačítka pro vytvoření nového uživatele, zrušení profilu aktuálního uživatele a uzavření formuláře.

3.1.7 Vytvoření nového uživatele

Nový uživatel se zadává pomocí dialogového formuláře *Vytvoření nového uživatele*, který lze zobrazit buď volbou příslušné položky ve formuláři *Výběr modulu činnosti* anebo pomocí menu hlavního okna aplikace *Moduly – Správa uživatelů – Vytvořit uživatele*.

Jedná se o klasický dialogový formulář pro vytvoření nového profilu přiřazením role vybrané osobě. I v tomto případě je nutné nejprve specifikovat osobu nového uživatele. Můžeme jednak pomocí příkazového tlačítka *Vybrat* zobrazit vyhledávací formulář,

pomocí kterého najdeme osobu, jenž je v systému již evidovaná, anebo pomocí tlačítka *Nová*, vytvoříme pomocí dialogu novou osobu. Každým z těchto uvedených případů vybereme osobu, které chceme přiřadit uživatelské oprávnění. V případě nutnosti změnit některé základní údaje o osobě, můžeme využít tlačítka *Upravit*, které zobrazí dialogové okno pro změnu jména osoby, hodnoty data narození a specifikaci pohlaví osoby.

Dalším krokem je výběr typu uživatelského oprávnění. Odpovídající hodnotu vybereme z nabídky výběrového seznamu.

Vytvoření nového uživatele

Osoba uživatele

jméno: **Petr** datum narození: **11.5.1990** Vybrat

příjmení: **Igráček** pohlaví: **muž** Nov Upr

Parametry uživatele

Typ uživatele: **nespecifikovaná** není specifikovaná

Přihlašování uživatele do systému VISitor

Uživatel nemusí zadat heslo Heslo:

Uživatel musí zadat heslo Kontrola hesla:

Identifikace uživatele

uživatel je aktuálně přihlášen jako Michal-PC-Win7\Michal

jméno uživatele v OS Windows:

Připraven. Vyberte prosím osobu uživatele. Storno Uložit

Následuje možnost nastavení parametrů pro způsob přihlašování do informačního systému. Můžeme zde nastavit možnost, zda bude uživatel povinen při vstupu do systému zadávat své uživatelské heslo, anebo bude vstup možný bez jeho zadání. Pokud je vybraná volba povinného zadání hesla, pak jsou zpřístupněna pole pro zadání hesla, která je nutné vyplnit stejným řetězcem představující heslo nového uživatele.

Další skupinou prvků jsou prvky pro nastavení hodnoty uživatelského jména v prostředí operačního systému. Toto jméno se bude používat pro detekci výchozího uživatele pro přihlašovací dialog. Pokud je nově zadávaný uživatel aktuálním uživatelem přihlášeným v operačním systému, pak zvolíme první možnost, anebo zvolíme druhou volbu a jméno uživatele je možné zadat pomocí klávesnice.

V tomto okamžiku je možné tlačítkem *Uložit* provést samotné uložení zadaných dat a vytvoření objektu nového uživatele. Stiskem tlačítka se nejprve provede validace zadaných dat. Pokud je procesem validace nalezena nějaká chyba – například chybně zadané heslo uživatele, je uživatel o této skutečnosti informován zobrazením textu v bublince lokální nápovědy prvku, ve které byla nalezena chyba, označením tohoto prvku varovnou ikonou a zobrazením textu v informačním prvku. Pokud je vše v pořádku, jsou zadaná data nového uživatele uložena. Tlačítkem *Storno* je formulář uzavřen bez uložení nových dat.

3.1.8 Typy uživatelů

Pro zdárný chod aplikace ve víceuživatelském režimu je výhodné rozdělit uživatele skupin podle typu prováděných operací a jiné operaci jim odepřít, aby svým zásahem nemohli způsobit nějaké ztráty v oblasti integrity dat. V případě informačního systému byly definovány tyto typy uživatelských oprávnění:

Název role	Popis role
Bez přístupu	Není specifikovaná, osoba bez přístupu
Default	Označení uživatelské role s oprávněním podle pracovního zařazení
Tester	Absolutní pravomoc nad všemi daty z důvodu testování funkcionality
Správce	Správce aplikace a uživatelských účtů
Zapisovatel	Zapisovatel údajů, vytváří důležité nové položky
Editor	Oprávnění měnit uložené údaje, vytváří slabé položky
Čtenář	Uživatel s oprávněním jen číst uložená data

Tabulka 10 – Výčet uživatelských rolí

- *Bez přístupu* - Tato role určena pro označení těch osob, které nemají žádný přístup k datům a aplikaci
- *Default* - uživatel, který má nastaven tento uživatelský profil přejímá oprávnění manipulovat s objekty podle zásad stanovených podle pracovního zařazení. Z toho vyplývá, že uživatel musí být zároveň i aktivní zaměstnanec
- *Tester* – tento profil není určen pro přidělení zaměstnancům, neboť jeho oprávnění odporuje bezpečnostním zásadám, protože uživatel, který je nositelem tohoto

oprávnění má možnost přistupovat ke všem zdrojům informačního systému bez omezení

- *Správce* – uživatelský profil určený pro spravování uživatelských účtů, administrativy zaměstnanců a globálního nastavení
- *Zapisovatel* – uživatel s oprávněním vytvářet nové záznamy velké důležitosti a měnit data
- *Editor* – uživatel, který může měnit data a vytvářet nové záznamy malé důležitosti
- *Čtenář* – uživatel s oprávněním pouze číst uložena data

Pojmem nové záznamy s velkou důležitostí je myšleno vytváření nových objektů, které mají velký význam na činnost informačního systému, například vytvoření nového dne návštěv.

3.1.9 Nastavení vlastností aplikace

Pomocí dialogu Nastavení aplikace je možné změnit některé parametry činnosti programu. Tento formulář je možné zobrazit pomocí příkazu v nabídce hlavního menu aplikace Moduly – Správa aplikace – Nastavení aplikace.

The image shows a screenshot of a software dialog box titled "Nastavení aplikace". It has four tabs: "Nastavení domácí věznice" (selected), "Zabezpečení aplikace", "Nastavení programu", and "Nastavení návštěv". The "Nastavení domácí věznice" tab contains the following fields and controls:

- Základní informace:**
 - Jméno aktuální věznice:
 - Ředitel(ka) věznice:
- Adresa věznice:**
 - Ulice: **Dyjská** Číslo: **4**
 - Město: **Znojmo** PSČ: **66902**

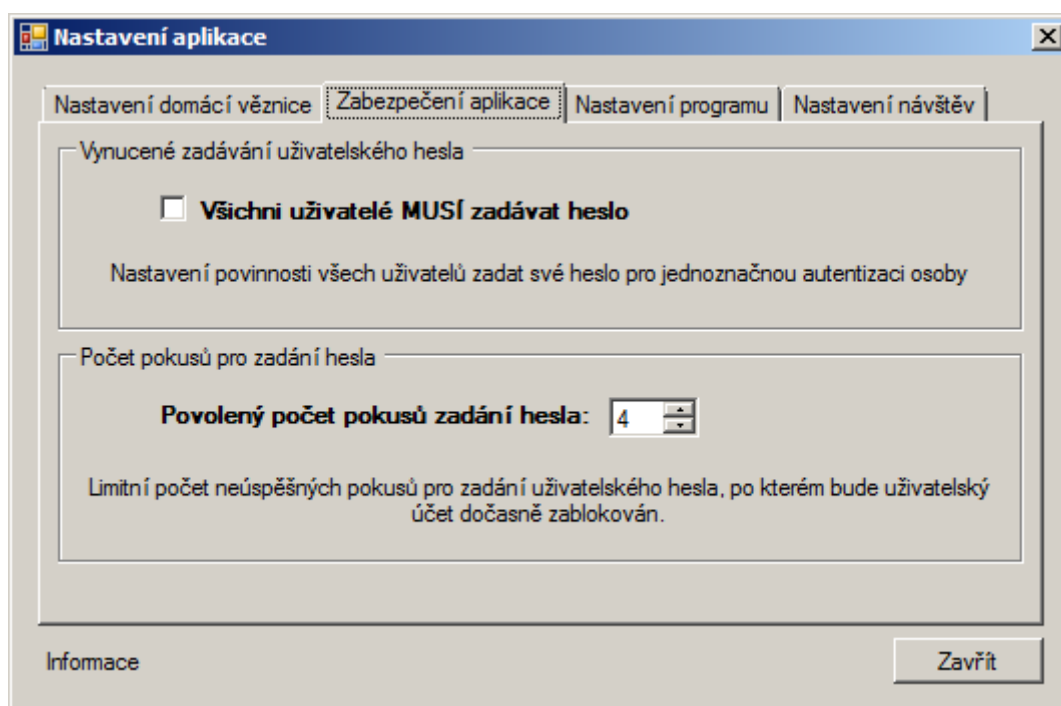
At the bottom right of the dialog is a button labeled "Vložit adresu". At the bottom left is the text "Informace" and at the bottom right is a button labeled "Zavřít".

Obrázek 30 – Formulář Nastavení aplikace

Tento formulář slouží k nastavení některých proměnných, které ovlivňují chod celého informačního systému. Na formuláři jsou ovládací prvky seskupeny podle svého významu

do jednotlivých karet. První karta *Nastavení domácí věznice* slouží ke specifikování názvu organizační jednotka, ve které je aplikace používána. K těmto údajům patří název věznice, adresa věznice a osoba zaměstnance nejvýše postavená v hierarchii zaměstnanců. Tyto hodnoty jsou používány při generování tiskových sestav nebo při zobrazení ubytovací struktury věznice.

Druhá karta obsahuje prvky pro stanovení úrovně zabezpečení při přihlašování uživatelů. Jedná se zejména o nastavení globální povinnosti všech uživatelů zadat své heslo při vstupu do informačního systému a definice maximálního počtu neúspěšných pokusů při zadávání uživatelského hesla.

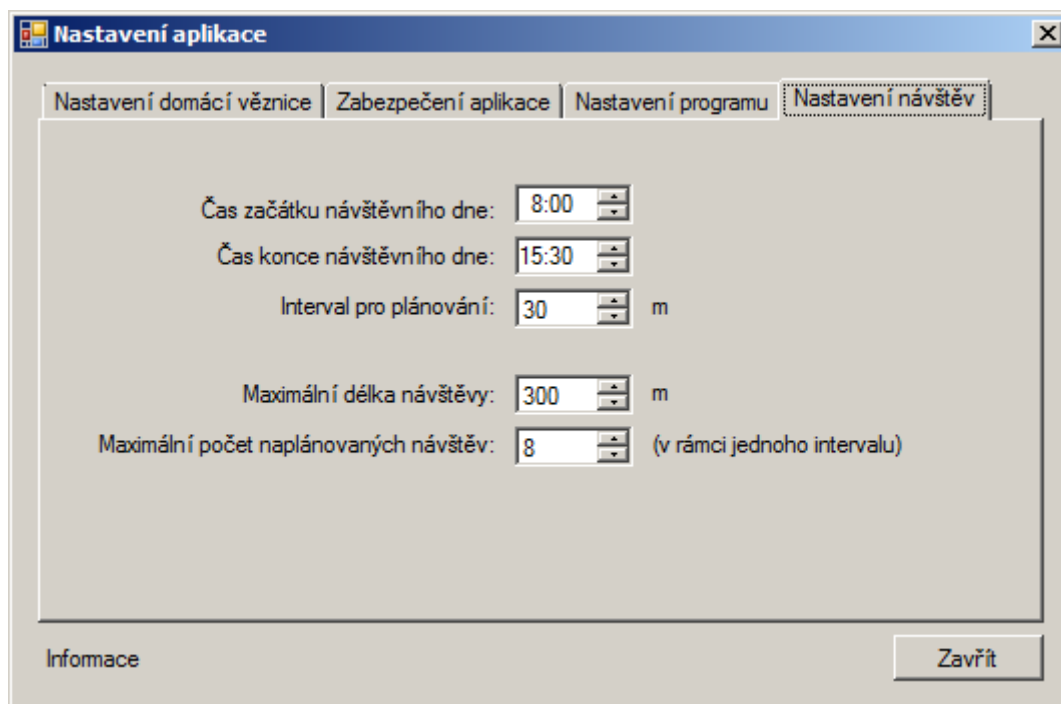


Obrázek 31 – Nastavení aplikace karta Zabezpečení aplikace

Třetí karta slouží pro nastavení názvu programu a popisu aplikace, který se objeví v titulku okna spuštěné aplikace. Další v této kartě nastavitelné vlastnosti aplikace je možnost aktivování nápovědy ovládacích prvků, která generuje text nápovědy pro každý ovládací prvek a v případě odchyzení kombinace kláves Shift + F1 zobrazí text nápovědy ve status baru hlavního okna. Další prvky jsou určeny pro potřeby ladění aplikace. Jedná se o povolení synchronizace databáze ovládacích prvků a aktivace módu tvorby nápovědy ovládacích prvků.

Poslední karta obsahuje prvky pro definici parametrů plánování návštěv. Jedná se o stanovení hodiny začátku a konce návštěvního dne a maximální délku návštěvy. Dalším

údajem je časový interval pro plánování návštěv, který stanovuje časové úseky tvořící atomické časové úseky v plánování návštěv, které pak slouží i jako interval pro naplnění výčtových polí pro zadání času začátku a konce návštěv, ale také slouží jako interval pro kontrolu počtu naplánovaných návštěv.



Obrázek 32 - Nastavení aplikace karta Nastavení návštěv

Všechny provedené změny se automaticky ukládají při uzavření formuláře.

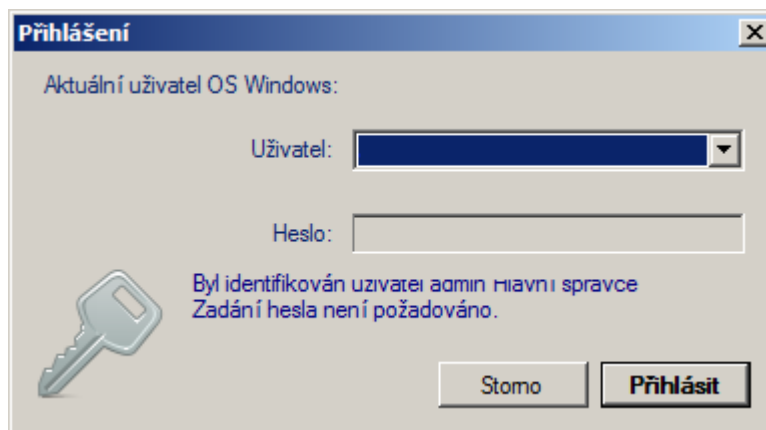
3.2 Základní příručka pro uživatele aplikace

V této části textu diplomové práce bude představena základní funkcionalita popisovaného informačního systému formou sestavené uživatelské příručky. Jedná se zejména o operace spojené s agendou plánování návštěv a evidenci vězňených osob.

3.2.1 Přihlášení uživatele

Pro řádné spuštění aplikace je nutné provést proces identifikace a autorizace uživatele. Tento postup je nutný, neboť je nutností zabezpečit přístup k datům a nástrojům informačního systému jen oprávněným osobám. Z tohoto důvodu je prvním zobrazeným formulářem aplikace dialog pro autentizaci uživatele. Tento formulář obsahuje výběrový seznam, jehož výběrová nabídka obsahuje všechny platné profily uživatelů. Po výběru profilu provede systém zjištění, zda vybraný uživatel je povinen zadat pro svůj vstup heslo. Pokud ano, pak systém zpřístupní pole pro zadání uživatelského hesla, jinak je pole

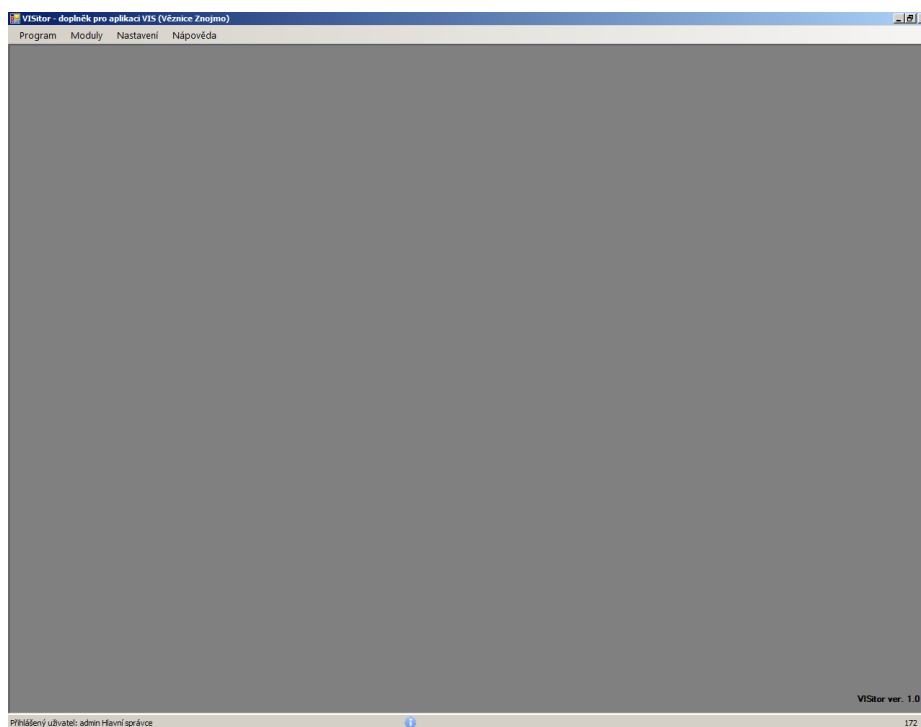
nepřístupné a uživatel se může ihned přihlásit. Pokud uživatel musí zadat heslo, pak před samotným přihlášením je ověřeno, zda bylo zadáno platné heslo. Jestliže bylo zadáno platné heslo, pak je uživatel přihlášen, jinak je nutno zadání hesla opakovat. Počet možných opakování je specifikován v nastavení bezpečnostních zásad aplikace. Pokud dojde k vyčerpání pak je přihlášení odmítnuto a aplikace se uzavírá.



Obrázek 33 – Dialog Přihlášení uživatele

3.2.2 Hlavní okno aplikace

Hlavní okno aplikace tvoří rodičovský kontejner pro všechny ostatní formuláře a dialogová okna, která jsou v průběhu činnosti aplikace zobrazena. Největší část tohoto formulářového okna zaujímá plocha, ve které jsou zobrazovány ostatní formuláře, lišta menu s příkazy pro ovládání funkčních prvků aplikace. Ve spodní části je umístěný stavový panel pro zobrazování některých vlastností spuštěné programy. Jedná se například o jméno aktuálně přihlášeného uživatele, zobrazení ukazatele průběhu déle trvajících operací či zobrazení textu pro nápovědu k ovládacím prvkům.



Obrázek 34 - Hlavní okno aplikace VISitor

3.2.3 Nastavení vlastností uživatele

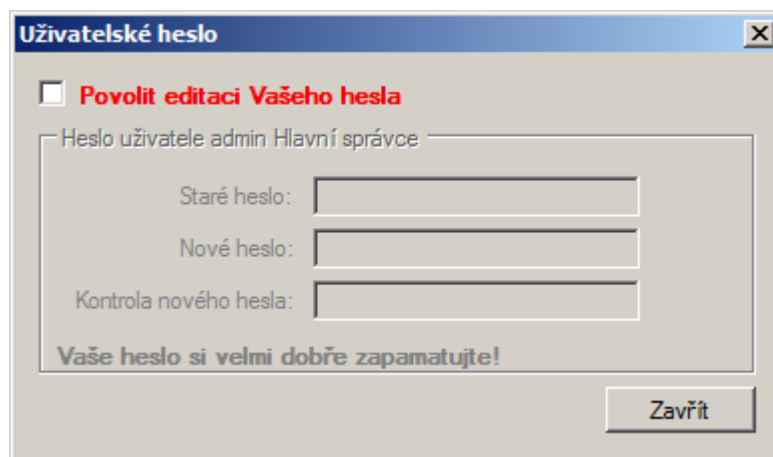
Každý přihlášený uživatel má možnost upravit parametry své osoby uživatele. Pro tyto účely je k dispozici formulář, pomocí kterého je možno upravit některé parametry osoby uživatele. Pokud je uživatel zároveň i zaměstnancem věznice, pak se zobrazí i prvky pro změnu některých parametrů zaměstnance – evidenční kód a typ pracovního zařazení.

Obrázek 35 – Formulář Profil uživatele

3.2.4 Změna uživatelského hesla

Každý uživatel má možnost provést změnu svého uživatelského hesla, kterým autorizuje svoji osobu při přihlašování do aplikace. Tato operace se provádí pomocí dialogu uživatelské heslo, který lze zobrazit pomocí příkazu v nabídce hlavního menu aplikace *Nastavení – Změna uživatelského hesla*. Tento formulář má z bezpečnostních důvodů všechna textová pole ve výchozím stavu nepřístupná. Pro jejich zpřístupnění je nutné zaškrtnout políčko *Povolit editaci hesla*. Pro samotnou změnu hesla je nutné zadat jednak staré heslo, a poté je nutné zadat hodnotu nového hesla a následně zadat toto heslo jednou, jako kontrolu nového hesla, neboť textová pole namísto znaků hesla zobrazují pouze zástupné znaky definované systémem. Při uzavření formuláře proběhne validace vložených

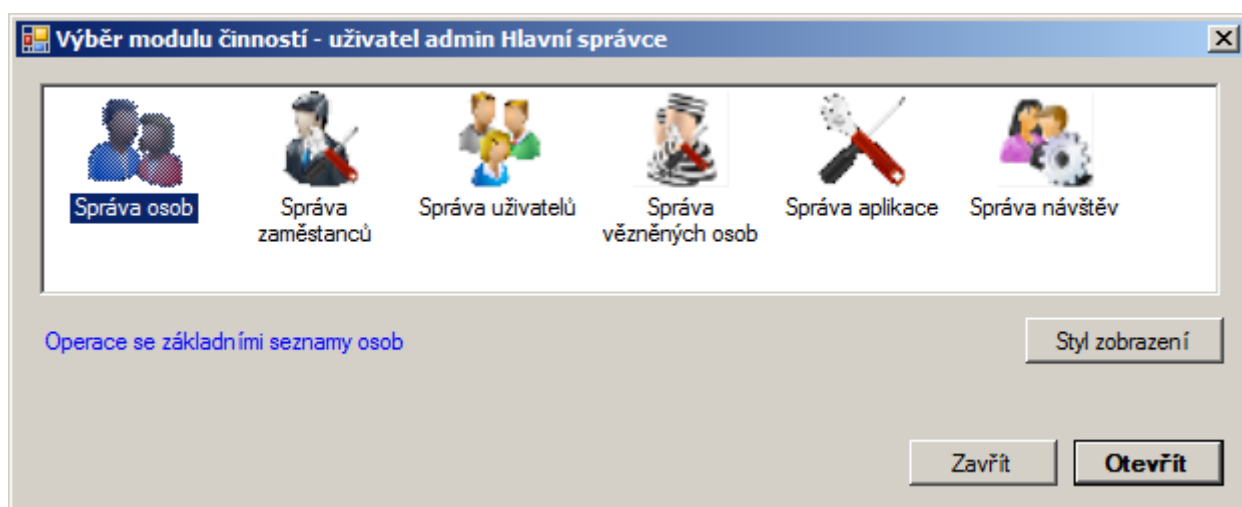
hesel. Pokud je nalezena neshoda, pak je o této skutečnosti uživatel informován pomocí bublinkové nápovědy a označením ovládacího prvku s chybnou hodnotou a varovnou ikonou. Pokud jsou všechny hodnoty v pořádku, dojde k uložení hodnoty do databáze a uživatel je informován o úspěšnosti provedené operace.



Obrázek 36 – Dialog pro změnu uživatelského hesla

3.2.5 Formulář pro výběr modulu činností

Při každém úspěšném spuštění aplikace je uživateli zobrazen výběr všech skupin činností, ke kterým jako přihlášený uživatel má oprávnění. Výběrem některé skupiny a následnou aktivací buď dvojklikem anebo tlačítkem Otevřít, se zobrazí podnabídka jednotlivých operací vybrané skupiny. Výběrem a následnou aktivací položky se spustí k položce přidružená funkcionálníta systému reprezentovaná otevřením příslušného formuláře.



Obrázek 37 – Formulář Výběr modulu činností

3.2.6 Zobrazení nápovědy k programu

Kdykoliv během práce s aplikací VISitor je možné pomocí kláves F1 zobrazit nápovědu umístěnou v externím souboru nápovědy. Nápověda se zobrazí v samostatném okně a bude-li existovat téma odpovídající aktuálnímu formuláři, pak se zobrazí téma nápovědy pro tento formulář, jinak se zobrazí hlavní stránka.

3.3 Správa návštěv

V této části textu se seznámíme s možnostmi správy návštěv vězňených osob. Návštěva je nárok vězňené osoby, který se řídí aktuálně platnými normami², které specifikují podmínky samotné realizace konání, ale také specifikuje administrativní úkony s touto agendou spojené.

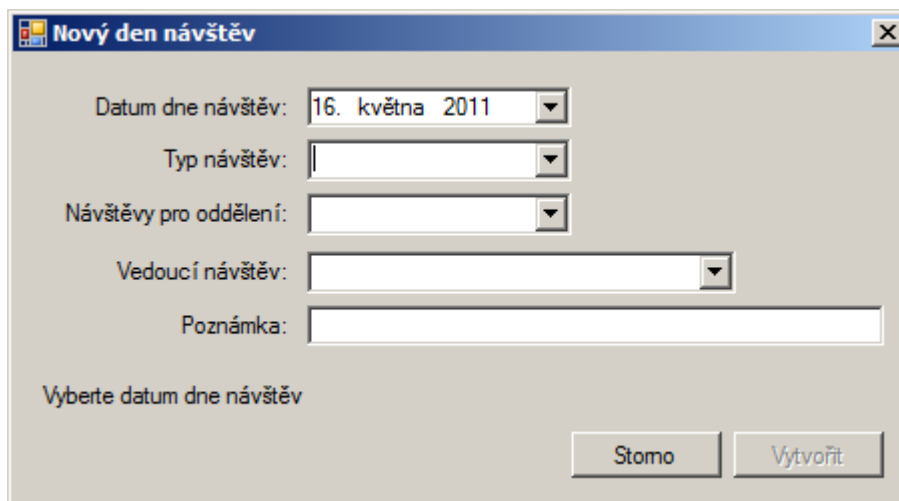
3.3.1 Vytvoření nového návštěvního dne

Realizace jednotlivých návštěv je plánována na konkrétní dny, které jsou nazývány Návštěvní dny. Pokud není v požadovaný den naplánovaný den návštěv, není možné návštěvu na tento den naplánovat. Proto je vhodné před samotným plánováním jednotlivých návštěv, vytvořit plán návštěvních dnů a následně potřebné návštěvní dny.

Pro založení návštěvního dne je aplikace vybavena vlastním formulářem, který slouží k zápisu parametrů návštěvního dne. Hlavním parametrem je datum dne návštěv, což je datum realizace naplánovaných návštěv. Jedinou podmínkou pro přijetí nové hodnoty data návštěvního dne je skutečnost, že na tento den nebyl dříve již naplánovaný jiný návštěvní den. Pokud ano, je uživatel o této skutečnosti informován zprávou zobrazenou v bublinkové nápovědě prvku s datem nové dne návštěv. Další parametry jsou pouze informačního charakteru a při jejich zadání nehrozí, že by docházelo k nějakým konfliktům při následném plánování návštěv. Jedná se o definování preferovaného typu návštěv, oddělení věznice, pro které je tento den určen, poznámky a pracovníka odpovědného za řádný průběh návštěvního dne a samotné realizace naplánovaných návštěv. Pokud jsou všechna požadovaná pole vyplněna, může uživatel tlačítkem Vytvořit, spustit proces

² Zákon 169/1999 Sb., o výkonu trestu odnětí svobody, ve znění pozdějších předpisů
Zákon 193/1993 Sb., o výkonu vazby, ve znění pozdějších předpisů

uložení dat nového dne návštěv. O úspěšnosti uložení bude uživatel informován informační zprávou.



Obrázek 38 – Formulář Nový návštěvní den

3.3.2 Blokace návštěvních dnů

Pro případ, že existuje důvod, proč nelze na specifické datum plánovat nějaké návštěvy, je možné vytvořený návštěvní den označit příznakem blokace, a tím znemožnit naplánování dalších návštěv na tento den. Pokud nastane situace, že jsou na tento den již nějaké návštěvy naplánovány, pak jsou tyto návštěvy stále platné, neboť blokace je platná pouze pro zápis nových návštěv. Tento způsob je možný využít zejména v případech, kdy není možné realizovat návštěvy v určitých dnech z organizačních důvodů.

3.3.3 Vytvoření nové návštěvy

Vytvoření nové návštěvy pomocí formuláře Nová návštěva je sekvencí několika kroků. Prvním krokem je výběr osoby vězně, která požaduje naplánování návštěvy. Osobu vězně lze vybrat pomocí příkazového tlačítka *Vybrat*, které zobrazí modální dialogové okno pro výběr vězněné osoby. Po výběru vězně, se zobrazí základní informace o vybrané osobě vězně.

Dalším krokem je specifikování dne návštěv. Pokud požadovaný den návštěv byl dříve vytvořen, pak se je součástí nabídky výběrového prvku, pokud v seznamu schází, je možné jej vytvořit pomocí příkazového tlačítka *Nový*, které spustí formulář pro vytvoření nového dne návštěv. Pokud je nový den v tomto formuláři vytvořen, pak se nastaví jako vybraný den návštěv pro naplánování nové návštěvy.

Proces vytvoření nové návštěvy pokračuje dalším krokem, a to je specifikování času zahájení návštěvy a její délky. Výběrem požadovaných hodnot z nabídek ovládacích prvků nastavíme časové parametry návštěvy. Hodnota konce návštěvy se automaticky dopočítává a je zde zobrazena jen jako pomocná informace. Ve stejné skupině ovládacích prvků jsou prvky pro určení návštěvní místnosti pro konání návštěvy a označení možnosti přijetí balíčky v rámci realizace návštěvy.

Nová návštěva

Osoba vězně

Jméno a příjmení: **Hele Jan** Vybrat

Evidenční kód: **asasde**

Umístění: **B / 224**

Typ vězně: **odsouzen - trest dozor**

Návštěvní den

Vybraný návštěvní den: **9.6.2011 Návštěvy A** Nový

Návštěvy (1,5 hodiny) Návštěvy A, vedoucí: psycholog Mgr. Cikvličná Nad'a

Návštěva

Čas zahájení: Délka: minut Čas ukončení:

Návštěvní místnost: Povoleno přijetí balíčku

Pozvané osoby

Na návštěvu je nutno pozvat alespoň jednu osobu, maximálně však 4 osoby.

manžel/manželka Heleová Jana, Praha, Radlická 16 Nová

Žádná osoba není vybraná.

Informace Storno Tisk Uložit

Obrázek 39 – Formulář Nová návštěva

Nyní je další postup podmíněn typem vybraného vězně, neboť zákonnými normami je specifikováno, u kterých typů vězněných osob je povinnost definovat osoby, které budou na návštěvu pozvány. Pokud je to nutné, pak se tyto vybírají z osob, které mají v systému definovaný vztah s osobou vězně a mají k e své osobě přiřazenu alespoň jednu adresu. Maximální počet pozvaných osob je opět limitován zákonnou normou. Pokud není požadovaná osoba zobrazena v seznamu osob k pozvání, pak je nutné do seznamu přidat novou osobu pomocí příkazového tlačítka *Nová*, které zobrazí dialog pro definici vztahu

osoby s osobou vězně. Pokud je třeba a je vybrán potřebný počet osob k pozvání, pak je možné uložit novou návštěvu pomocí tlačítka *Uložit* nebo pomocí tlačítka *Tisk* se provede sdružená operace uložení a generování tiskové sestavy. Tím je proces naplánování nové návštěvy dokončen.

3.3.4 Definice osoby ve vztahu s vězněnou osobou

Pokud je pro potřeby vytvoření nové návštěvy zapotřebí definovat nový vztah mezi osobou vězně a další osobou, pak je možné provést tuto operaci pomocí formuláře *Definice vztahu mezi osobami*. Pokud je tento formulář zobrazen voláním z jiného formuláře, pak je automaticky nastavena osoba vězně, případně je možné osobu vězně nastavit pomocí tlačítka *Vybrat*, které zobrazí dialog pro výběr vězněné osoby.

Dalším krokem je stanovení druhé osoby v popisovaném vztahu. Osobu lze určit buď pomocí tlačítka *Vybrat* výběrem z osob uložených v systému, nebo lze vytvořit novou osobu tlačítkem *Nový*.

Definice vztahu mezi osobami

Osoba vězně

Jméno a příjmení: Hele Jan Vybrat

Evidenční kód: asasde

Umístění: B / 224

Typ vězně: odsouzen - trest dozor

Osoba v definovaném vztahu s osobou Hele Jan

jméno: - datum narození: - Vybrat

příjmení: - pohlaví: - Nová

Adresa osoby ve vztahu s osobou vězně

Druh adresy: [adresa není zadána]

Ulice: Číslo: Nová

Město: PSČ:

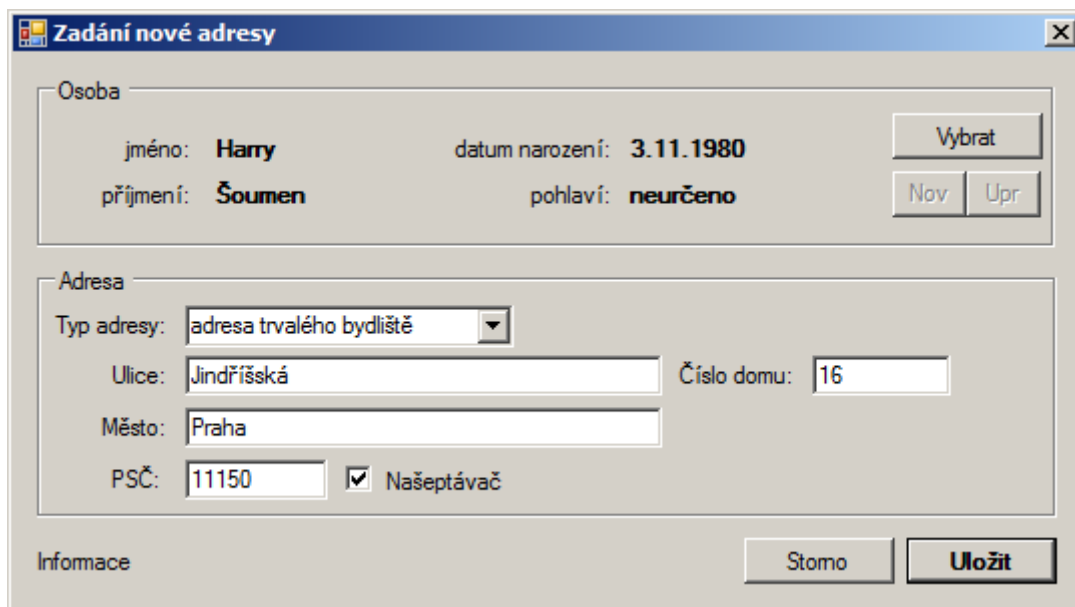
Specifikace vztahu mezi osobami

Označení vztahu: [vzdálenost]

Informace Storno Uložit

Obrázek 40 – Formulář Definice vztahu mezi osobami

Po definici druhé osoby, je možné této osobě pomocí tlačítka *Nová* v kontejneru *Adresa osoby* přiřadit libovolný typ adresy. Po aktivaci tlačítka se zobrazí dialogové okno s možnostmi vytvoření nové adresy a určení druhu této adresy. Formulář obsahuje informace o osobě, které se nová adresa přiřadí a pole pro zadání parametrů nové adresy. Příkazem *Uložit* se nová adresa uloží a přiřadí se jako adresa s definovaného typu vybrané osobě.



The image shows a Windows-style dialog box titled "Zadání nové adresy". It is divided into two main sections: "Osoba" and "Adresa".

Osoba section:

- jméno: Harry
- příjmení: Šoumen
- datum narození: 3.11.1980
- pohlaví: neurčeno

Buttons: "Vybrat", "Nov", "Upr".

Adresa section:

- Typ adresy: adresa trvalého bydliště (dropdown menu)
- Ulice: Jindřišská
- Číslo domu: 16
- Město: Praha
- PSC: 11150
- Našeptávač

Buttons: "Storno", "Uložit".

Obrázek 41 – Formulář Zadání nové adresy

Posledním parametrem definice vzájemného vztahu mezi osobou vězně a vybranou osobou je definovat vzájemný vztah výběrem odpovídající položky z výběrového prvku. Pokud jsou všechna požadovaná pole vyplněna, je možné uložit definici vztahu osob tlačítkem *Uložit*. Aktivace tohoto tlačítka spustí proces validace, které provede kontrolu zadaných dat. Pokud je nalezena nějaká chyba, pak je označena bublinkovou nápovědou a je označena výstražnou ikonou, jinak se definice uloží a formulář uzavře.

3.3.5 Změna stavu a zrušení naplánované návštěvy

V případě, že je nutné zrušit nebo změnit stav naplánované návštěvy, nejprve se musí najít její výskyt v seznamu návštěvních dnů. Následně pomocí lokální nabídky nebo dvojklikem zobrazíme formulář s detailem naplánované návštěvy. V tomto formuláři je mimo jiné možnost tisku sestavy povolenky k návštěvě, ale také změnit stav návštěvy. Změna stavu se provádí kliknutím na tlačítko *Změna stavu* a následným výběrem nové hodnoty požadovaného stavu návštěvy. Pokud z nabídky vybereme položku *zrušená*, nastaví se návštěva jako zrušená.

Detail návštěvy (1545502086)

Osoba vězně

Jméno a příjmení: **Kačenka Miroslav**

Evidenční kód: **kn7844**

Umístění: **A / 101**

Typ vězně: **obviněn - vazba C**

Návštěva

Datum: **31.5.2011** Den návštěv: **31.5.2011 Návštěvy C**

Začátek: **10:00** Konec: **12:30** Délka: **150**

Balík: **ANO** Místnost: **malá**

Stav: **naplánována** Navštívenka: **zatím netisknuta**

Žádné pozvané osoby

Obrázek 42 – Formulář Detail návštěvy

Alternativou k tomuto postupu je nalezení návštěvy v seznamu návštěv návštěvního dne, v seznamu vybrat požadovanou návštěvu a kliknutí pravého tlačítka zobrazit lokální nabídku, kde zvolením příkazu *Zrušit* provedeme zrušení naplánované návštěvy.

3.3.6 Tisk seznamu naplánovaných návštěv

Tisk sestavy se seznamem naplánovaných návštěv na návštěvní den je možné pomocí formuláře *Návštěvní dny*, který lze zobrazit pomocí nabídky menu hlavního okna aplikace *Moduly – Správa návštěv – Procházet návštěvní dny*. V tomto formuláři pak vyhledat požadovaný návštěvní den a pomocí příkazového tlačítka *Tisk seznamu* se zobrazí sestava seznamu naplánovaných návštěv na vybraný den, kterou je pak možno i vytisknout na zvolené tiskárně.

3.3.7 Zobrazení naplánovaných návštěv

Přehled naplánovaných návštěv je možné získat pomocí formuláře *Návštěvní dny*, který je možno zobrazit pomocí příkazu *Procházet návštěvy* nabídce menu hlavního okna *Moduly –*

Správa návštěv nebo je možné jej zobrazit pomocí příkazu v dialogovém okna pro výběr modulu činností. Pomocí tohoto formuláře je možné zobrazit sestavu se seznamem naplánovaných návštěv, zadat novou návštěvu, vytvořit nový návštěvní den.

Dále je možné pomocí tohoto formulářového okna označit den návštěv jako blokový. Nastavení blokace návštěvního dne se provádí pomocí příkazu lokálního menu zobrazeného nad položkou ve stromovém prohlížeči představující Návštěvní den. Blokové dny jsou pak zvýrazněny jinou barvou.

Obrázek 43 – Formulář Návštěvní dny

3.4 Správa vězňů

V této části textu se seznámíme s možnostmi základní administrativy vězňů, která pro potřeby této aplikace spočívá v udržování základních osobních informací vězňů a informace o jejich aktuálním umístění v hierarchii ubytovacích prostor věznice.

3.4.1 Vložení nové osoby jako vězně

Proces založení nového vězně spočívá v operaci přiřazení osobě role vězeň. Pro tuto činnost je přizpůsoben formulář *Vytvoření nového vězně*, který obsahuje prvky pro provedení takového přiřazení. Jako první je prvek pro specifikování osoby vězně. Tuto

osobu lze buď *Vybrat* z osob, které jsou již v systému VISitor dříve evidovány, nebo lze pomocí tlačítka *Nová* vytvořit novou osobu. Pokud je odpovídající osoba zvolena, pak je nutné definovat typ vězněné osoby pomocí výběru odpovídajících položek ve výběrových seznamech v poli Typ věznění.

The screenshot shows a software window titled "Vytvoření nového vězně". It contains three main sections of input fields:

- Osoba vězně:** Fields for "jméno:", "příjmení:", "datum narození:", and "pohlaví:". There are "Vybrat" and "Nová" buttons to the right.
- Typ věznění:** Two dropdown menus for "typ vězně:" and "podskupina:".
- Umístění:** Two dropdown menus for "oddělení:" and "ložnice/cela:", and a text input field for "VIS kód:".

At the bottom of the window, there is a label "Připraven." and two buttons: "Storno" and "Uložit".

Obrázek 44 – Formulář Vytvoření nového vězně

Další položkou je definice umístění ve věznici, pomocí výběru položky z výběrového seznamu. Tím se osoba zařadí na ložnici/celu a oddělení. Další položkou je pole pro zadání evidenčního VIS kódu pro jednoznačnou identifikaci vězněné osoby.

Uložení nové osoby vězně se provede pomocí tlačítka *Uložit*, které nejprve provede validaci dat, která v případě zjištění chyby proces ukládání dat přeruší a označí ovládací prvek s chybnou hodnotou. Pokud je vše v pořádku, pak jsou data nové osoby vězně uložena.

3.4.2 Zobrazení informací o vězněné osobě

Pomocí formuláře *Průzkumník věznice*, který zobrazuje hierarchii systému oddělení, cel a ložnic, je možné zobrazit seznam vězněných osob umístěných ve věznici. Vyvolání lokální nabídky nad položkou představující vězněnou osobu a zvolením příkazu *Zobrazit informace* nebo provedením dvojkliku na požadovanou osobou, zobrazí formulářové okno *Detail vězně*, v kterém jsou zobrazeny podrobné informace o vězněné osobě.

Detail vězně

Vybraná osoba vězně

Jméno a příjmení: **Kačenka Miroslav** Vybrat

Evidenční kód: **kn7844**

Umístění: **A / 101**

Typ vězně: **obviněn - vazba C**

Kačenka Miroslav - přímý pohled

Typ fotografie: přímý pohled

Hlavní fotografie

Nová Nahradit Odstranit

Informace o osobě | Informace o aktuální věznění | Fotografie | Návštěvy | Příbuzné osoby

Jméno osoby

jméno: Miroslav + jméno

příjmení: Kačenka + příjmení

Datum narození

Datum narození: 16. listopadu 1986

datum narození není známo

Vzdělání

Titul před jménem:

Titul za jménem:

Nový vězeň Informace Zavřít

Obrázek 45 – Formulář Detail vězně

Formulář Detail vězně zobrazuje všechny informace, které jsou pro vězněné osoby v aplikaci VISitor dostupné. Ovládací prvky jsou tematicky rozděleny pomocí karet do skupin, aby bylo jejich zobrazení čitelné a přehledné.

3.5 Správa ubytování vězněných osob

Pro potřeby informačního systému VISitor je nutné evidovat u vězněných osob aktuální umístění, aby bylo možno přesně určit příslušnost k oddělení věznice a s tím spojené návštěvní dny.

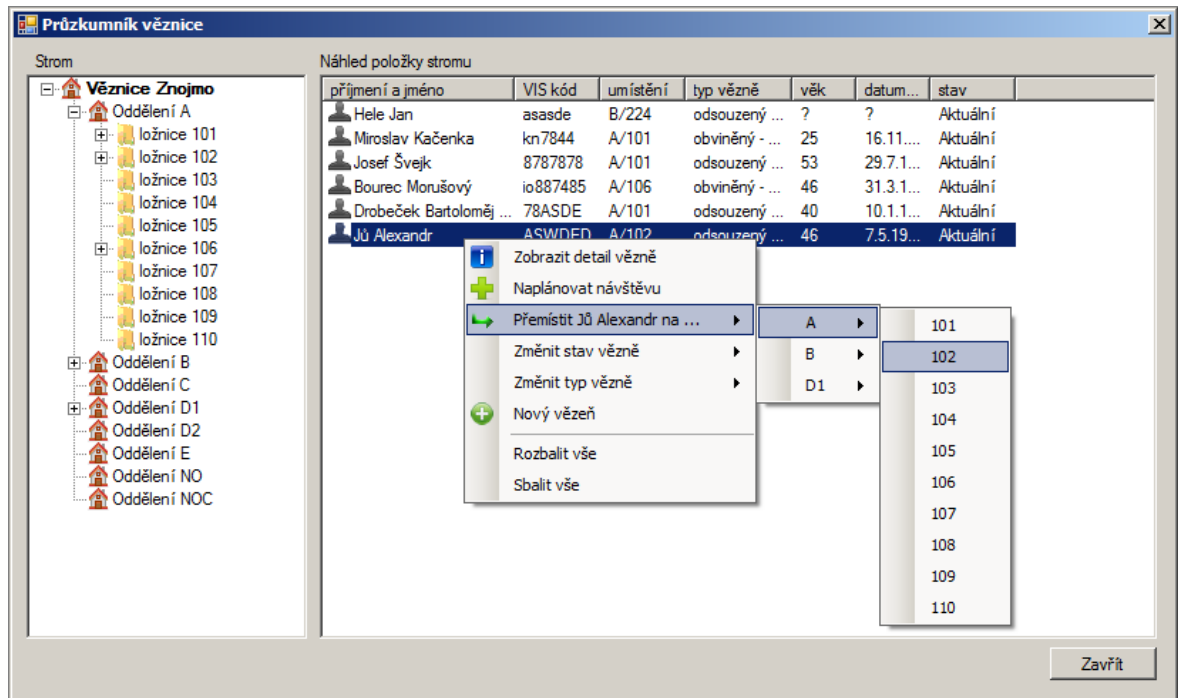
3.5.1 Změna umístění vězně

Základním prvkem pro správu ubytování vězněných osob pomocí aplikace VISitor je formulář nazvaný Průzkumník věznice. Pomocí toho ovládacího prvku je možno jednak zobrazit aktuální stav umístění osob, ale pomocí implementovaných nástrojů snadno změnit umístění vězněných osob.

Prvním způsobem pro změnu místa ubytování je využití lokální nabídky vybrané osoby v seznamu vězněných osob. Pomocí podpoložek nabídky *Přemístit* lze volbou položky v menu určit cílovou ložnici/celu, na kterou chceme provést přemístění vězněné osoby.

Druhou možností je využití technologie *táhni a pusť*³, kdy prostým přetažením osoby na cílovou místnost ve stromovém prohlížeči provedeme samotné přemístění vězněné osoby.

Ať zvolíme jakýkoliv způsob, vždy je nutné pro samotné dokončení operace přemístění vězněné osob potvrdit své stanovisko v zobrazeném dialogu.



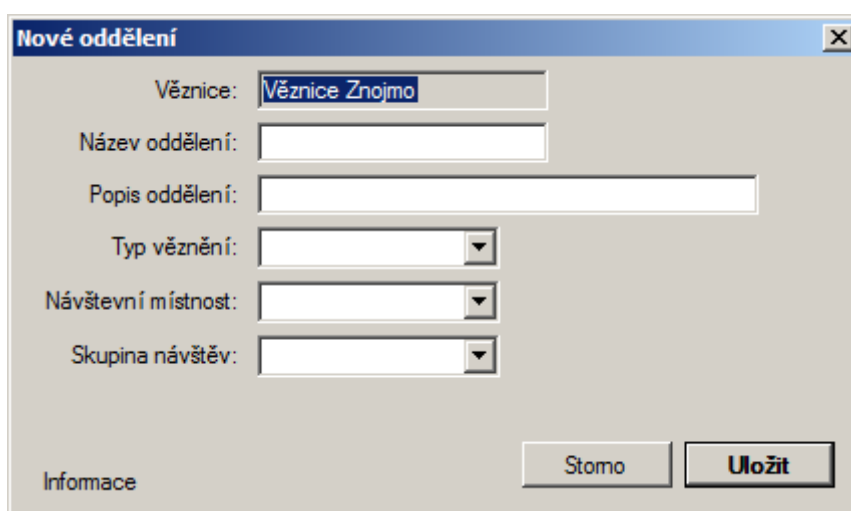
Obrázek 46 – Formulář Průzkumník věznic – změna umístění vězně

³ Překlad anglického výrazu Drag&Drop

3.5.2 Přidání nového oddělení

Pomocí lokální nabídky nad položkou představující věznicí ve stromovém prohlížeči formuláře *Průzkumník věznic* je možné pomocí příkazu *Nové oddělení* zobrazit dialog *Nové oddělení*, pomocí kterého lze definovat a vytvořit nové oddělení věznic. Parametry nového oddělení jsou jeho název, popis oddělení, specifikace typu vězňených osob, pro které je oddělení určeno, výchozí návštěvní místnost a přiřazení do návštěvní skupiny.

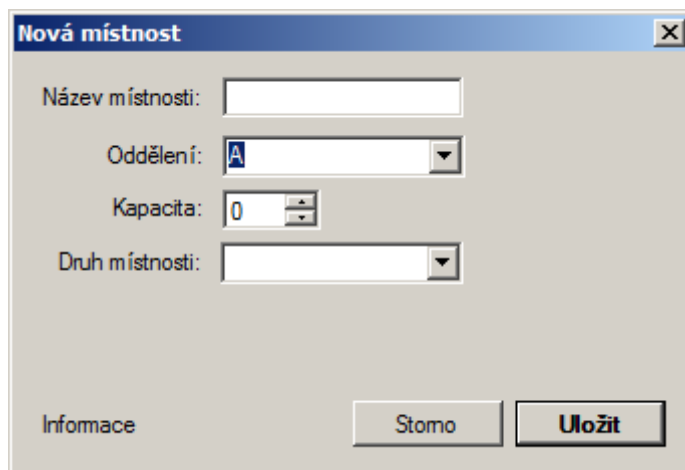
Tlačítkem Uložit se provede validace dat, a v případě, že data jsou v pořádku, jsou následně uložena a nové oddělení je zdárně vytvořeno. V případě, že byl nalezen problém v zadaných datech, je ukládání přerušeno a ovládací prvek s chybnou hodnotou je zvýrazněn.



Obrázek 47 – Formulář Nové oddělení

3.5.3 Předání nové ložnice-cely

Podobnou operací jako je vytvoření nového oddělení věznic, je vytvoření nové cely/ložnice. Operace se opět provádí ve formuláři *Průzkumník věznic*, kdy se vyvolá lokální nabídka nad položkou představující oddělení, ve které má být nová místnost zařazena. V nabídce je dostupný příkaz *Nová místnost na oddělení*, který zobrazí dialogové okno pro zadání parametrů nové místnosti – název, spádové oddělení, kapacita a specifikace druhu místnosti. Uložení nové místnosti se provede tlačítkem Uložit, které nejprve spustí validaci dat a v případě, že nebyla nalezena chyba, je nová místnost vytvořena, jinak je uživatel upozorněn na nedostatky zvýrazněním prvku s chybnou hodnotou.



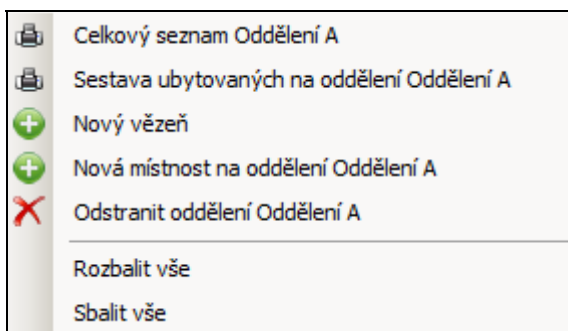
Obrázek 48 – Formulář Nová místnost

3.5.4 Změna názvu oddělení, změna názvu místnosti

Změna názvu místnosti nebo oddělení se provádí editací příslušné položky ve stromovém prohlížeči formuláře Průzkumník věznice. Nejprve je nutné položku kliknutím vybrat a následným dalším kliknutím se aktivuje editační režim položky. Ukončení editace se provede stiskem klávesy Enter.

3.5.5 Sestava ubytování oddělení

Pomoc lokální nabídky položek představující oddělení věznice je možné pomocí příkazů *Celkový seznam* a *Sestava ubytovaných* zobrazit náhled sestavy představující v prvním případě abecední seznam vězněných osob zařazených na vybrané oddělení, a ve druhém případě se jedná o zobrazení sestavu vyjadřující obsazení jednotlivých místností spadající do vybraného oddělení.



Obrázek 49 - Lokální nabídka menu pro oddělení věznice

ZÁVĚR

Popisovaná aplikace VISitor vznikla na základě několikaleté zkušenosti s činností v rámci agendy ve vězeňském prostředí, oddělení výkonu trestu a výkonu vazby. Před téměř desítkou roků začaly mé první pokusy o zjednodušení administrativních úkonů zaměstnanců vězeňské služby pomocí výpočetní techniky, neboť v té době „kamenné“ se využívalo systému kolujících sešitkových agend, kdy se posouvaly papírové doklady z jedné kanceláře do druhé rychlostí blízké nule, a tak se nezdálo, že se vyskytl problém při rezervaci termínu na provádění návštěv. Nejednou vznikl problém s překročením kapacity návštěvní místnosti díky vzájemné nedohodě jednotlivých spolupracovníků. To bylo pro mne prvním impulsem vytvořit systém, který by nějakým způsobem dokázal eliminovat takové problémy.

Proto jsem se začal formou samostudia učit tvorbě informačních systémů. Začaly vznikat první projekty databázových aplikací stvořených pomocí prostředí *Microsoft Access 97*, na kterých jsem postupně zdokonaloval své programátorské myšlení a návyky. Během několika let se stal jeden z mých úspěšných projektů, vytvořených právě pomocí jádra *Microsoft Access* téměř neodmyslitelnou součástí či pracovní pomůckou zaměstnanců Věznice Znojmo.

Na takto získaných základech jsem se rozhodl použít své znalosti prostředí a nabyté zkušenosti se správou informačního systému v prostředí vězeňské agendy pro vytvoření další verze aplikace, která by splnila kritéria novějšího systému, než jsou ty dosavadní.

Je nutno také poukázat na fakt, že před několika lety byl uveden do provozu oficiální informační systém VIS⁴, který je již od základu navržen jako jednotný systém pro všechny organizační jednotky v celé Vězeňské službě. Dle mého názoru přípravné řízení, sběr a analýza požadavků vyvíjeného systému nebyla provedena k prospěchu koncových uživatelů, a tak se od počátku ostrého nasazení uživatelé potýkají s absencí některých nástrojů, které by takový nový systém mohl a měl mít. A nyní je již vývoj softwaru VIS definitivně ukončen, tudíž jakékoliv změny ve funkcionalitě na podněty uživatelů jsou vyloučené.

⁴ Vězeňský Informační Systém – zakázkový software pro VSČR zpracovaný pod záštitou firmy Microsoft

Proto jednou z prvotních myšlenek při vzniku systému VISitor byla snaha o to, aby se tato aplikace stala jakýmsi doplňkem oficiálního informačního systému VIS, obohacovala jej o aplikační prostředí, které by bylo propracované podle skutečných požadavků uživatelů, a poskytovala by některé chybějící prvky funkcionality hlavního oficiálního informačního systému. Bohužel k žádné takové spolupráci nedošlo, neboť systém VIS není navržen pro sdílení informací a je tak nepřístupný externím aplikacím. Z tohoto důvodu je nemožné jej využívat jako oficiální databázi, zdroj dat pro jiné aplikace, a proto jsem musel pro svou aplikaci použít vlastní řešení ukládání dat a vlastní systém evidence vězněných osob, zaměstnanců a samotných návštěv.

V současné době je stav takový, že zaměstnanci používají dva informační systémy. Jeden oficiální Vězeňský informační systém VIS, který je nutno používat pro zachování integrity informací v rámci celého sektoru Vězeňské služby České republiky a druhý neoficiální informační systém, vytvořený mou osobou na základě skutečných potřeb uživatelů, systém který poskytuje potřebnou funkcionality prospívající kvalitě jejich práce.

CONCLUSION

The described application of VISitor was based on years of experience in the activities on the agenda in the prison environment department of the sentence and remand. Nearly ten years have started my first attempts to simplify the paperwork prison staff using computer technology, because at the time of the "stone" is used the system circulating book agenda, the paper is shifted delays from one office to another rate close to zero, so often happened that a problem occurred during the booking date of visits. So often have a problem with congestion visiting room through mutual associates of disagreement. That was my first impulse to create a system that would somehow managed to eliminate such problems.

So I began to learn through self-creation of information systems. The first projects were created database applications created by using the Microsoft Access 97, on which I have gradually improved his programming and thinking habits. During a few years became one of my successful projects, created just using the core Microsoft Access, or almost an integral working tool Prison Znojmo's staff.

On the basis thus obtained, I decided to use their knowledge of the environment and experiences in the management information system in the penitentiary agenda for a further version, which would meet the other criteria newer system than the existing ones.

It should also highlight the fact that a few years ago was put into official operation of an information system VIS, which has been designed from the ground up as a single system for all organizational units across the Prison Service. In my opinion, pre-trial proceedings, collection and analysis requirements of the developed system was implemented for the benefit of end users, and so sharp from the start deployment, users faced with the absence of certain instruments, that new system could and should have. And now there is no software development VIS finally ended, so any changes in functionality to input the user are excluded.

Therefore, one of the primary ideas of the VISitor system was that this application has become a kind of complement to the official information of the VIS, which so enrich the application environment, which would elaborate the actual needs of users, and thus would complement some missing functionality of the main elements of official information system. Unfortunately there was no cooperation, as the VIS system is designed for sharing, and is thus inaccessible to external applications. For this reason, it is impossible to use the database as the official source of data for other applications, so I had to for my application

to use custom storage solutions and its own system of accounting for prisoners, staff and agenda of visits.

SEZNAM POUŽITÉ LITERATURY

- [1] KANISOVÁ, Hana; MÜLLER, Miroslav. *UML srozumitelně*. 2004. Brno : Computer Press, 2004. 155 s. ISBN 80-251-0231-9
- [2] PETROUTSOS, Evangelos. *Myslíme v jazyku Visual basic .NET : 1. díl*. Praha : Grada Publishing a.s., 2003. 676 s. ISBN 80-247-0371-8
- [3] PETROUTSOS, Evangelos. *Myslíme v jazyku Visual basic .NET : 2. díl*. Praha : Grada Publishing a.s., 2003. 540 s. ISBN 80-247-0372-8.
- [4] VIESCAS, John; CONRAD, Jeff. *Mistrovství v Microsoft Office Access 2007*. Brno : Computer Press, 2008. 1224 s. ISBN 978-80-251-2162-7.
- [5] DVOŘÁK, Miloš. *Objektová analýza, návrh a programování* [online]. 2003 [cit. 2011-05-16]. Návrhové vzory (Design Paterns). Dostupné z WWW: <http://objekty.vse.cz/Objekty/Vzory>
- [6] FORD, Sara. *266 tipů a triků pro Microsoft Visual Studio*. Brno : Computer Press, 2009. 224 s. ISBN 978-80-251-2554-0
- [7] LACKO, Luboslav. *SQL - Hotová řešení*. Brno : Computer Press, 2003. 296 s. ISBN 80-7226-975-5.
- [8] HANÁK, Ján. *Visual Basic 2010 – Hotové riešenia* [online]. Praha : Microsoft, 2010 [cit. 2011-05-16]. Dostupné z WWW: http://download.microsoft.com/download/2/7/9/27983464-6245-47C3-A2A9-FC57E05581A4/Visual_Basic_2010_riesenia.pdf.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

VISitor	Název popisované aplikace.
VIS	Název vězeňského informačního systému
Drag&Drop	Anglický název pro techniku „táhni a pusť“

SEZNAM OBRÁZKŮ

Obrázek 1 – Proces realizace požadavku naplánování návštěvy	19
Obrázek 2 – Proces přihlášení uživatele do systému	20
Obrázek 3 – Základní Use case diagram aplikace	22
Obrázek 4 – Use Case diagram Plánování návštěv	23
Obrázek 5 – Use Case diagram případu užití Zadání nové osoby ve vztahu.....	26
Obrázek 6 – Use Case diagram pro Vytvoření nového uživatele	29
Obrázek 7 – Use case diagram pro Zrušení naplánované návštěvy	31
Obrázek 8 - Sekvenční diagram Naplánování nové návštěvy	33
Obrázek 9 – Sekvenční diagram Zrušení naplánované návštěvy	34
Obrázek 10 – Závislosti seskupení tříd	35
Obrázek 11 – Diagram třídy seskupení Uživatel aplikace.....	36
Obrázek 12 – Diagram třídy seskupení Vězeň	37
Obrázek 13 – Diagram třídy seskupení Návštěvní den.....	38
Obrázek 14 – Diagram třídy seskupení Návštěva vězněné osoby	39
Obrázek 15 - Diagram třídy seskupení Pozvané osoby	40
Obrázek 16 – Data diagram pro uložení objektu uživatele aplikace	42
Obrázek 17 – Data diagram pro uložení dat definice vězněné osoby	43
Obrázek 18 – Data diagram pro uložení struktury jména osoby	44
Obrázek 19 – Data diagram pro přiřazení adresy osobě	45
Obrázek 20 – Data diagram pro Plánování návštěv	46
Obrázek 21 – Data diagram pro uložení historie ubytování vězně.....	47
Obrázek 22 – Data diagram pro uložení definice vzájemného vztahu osob.....	48
Obrázek 23 – Formulář pro nastavení zdrojové databáze.....	51
Obrázek 24 – Prvotní přihlášení do aplikace	52
Obrázek 25 – Formulář pro přehled zaměstnanců	53
Obrázek 26 – Formulář Detail zaměstnance	54
Obrázek 27 – Dialog pro vložení nového zaměstnance.....	55
Obrázek 28 – Zobrazení chyby validace.....	55
Obrázek 29 – Formulář pro editaci uživatelských účtů	56
Obrázek 30 – Formulář Nastavení aplikace.....	60
Obrázek 31 – Nastavení aplikace karta Zabezpečení aplikace	61
Obrázek 32 - Nastavení aplikace karta Nastavení návštěv	62

Obrázek 33 – Dialog Přihlášení uživatele.....	63
Obrázek 34 - Hlavní okno aplikace VISitor	64
Obrázek 35 – Formulář Profil uživatele	65
Obrázek 36 – Dialog pro změnu uživatelského hesla.....	66
Obrázek 37 – Formulář Výběr modulu činností	66
Obrázek 38 – Formulář Nový návštěvní den	68
Obrázek 39 – Formulář Nová návštěva	69
Obrázek 40 – Formulář Definice vztahu mezi osobami	70
Obrázek 41 – Formulář Zadání nové adresy.....	71
Obrázek 42 – Formulář Detail návštěvy	72
Obrázek 43 – Formulář Návštěvní dny.....	73
Obrázek 44 – Formulář Vytvoření nového vězně.....	74
Obrázek 45 – Formulář Detail vězně.....	75
Obrázek 46 – Formulář Průzkumník věznice – změna umístění vězně.....	76
Obrázek 47 – Formulář Nové oddělení.....	77
Obrázek 48 – Formulář Nová místnost.....	78
Obrázek 49 - Lokální nabídka menu pro oddělení věznice	78

SEZNAM TABULEK

Tabulka 1 - Funkční požadavky.....	18
Tabulka 2 – Nefunkční požadavky	18
Tabulka 3 – Hlavní scénář Naplánování nové návštěvy.....	25
Tabulka 4 – Alternativní scénář ALT101 Naplánování nové návštěvy.....	25
Tabulka 5 - Alternativní scénář ALT102 Naplánování nové návštěvy	26
Tabulka 6 - Zadání nové osoby ve vztahu s osobou vězně.....	28
Tabulka 7- Alternativní scénář ALT201 definice osob ve vzájemném vztahu	28
Tabulka 8 – Hlavní scénář pro Vytvoření nového uživatele	30
Tabulka 9 – Hlavní scénář Zrušení naplánované návštěvy.....	32
Tabulka 10 – Výčet uživatelských rolí	59