

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO  
KATEDRA INFORMATIKY

## BAKALÁŘSKÁ PRÁCE

Informační systém pro Centrum zdraví



2010

Michal Gottwald

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně.

7. srpna 2010

Michal Gottwald

## **Anotace**

*Tato bakalářská práce popisuje řešení SW produktu pro podporu běžných administrativních činností, důležitých pro fungování soukromého subjektu poskytujícího služby v oblasti zdravého životního stylu a volnočasových aktivit.*

Děkuji panu Mgr. Martinovi Dostálovi, Ph.D. za vedení této bakalářské práce.  
Dále pak své rodině za podporu při studiu.

# Obsah

<b>1. Úvod</b>	<b>1</b>
<b>2. Zadání bakalářské práce</b>	<b>2</b>
2.1. Specifikace zadání . . . . .	2
2.2. Klíčové funkce aplikace . . . . .	2
<b>3. Uživatelská dokumentace</b>	<b>3</b>
3.1. Systémové požadavky . . . . .	3
3.2. Instalace aplikace . . . . .	3
3.3. Popis aplikace . . . . .	8
3.3.1. Spuštění aplikace . . . . .	8
3.3.2. Přihlášení do aplikace . . . . .	8
3.3.3. Hlavní okno aplikace . . . . .	8
3.3.4. Registr klientů . . . . .	10
3.3.5. Denní rozpis . . . . .	16
3.3.6. Změna hesla . . . . .	17
3.3.7. Tisk ceníku . . . . .	18
3.3.8. Evidence zaměstnanců . . . . .	18
3.3.9. Evidence služeb/výkonů . . . . .	19
3.3.10. Evidence majetku . . . . .	21
3.3.11. Evidence klientů . . . . .	22
3.3.12. Šablony jídelníčků . . . . .	23
3.3.13. Součtová sestava . . . . .	24
3.4. Odinstalování aplikace . . . . .	24
<b>4. Programátorská část</b>	<b>26</b>
4.1. Použité technologie . . . . .	26
4.2. Základní use case diagram . . . . .	26
4.3. Analytický model tříd . . . . .	27
4.4. Architektura aplikace . . . . .	27
4.5. Datová vrstva (Data Access Layer) . . . . .	27
4.6. Logická vrstva (Business Logic Layer) . . . . .	28
4.7. Prezentační vrstva (Presentation Layer) . . . . .	33
4.8. Databáze . . . . .	35
4.9. Tiskové výstupy . . . . .	35
4.10. Instalátor . . . . .	35
<b>Závěr</b>	<b>37</b>
<b>Reference</b>	<b>38</b>
<b>A. Popis obsahu příloženého CD</b>	<b>39</b>

## Seznam obrázků

1.	Hlavní formulář aplikace . . . . .	8
2.	Registr klientů - Seznam klientů . . . . .	10
3.	Registr klientů - Detail klienta . . . . .	11
4.	Registr klientů - záložka Rozpis cvičebního plánu . . . . .	12
5.	Plánovací kalendář . . . . .	14
6.	Registr klientů - záložka Jídelníčky . . . . .	15
7.	Návrh jídelníčku . . . . .	16
8.	Denní rozpis . . . . .	17
9.	Změna hesla . . . . .	17
10.	Údaje o zaměstnanci . . . . .	19
11.	Údaje o službě/výkonu . . . . .	20
12.	Majetková karta . . . . .	22
13.	Údaje o klientovi . . . . .	23
14.	Návrh šablony jídelníčku . . . . .	24
15.	Základní use case diagram . . . . .	26
16.	Analytický model tříd . . . . .	27
17.	Datový model . . . . .	36

# 1. Úvod

Každá organizace musí vedle své hlavní činnosti vykonávat mnoho dalších administrativních úkonů. Velké organizace používají podnikové informační systémy, jiné používají systémy vyvinuté na zakázku ale některé menší používají ještě papírovou formu. Pro jednu takovou organizaci je určena aplikace popisovaná v rámci této bakalářské práce.

## **2. Zadání bakalářské práce**

### **2.1. Specifikace zadání**

Vytvořit aplikaci pro podporu běžných administrativních činností, důležitých pro fungování soukromého subjektu poskytujícího služby v oblasti zdravého životního stylu a volnočasových aktivit.

### **2.2. Klíčové funkce aplikace**

- Evidence klientů a jejich cvičebních plánů
- Možnost sestavit cvičební plán pro klienta
- Možnost sestavit jídelníček pro klienta
- Možnost zobrazit a vytisknout denní rozpis
- Evidence zaměstnanců
- Evidence poskytovaných služeb/výkonů
- Evidence majetku, tisk inventurního soupisu
- Možnost zobrazit a vytisknout sestavu o počtu provedených výkonů a zaplacené částce



## 3. Uživatelská dokumentace

Aplikace byla vytvořena pro potřeby soukromého subjektu, který poskytuje služby v oblasti zdravého životního stylu a volnočasových aktivit. Podporuje rutinní činnosti zaměstnanců, jako je vytváření cvičebních plánů pro klienty nebo návrh jídelníčků. Manažerovi umožňuje vedení evidence zaměstnanců, poskytovaných služeb/výkonů a majetku. Součtovou sestavu může použít pro kontrolu hospodaření.

### 3.1. Systémové požadavky

Aplikace využívá pro svou funkčnost běhové prostředí Microsoft .Net Framework 3.5 a Microsoft SQL Server Compact 3.5. V průběhu instalace aplikace je provedena kontrola, zda jsou tyto produkty nainstalovány. V případě jejich nedostupnosti zobrazí průvodce instalací upozornění a instalace bude ukončena.

Běhové prostředí Microsoft .Net Framework 3.5 můžete nainstalovat spuštěním souboru `dotnetfx35.exe` umístěného v adresáři `bin\microsoft` na instalačním CD.

Microsoft SQL Server Compact 3.5. můžete nainstalovat spuštěním souboru `SQL_Server_Compact_3.5_SP1.msi` umístěného v adresáři `bin\microsoft` na instalačním CD.

Doporučená minimální konfigurace:

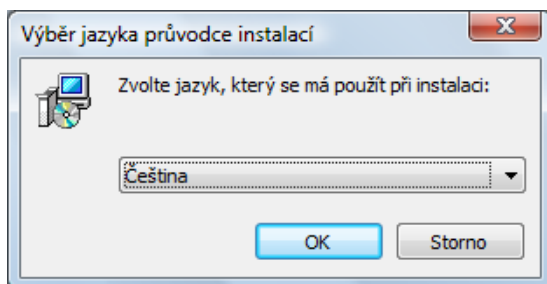
- procesor: 400MHz Pentium nebo kompatibilní
- RAM: 256MB
- operační systém: Microsoft Windows XP a vyšší
- hard disk: přibližně 500MB volného místa v případě nutnosti instalace Microsoft .NET Framework 3.5 a Microsoft SQL Server Compact 3.5.
- webový prohlížeč: například Mozilla Firefox, Internet Explorer apod.

### 3.2. Instalace aplikace

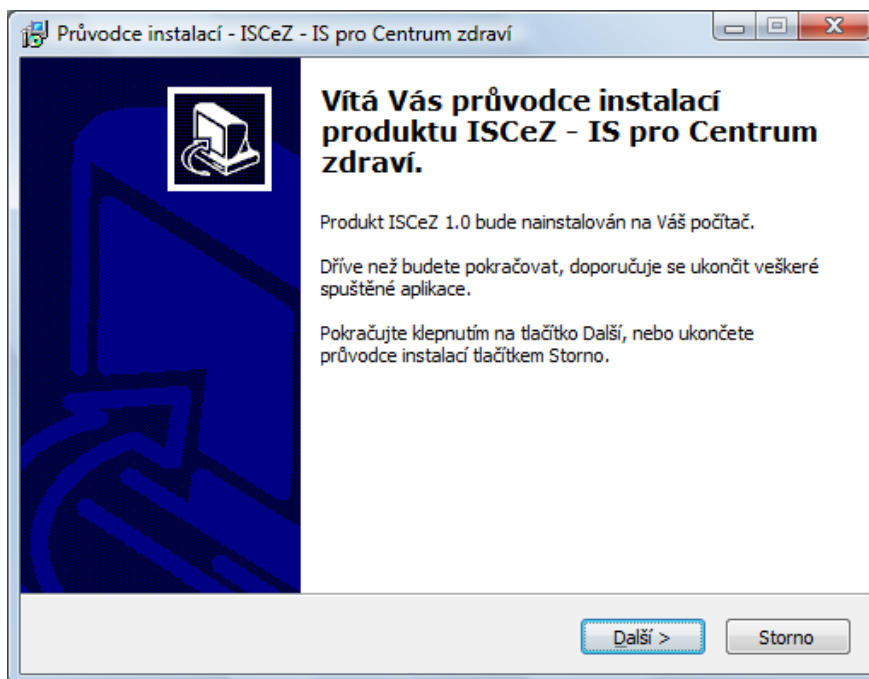
Instalaci musí provádět uživatel, který má nastaveno oprávnění zápisu do adresáře zadaného v kroku 3. Po ukončení instalace je nutné zkontrolovat nastavení oprávnění zápisu do souboru `Data\iscez.sdf` pro všechny uživatele, kteří budou aplikaci používat.

Instalaci aplikace zahájíte spuštěním souboru `ISCeZ_setup.exe` umístěného v adresáři `bin` na instalačním CD. Následně postupujte podle pokynů průvodce instalací:

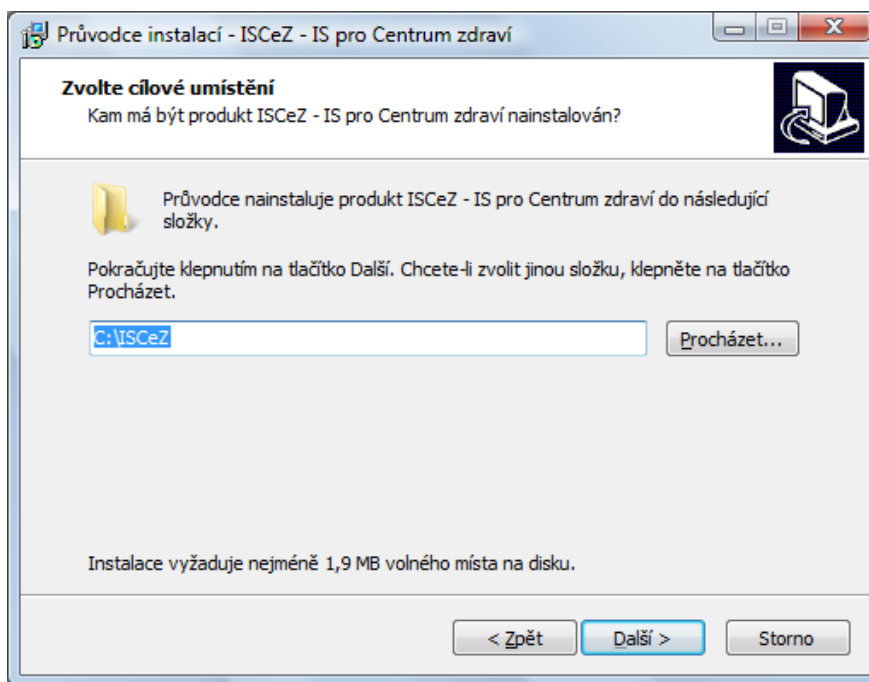
1. Zvolte jazyk instalace a klikněte myší na tlačítko OK.



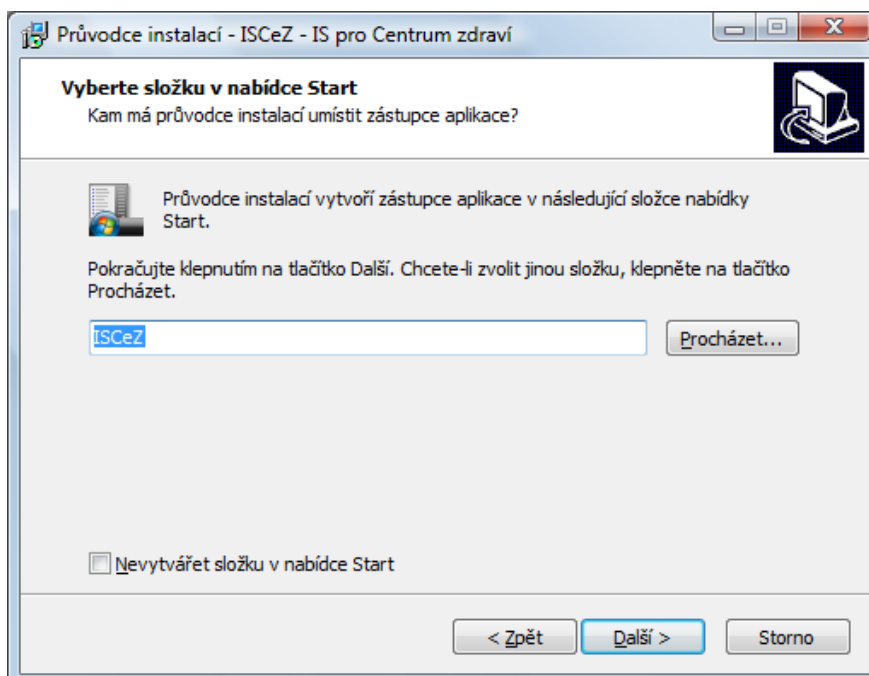
2. Průvodce instalací zobrazí uvítací obrazovku. Klikněte myší na tlačítko Další.



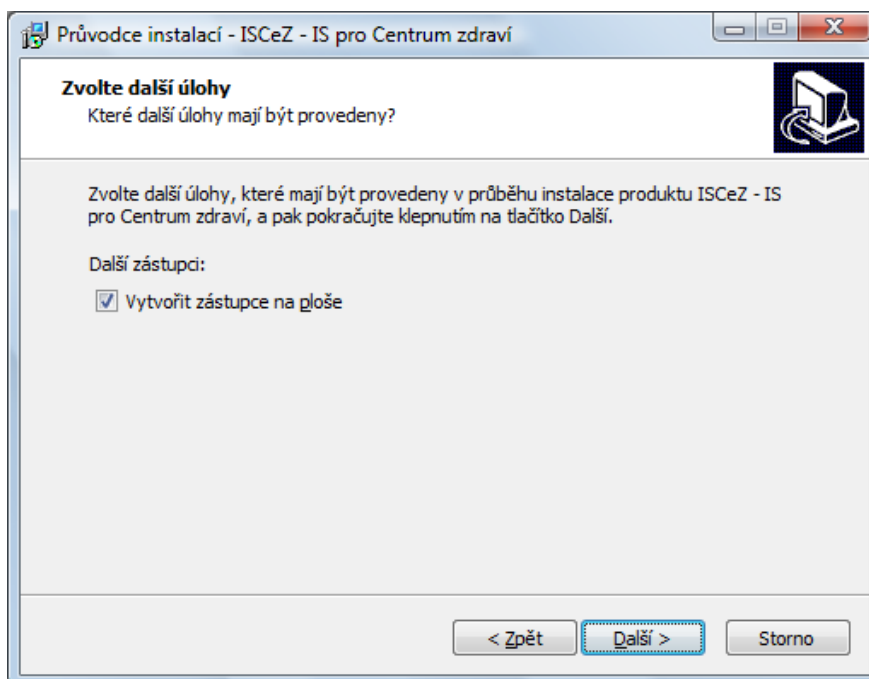
3. Postupujte podle pokynů průvodce instalací a klikněte myší na tlačítko Další.



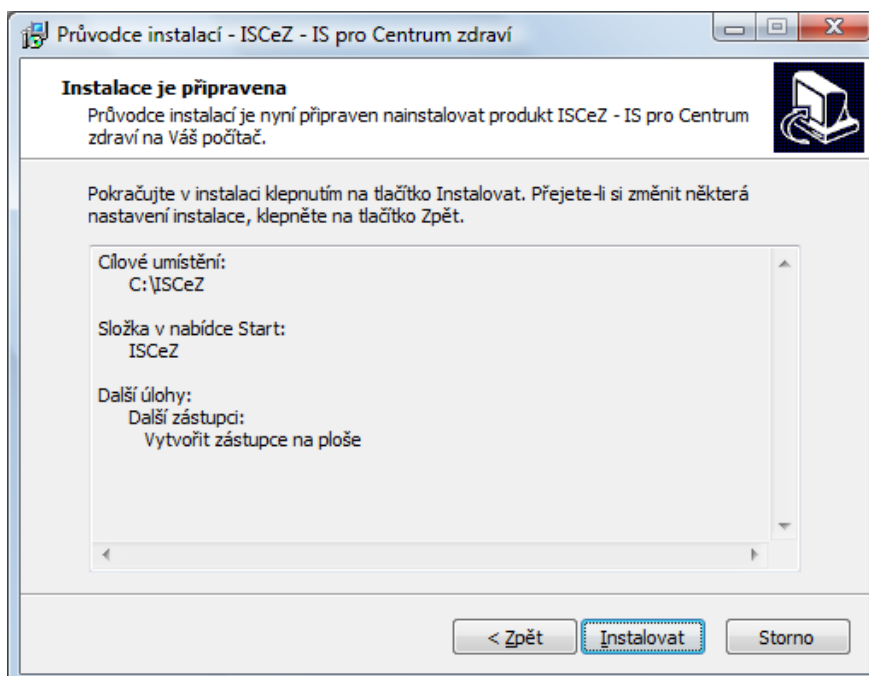
4. Postupujte podle pokynů průvodce instalací a klikněte myší na tlačítko Další.



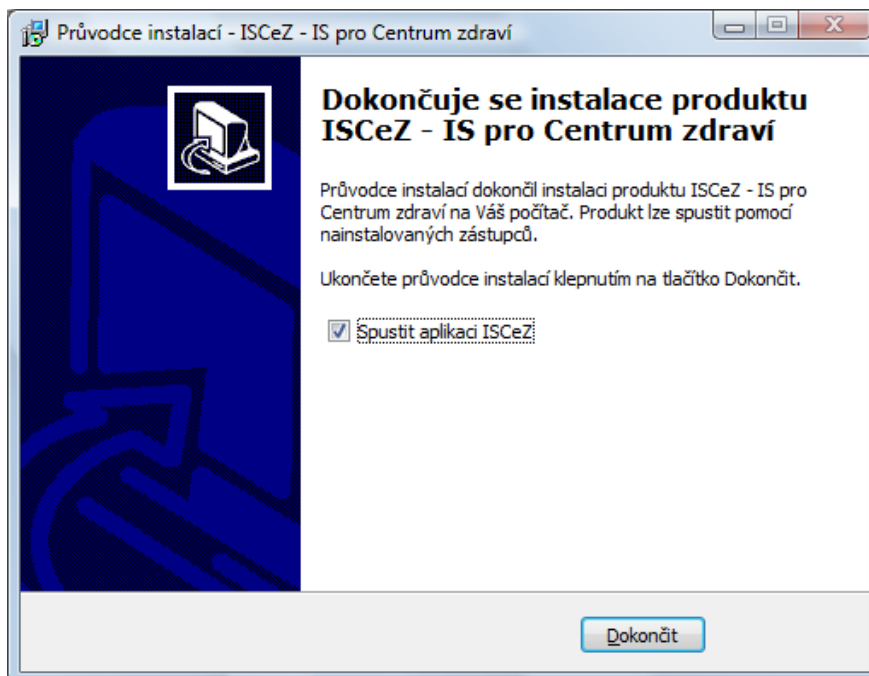
5. Postupujte podle pokynů průvodce instalací a klikněte myší na tlačítko Další.



6. Postupujte podle pokynů průvodce instalací a klikněte myší na tlačítko Instalovat.



7. Pokud nechcete po ukončení průvodce instalací spustit aplikaci, zrušte označení u položky „Spustit aplikaci IS pro Centrum zdraví“. Klikněte myší na tlačítko **Dokončit**.



Tímto krokem je instalace dokončena.

V aplikaci jsou u klientů a zaměstnanců připravena fiktivní testovací data, která neodpovídají skutečnosti.

### 3.3. Popis aplikace

#### 3.3.1. Spuštění aplikace

Aplikaci je možné spustit pomocí položky IS pro Centrum zdraví v nabídce Start > Programy > ISCeZ nebo pomocí zástupce na ploše.

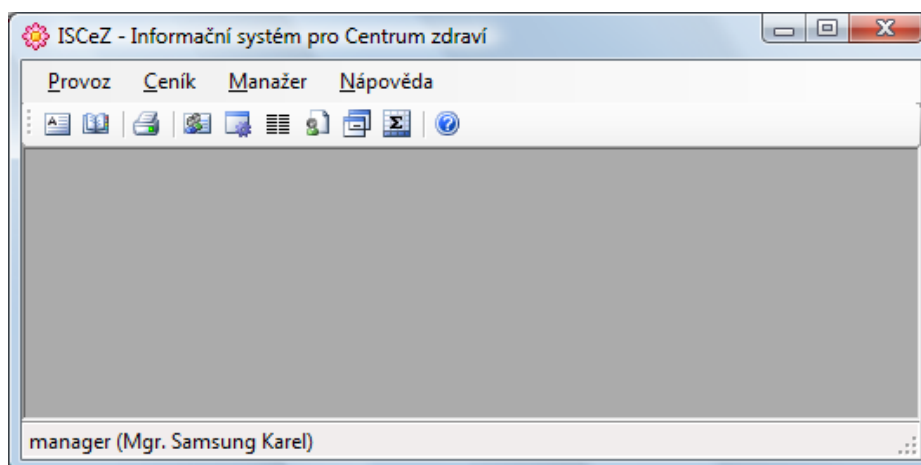
#### 3.3.2. Přihlášení do aplikace

Po spuštění aplikace se zobrazí dialogové okno pro zadání uživatelského jména a hesla. Po úspěšném přihlášení se uživateli zpřístupní vybrané položky menu a tlačítka nástrojové lišty podle přiřazených rolí. Pro otestování funkcí aplikace můžeme použít tato uživatelská jména (v závorce je uvedena přiřazená role): *recepce(Recepce)*, *cvicitel(Cvičitel)*, *manager(Manažer)*. Heslo je pro všechna uvedená uživatelská jména stejné: *password*

#### 3.3.3. Hlavní okno aplikace

Vlastní aplikace je tvořena hlavním formulářem který funguje jako MDI kontejner a v rámci kterého jsou zobrazovány všechny další formuláře. Podoba hlavního formuláře aplikace je zobrazena na obrázku č.1. Formulář lze rozdělit na tři části:

- menu
- nástrojová lišta
- plocha hlavního formuláře



Obrázek 1. Hlavní formulář aplikace

Menu aplikace obsahuje tyto položky:

- **Provoz**

**Registr klientů** – zobrazí formulář s registrem klientů.

**Denní rozpis** – zobrazí formulář s denním rozpisem pro dané datum, službu/výkon a cvičitele.

**Přihlásit** – zobrazí formulář pro zadání uživatelského jména a hesla.

**Změnit heslo** – zobrazí formulář pro změnu hesla.

**Odhlásit** – provede odhlášení uživatele.

**Konec** – ukončí aplikaci.

- **Ceník**

**Tisk ceníku** – vytiskne ceník.

- **Manažer**

**Evidence zaměstnanců** – zobrazí formulář se seznamem zaměstnanců.

**Evidence služeb/výkonů** – zobrazí formulář se seznamem služeb/výkonů.

**Evidence majetku** – zobrazí formulář se seznamem majetku.

**Evidence klientů** – zobrazí formulář se seznamem klientů.

**Šablony jídelníčků** – zobrazí formulář se seznamem šablon jídelníčků.

**Součtová sestava** – zobrazí součtovou sestavu za daný měsíc a rok.

- **Nápověda**

**Obsah** – zobrazí obsah nápovědy.

**O aplikaci** – zobrazí stručné informace o aplikaci.

Dostupnost vybraných položek menu závisí na rolích, které uživateli aplikace přiřadil manažer. Používané role jsou: Recepční, Cvičitel, Manažer. Uživatel s rolí Manažer má dostupné všechny položky menu.

Pro snazší navigaci jsou některé položky menu zpřístupněny pomocí tlačítek nástrojové lišty. Po najetí kurzorem myši na tlačítko se zobrazí nápověda k funkci tlačítka.




### 3.3.4. Registr klientů

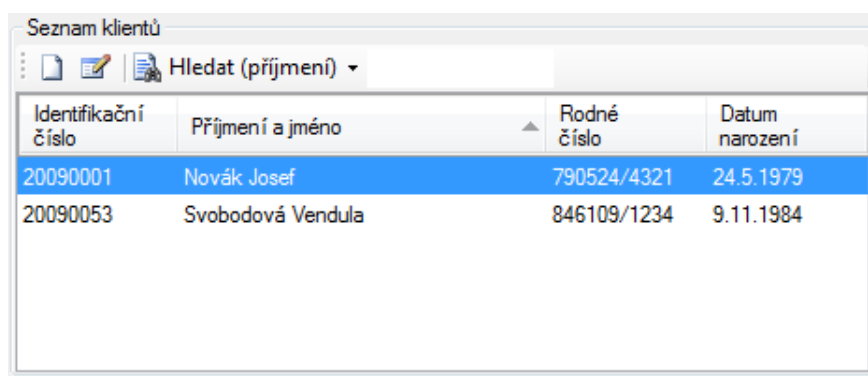
Registr klientů je dostupný uživateli, který má přiřazenou některou z rolí Recepce, Cvičitel nebo Manažer. Formulář je rozdělen na tři části:

- Seznam klientů
- Detail klienta
- Spodní část se záložkami „Rozpis cvičebního plánu“ a „Jídelníčky“

#### Seznam klientů

V této části (viz obr.2.) je zobrazen seznam všech klientů.

- Tlačítko  Založit nového klienta v nástrojové liště zobrazí formulář pro založení nového klienta. Popis tohoto formuláře je uveden v části 3.3.11..
- Tlačítko  Upravit údaje o klientovi v nástrojové liště zobrazí formulář pro editaci údajů aktuálně označeného klienta.
- Pomocí tlačítka  Hledat lze vybrat podmínku k vyhledávání v seznamu klientů. Vyhledávaný řetězec se zadává do textového pole vedle tlačítka.



Identifikační číslo	Příjmení a jméno	Rodné číslo	Datum narození
20090001	Novák Josef	790524/4321	24.5.1979
20090053	Svobodová Vendula	846109/1234	9.11.1984

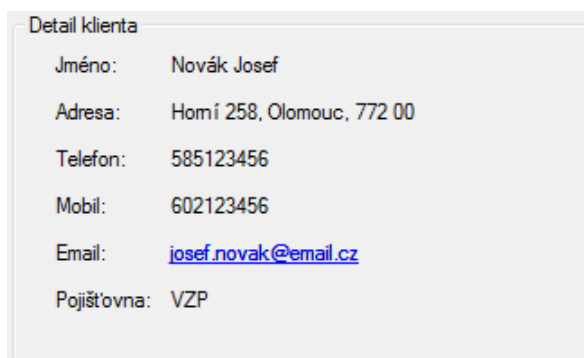
Obrázek 2. Registr klientů - Seznam klientů

Seznam lze seřadit dle hodnot daného sloupce. Kliknutím levého tlačítka myši na zvolený sloupec se seznam seřadí sestupně/vzestupně.



## Detail klienta

V této části (viz obr.3.) jsou zobrazeny detailní informace o aktuálně vybraném klientovi.






The screenshot shows a window titled "Detail klienta" with the following information:

Jméno:	Novák Josef
Adresa:	Horní 258, Olomouc, 772 00
Telefon:	585123456
Mobil:	602123456
Email:	<a href="mailto:josef.novak@email.cz">josef.novak@email.cz</a>
Pojišťovna:	VZP

Obrázek 3. Registr klientů - Detail klienta

## Záložka Rozpis cvičebního plánu

Na této záložce (viz obr.4.) je zobrazen seznam objednaných/poskytnutých služeb/výkonů pro aktuálně označeného klienta. Seznam je seříděn sestupně podle data a času.

- Tlačítko  **Nový plán** v nástrojové liště zobrazí formulář s plánovacím kalendářem. Popis tohoto formuláře následuje dále.
- Tlačítko  **Upravit záznam** v nástrojové liště slouží k editaci aktuálně označeného záznamu. Např. změna již naplánovaného termínu, dodatečné zaplacení. Editace se provádí ve formuláři s plánovacím kalendářem.
- Pomocí tlačítka  **Vymazat záznam** lze vymazat aktuální záznam. Tlačítko je aktivní pouze v případě, kdy není označeno pole „Zaplaceno“. Vymazání záznamu je nutné potvrdit v dialogovém okně.

Datum a čas	Služba/výkon	Cvičitel	Cena	DPH	Cena s DPH	Zaplaceno	Poznámka
29.4.2010 8:30	Kondiční cvičení	Zámal Jiří	50,00 Kč	0 %	50,00 Kč	<input type="checkbox"/>	
22.4.2010 8:30	Kondiční cvičení	Zámal Jiří	50,00 Kč	0 %	50,00 Kč	<input checked="" type="checkbox"/>	

Obrázek 4. Registr klientů - záložka Rozpis cvičebního plánu

### Formulář Plánovací kalendář

Formulář, jehož podoba je zobrazena na obrázku č.5., umožňuje vybranému klientovi naplánovat čerpání služby/výkonu nebo změnit již naplánovaný termín. V měsíčním kalendáři je automaticky zvýrazněno tučným písmem aktuální datum. Pokud se jedná o změnu již naplánovaného termínu, je zvýrazněno tučným písmem i původní datum a v plánovací tabulce je pole označeno zelenou barvou.

Postup plánování:

1. Z rozbalovacího seznamu „Služba/výkon“ vybereme požadovanou službu/výkon. Do pole „Cvičitel“ se automaticky doplní jméno cvičitele, který je přiřazen dané službě/výkonu. Do textového pole „Cena s DPH“ se automaticky doplní cena za službu/výkon.
2. Pokud je to nutné, z rozbalovacího seznamu „Cvičitel“ vybereme jiného cvičitele.
3. Pokud je to nutné, v textovém poli „Cena s DPH“ opravíme cenu za službu/výkon.
4. Pokud je to nutné, v textovém poli „DPH“ opravíme DPH.
5. V měsíčním kalendáři dvojklikem myši vybereme požadované datum. Toto datum se poté zvýrazní tučným písmem a vloží se do plánovací tabulky. Nebo můžeme použít kontextové menu, které zobrazíme kliknutím pravým tlačítkem myši na požadované datum a vybereme odpovídající volbu. Do plánovací tabulky se automaticky doplní již dříve objednaní klienti. Jedná-li se o aktuálního klienta, je pole označeno modrou barvou.
6. Pokud má být služba/výkon naplánována na stejný čas, dvojklikem na vybraný čas jméno klienta zapíšeme do všech datumů v plánovací tabulce.

Pokud má být služba/výkon naplánována na různé časy, jméno klienta zapíšeme dvojklikem přímo do vybraného pole plánovací tabulky. Pole se označí zelenou barvou.

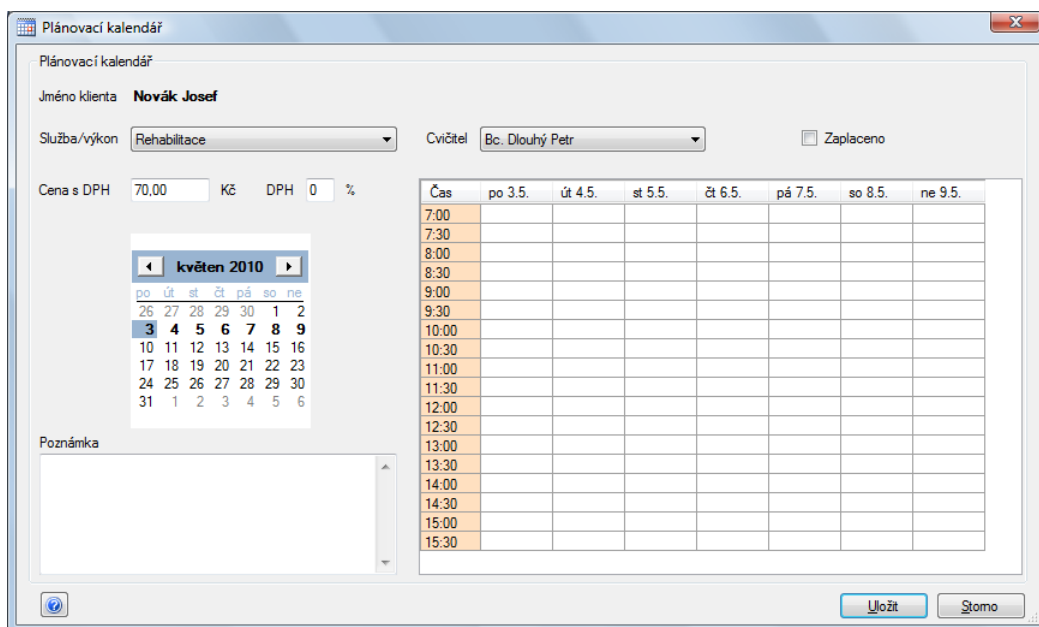
7. Pokud je to nutné, vyplníme textové pole „Poznámka“.
8. V případě, že klient zaplatí částku za čerpání služby/výkonu, označíme zaškrtačací políčko „Zaplaceno“.
9. Pro dokončení plánování stiskneme tlačítko **Uložit**.
10. Pokud je označeno zaškrtačací políčko „Zaplaceno“, zobrazí se formulář s účtenkou. Číslo účtenky se vygeneruje automaticky. Pomocí tlačítka **Tisk** účtenku vytiskneme. Účtenka se nejdříve zobrazí v okně webového prohlížeče z něhož je možné provést fyzicky tisk účtenky. Tlačítkem **Uložit** se plán uloží do seznamu na záložce „Rozpis cvičebního plánu“.

Postup změny již naplánovaného termínu:

1. V měsíčním kalendáři dvojklikem myši vybereme požadované datum. Toto datum se poté zvýrazní tučným písmem a vloží se do plánovací tabulky. Do plánovací tabulky se také automaticky doplní již dříve objednaní klienti. Jedná-li se o aktuálního klienta, je pole označeno modrou barvou.
2. Nový termín vybereme dvojklikem myši přímo do vybraného pole plánovací tabulky. Pole se označí zelenou barvou. Původní termín se označí červenou barvou.
3. Pro dokončení změny termínu stiskneme tlačítko **Uložit**.

Postup při dodatečném zaplacení:





1. Označíme zaškrtačací políčko „Zaplaceno“.
2. Stiskneme tlačítko **Uložit**.
3. Zobrazí se formulář s účtenkou. Číslo účtenky se vygeneruje automaticky. Pomocí tlačítka **Tisk** účtenku vytiskneme. Účtenka se nejdříve zobrazí v okně webového prohlížeče z něhož je možné provést fyzicky tisk účtenky. Tlačítkem **Uložit** se změna uloží.



Obrázek 5. Plánovací kalendář

### Záložka Jídelníčky

Na této záložce (viz obr.6.) je zobrazen seznam navržených jídelníčků pro aktuálně označeného klienta. V pravé části je zobrazen detailní rozpis aktuálně označeného záznamu.

- Tlačítko  **Nový jídelníček** v nástrojové liště zobrazí formulář pro návrh nového jídelníčku. Popis tohoto formuláře následuje dále.
- Tlačítko  **Upravit záznam** v nástrojové liště slouží k editaci aktuálně označeného záznamu.
- Pomocí tlačítka  **Vymazat záznam** lze vymazat aktuální záznam. Vymazání záznamu je nutné potvrdit v dialogovém okně.
- Pomocí tlačítka  **Tisk** vytiskneme zobrazený jídelníček. Jídelníček se nejdříve zobrazí v okně webového prohlížeče z něhož je možné provést fyzický tisk.



Datum	Popis
22.4.2010	Dietní režim
29.4.2010	Dietní režim

1. den		
snídaně	50g Tmavé pečivo s drůbeží šunkou a okurkou	(1140kJ)
svačina	Knäckenbrot s pomazánkovým máslem a mrkví	(1180kJ)
oběd	150g Kuřecí prsíčka se špenátem a vařenými těstovinami	(1975kJ)
svačina	Tvaroh s bobulovitým ovocem	(660kJ)
večeře	150g Kuřecí prsa v aspiku, 2ks křehký chléb	(940kJ)
2. den		
snídaně	50g Sendvič se sušenou šunkou	(1040kJ)
svačina	Mandarinka s bílým jogurtem	(650kJ)
oběd	250g Papoušek na zelenině se žampiony a vařeným bramborem	(1720kJ)


Obrázek 6. Registr klientů - záložka Jídelníčky

### Formulář Návrh jídelníčku

Formulář, jehož podoba je zobrazena na obr.7., umožňuje založit nový návrh jídelníčku nebo údaje editovat. Pokud zakládáme nový návrh jídelníčku, je tabulka s detailním rozpisem přednastavena na 7 dnů. Do pole „Datum“ zadáme datum, od kterého bude klient jídelníček používat. Do textového pole „Popis“ vyplníme krátký popis obsahu jídelníčku. Další dny můžeme přidat pomocí kontextového menu, které se zobrazí po kliknutí pravým tlačítkem na tabulku s detailním rozpisem. Jídelníček můžeme vybrat z již přednastavených šablon. Požadovanou šablonu vybereme z rozbalovací nabídky „Výběr šablony“ a stiskneme tlačítko  Použít vybranou šablonu. Navrhnutý jídelníček můžeme také uložit jako šablonu. Do pole „Výběr šablony“ vyplníme název šablony a stiskneme tlačítko  Uložit jako šablonu. Tlačítkem Uložit se navrhnutý jídelníček uloží do seznamu na záložce „Jídelníčky“. Tlačítkem Storno zavřeme formulář bez uložení změn.

Obrázek 7. Návrh jídelníčku

### 3.3.5. Denní rozpis

Denní rozpis je dostupný uživateli, který má přiřazenou některou z rolí Recepce, Cvičitel nebo Manažer. Formulář, jehož podoba je zobrazena na obr.8., slouží k zobrazení denního rozpisu pro dané datum, službu/výkon a cvičitele. Automaticky je přednastaven na aktuální datum a všechny objednané služby. Pokud má uživatel přiřazenou roli Cvičitel, zobrazí se denní rozpis pouze pro něj. Pomocí tlačítka  Tisk denní rozpis vytiskneme. Denní rozpis se nejdříve zobrazí v okně webového prohlížeče z něhož je možné provést fyzicky tisk.

Čas	Služba/výkon	Cvičitel	Jméno klienta
7:00			
7:30			
8:00			
8:30			
9:00	Masáž	Bc. Dlouhý Petr	Ing. Svobodová Vendula
9:30			
10:00	Rehabilitace	Bc. Dlouhý Petr	Novák Josef
10:30			
11:00			
11:30			

Obrázek 8. Denní rozpis

### 3.3.6. Změna hesla

Formulář, jehož podoba je zobrazena na obr.9., umožňuje změnit heslo aktuálně přihlášeného uživatele. Délka hesla musí být minimálně 6 znaků. Při chybném vyplnění se vedle textového pole objeví chybová ikona . Po najetí nad ikonu kurzorem myši se zobrazí chybové hlášení.

Postup změny hesla:

1. Do textového pole „Aktuální heslo“ zadáme aktuální heslo.
2. Do textového pole „Nové heslo“ zadáme nové heslo.
3. V textovém poli „Potvrzení hesla“ potvrdíme nové heslo.
4. Stiskneme tlačítko **Změnit**. O úspěšné změně hesla budeme informováni.




Obrázek 9. Změna hesla

### 3.3.7. Tisk ceníku

Tato položka menu slouží k tisku ceníku poskytovaných služeb/výkonů. Ceník se nejdříve zobrazí v okně webového prohlížeče z něhož je možné provést fyzicky tisk ceníku.


### 3.3.8. Evidence zaměstnanců

Evidence zaměstnanců je dostupná uživateli, který má přiřazenou roli Manažer. Zaměstnanci, kterým již skončil pracovní poměr, jsou v seznamu označeni šedým podkladem. V horní části formuláře je zobrazena nástrojová lišta, která obsahuje tlačítka pro navigaci v seznamu a:

- tlačítko  Přidat nového zaměstnance pro založení nového zaměstnance,
- tlačítko  Upravit údaje o zaměstnanci pro editaci údajů aktuálně označeného zaměstnance,
- tlačítko  Vymazat záznam pro vymazání aktuálně označeného zaměstnance.

Vymazání aktuálně označeného zaměstnance je nutné potvrdit v dialogovém okně. Pokud zvolíme možnost Ano, provede se kontrola, zda není zaměstnanec evidován jako odpovědná osoba majetku nebo není odpovědná osoba služby/výkonu nebo není evidován jako cvičitel. Pokud neproběhne kontrola úspěšně, je o tom uživatel informován chybovým hlášením.

#### Formulář pro editaci údajů o zaměstnanci

Formulář, jehož podoba je zobrazena na obr.10., umožňuje založit nového zaměstnance nebo údaje editovat. Povinná pole jsou zvýrazněna tučným písmem. Pokud zakládáme nového zaměstnance, je datum v poli „Začátek PP“ (*Začátek pracovního poměru*) nastaven na první den aktuálního měsíce a datum v poli „Konec PP“ (*Konec pracovního poměru*) nastaven na 31.12.9999. Pokud je označeno zaškrtačací políčko „Externista“, nemusíme vyplňovat datum „Začátek PP“ a „Konec PP“. Po stisknutí tlačítka **Uložit** se provede kontrola vyplnění povinných polí, formátu data narození, data začátku a konce PP a délky hesla. Délka hesla je min. 6 znaků. Dále je provedena kontrola existence alespoň jednoho zaměstnance s rolí Manažer. Při chybném vyplnění se vedle textového pole objeví chybová ikona . Po najetí nad ikonu kurzorem myši se zobrazí chybové hlášení. Tlačítkem **Storno** zavřeme formulář bez uložení změn.





Obrázek 10. Údaje o zaměstnanci

### 3.3.9. Evidence služeb/výkonů




Evidence služeb je dostupná uživateli, který má přiřazenou roli Manažer. Skupina služeb/výkonů je zvýrazněna tučným písmem. V horní části formuláře je zobrazena nástrojová lišta, která obsahuje tlačítka pro navigaci v seznamu a:

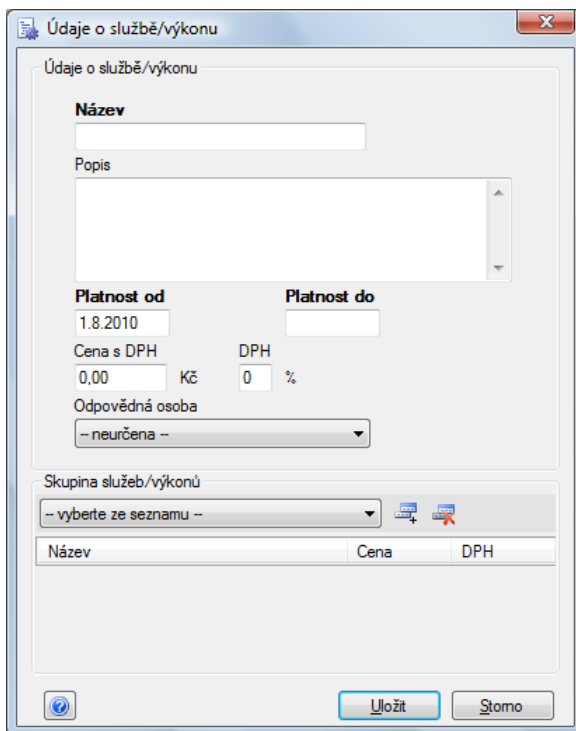
- tlačítko Přidat novou služby/výkon pro založení nové služby/výkonu,
- tlačítko Upravit údaje o službě/výkonu pro editaci údajů aktuálně označené služby/výkonu,
- tlačítko Vymazat záznam pro vymazání aktuálně označené služby/výkonu,
- tlačítko Tisk ceníku pro tisk ceníku,
- tlačítko Nastavit filtr vyhledávání pro nastavení filtru vyhledávání.

Vymazání aktuálně označené služby/výkonu je nutné potvrdit v dialogovém okně. Pokud zvolíme možnost Ano, provede se kontrola, zda již nebyla služba/výkon naplánována nebo nepatří do skupiny služeb/výkonů. Pokud neproběhne kontrola úspěšně, je o tom uživatel informován chybovým hlášením.

Pomocí tlačítka  Tisk ceníku vytiskneme ceník. Ceník se nejdříve zobrazí v okně webového prohlížeče z něhož je možné provést fyzicky tisk. Pomocí tlačítka  Nastavit filtr vyhledávání máme možnost označit volbu „Vybrat vše“, která zobrazí i služby/výkony se skončenou dobou platnosti.

### Formulář pro editaci údajů o službě/výkonu

Formulář, jehož podoba je zobrazena na obr.11., umožňuje založit novou službu/výkon nebo údaje editovat. Povinná pole jsou zvýrazněna tučným písmem. Pokud zakládáme novou službu/výkon, je datum v poli „Platnost od“ nastaveno na první den aktuálního měsíce. Ve spodní části formuláře je část pro vytvoření/úpravy skupiny služeb/výkonů. Tlačítkem  Přidat službu/výkon do skupiny přidáme službu vybranou z rozbalovacího seznamu do skupiny. Tlačítkem  Odebrat službu/výkon ze skupiny odebereme označenou službu ze skupiny. Po stisknutí tlačítka Uložit se provede kontrola vyplnění povinných polí, formátu data v poli „Platnost od“ a data v poli „Platnost do“. Pokud není pole „Platnost do“ vyplněno, nastaví se na datum 31.12.9999. Při chybném vyplnění se vedle textového pole objeví chybová ikona . Po najetí nad ikonu kurzorem myši se zobrazí chybové hlášení. Tlačítkem Storno zavřeme formulář bez uložení změn.



Údaje o službě/výkonu

Údaje o službě/výkonu

**Název**

Popis

**Platnost od** 1.8.2010 **Platnost do**

Cena s DPH 0,00 Kč DPH 0 %

Odpovědná osoba --neurčena--

Skupina služeb/výkonů






-- vyberte ze seznamu --



Název	Cena	DPH
-------	------	-----

Obrázek 11. Údaje o službě/výkonu


### 3.3.10. Evidence majetku

Evidence majetku je dostupná uživateli, který má přiřazenou roli Manažer. V horní části formuláře je zobrazena nástrojová lišta, která obsahuje tlačítka pro navigaci v seznamu a:

- tlačítko  **Založit nový majetek** pro založení nového majetku,
- tlačítko  **Upravit údaje o majetku** pro editaci údajů aktuálně označeného majetku,
- tlačítko  **Vymazat záznam** pro vymazání aktuálně označeného majetku,
- tlačítko  **Tisk inventurního soupisu** pro tisk inventurního soupisu,
- tlačítko  **Nastavit filtr vyhledávání** pro nastavení filtru vyhledávání.

Vymazání aktuálně označeného majetku je nutné potvrdit v dialogovém okně. Pomocí tlačítka  **Tisk inventurního soupisu** vytiskneme inventurní soupis, který se vytvoří z aktuálně zobrazených záznamů. Inventurní soupis se nejdříve zobrazí v okně webového prohlížeče z něhož je možné provést fyzicky tisk. Pomocí tlačítka  **Nastavit filtr vyhledávání** můžeme vybrat podle čeho se bude vyhledávat v seznamu majetku. Vyhledávaný řetězec zadáme do textového pole vedle tlačítka. Pokud označíme volbu „Zobrazit vyřazený majetek“, do seznamu se doplní i vyřazený majetek, který je označen šedým podkladem.

#### Formulář pro editaci údajů o majetku

Formulář, jehož podoba je zobrazena na obr.12., umožňuje založit nový majetek nebo údaje editovat. Povinná pole jsou zvýrazněna tučným písmem. Pokud zakládáme nový majetek, vygeneruje se nové inventurní číslo ve tvaru YYYYMMXXXX, kde: YYYY je aktuální rok, MM je číslo aktuálního měsíce, XXXX je pořadové číslo v aktuálním měsíci a roku. Po stisknutí tlačítka **Uložit** se provede kontrola vyplnění povinných polí. Při chybném vyplnění se vedle textového pole objeví chybová ikona . Po najetí nad ikonu kurzorem myši se zobrazí chybové hlášení. Tlačítkem **Storno** zavřeme formulář bez uložení změn.

Obrázek 12. Majetková karta

### 3.3.11. Evidence klientů

Evidence klientů je dostupná uživateli, který má přiřazenou roli Manažer. V horní části formuláře je zobrazena nástrojová lišta, která obsahuje tlačítka pro navigaci v seznamu a:

- tlačítko Přidat nového klienta pro založení nového majetku,
- tlačítko Upravit údaje o klientovi pro editaci údajů aktuálně označeného klienta,
- tlačítko Vymazat záznam pro vymazání aktuálně označeného klienta.

Vymazání aktuálně označeného klienta je nutné potvrdit v dialogovém okně.




#### Formulář pro editaci údajů o klientovi

Formulář, jehož podoba je zobrazena na obr.13., umožňuje založit nového klienta nebo údaje editovat. Povinná pole jsou zvýrazněna tučným písmem. Pokud zakládáme nového klienta, vygeneruje se identifikační číslo klienta ve tvaru YYYYXXXX, kde: YYYY je aktuální rok a XXXX je pořadové číslo v aktuálním roku. Po stisknutí tlačítka Uložit se provede kontrola vyplnění povinných polí a formátu data narození. Při chybném vyplnění se vedle textového pole objeví chybová ikona . Po najetí nad ikonu kurzorem myši se zobrazí chybové hlášení. Tlačítkem Storno zavřeme formulář bez uložení změn.

Obrázek 13. Údaje o klientovi


### 3.3.12. Šablony jídelníčků

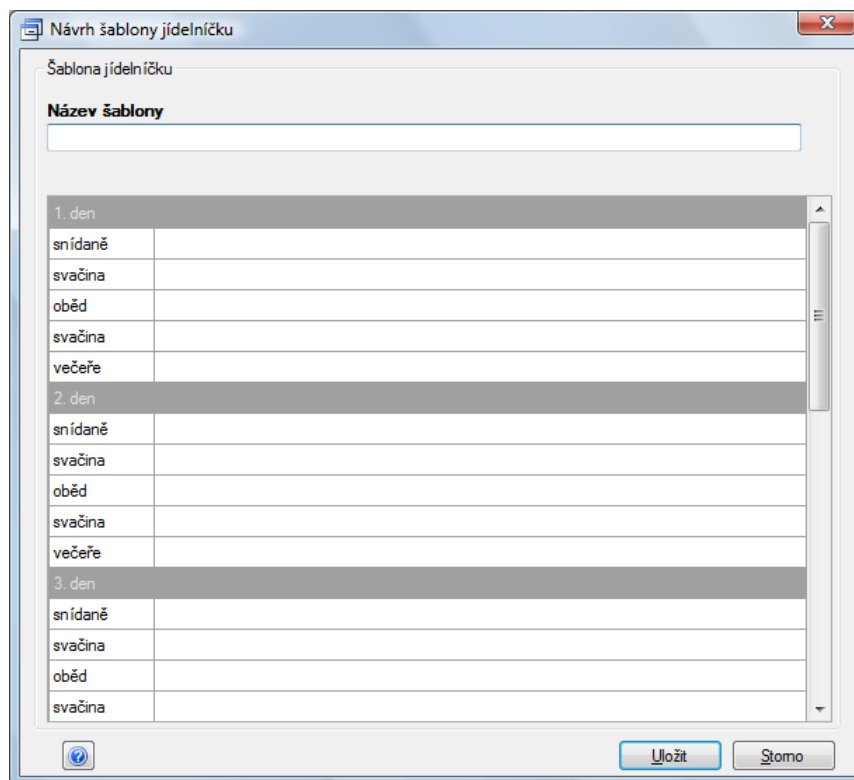
Šablony jídelníčků jsou dostupné uživateli, který má přiřazenou roli Manažer. V levé části formuláře je zobrazen seznam šablon jídelníčků. Nad seznamem je zobrazena nástrojová lišta, která obsahuje:

- tlačítko  **Nová šablona** pro založení nové šablony,
- tlačítko  **Upravit šablonu** pro editaci údajů aktuálně vybrané šablony,
- tlačítko  **Vymazat šablonu** pro vymazání aktuálně vybrané šablony.

Vymazání aktuálně označené šablony jídelníčku je nutné potvrdit v dialogovém okně. V pravé části formuláře je zobrazen detailní rozpis aktuálně vybrané šablony jídelníčku.


#### Formulář pro editaci šablony jídelníčku

Formulář, jehož podoba je zobrazena na obr.14., umožňuje založit novou šablonu jídelníčku nebo údaje editovat. Povinná pole jsou zvýrazněna tučným písmem. Pokud zakládáme novou šablonu jídelníčku, je tabulka s detailním rozpisem přednastavena na 7 dnů. Další dny můžeme přidat pomocí kontextového menu, které se zobrazí po kliknutí pravým tlačítkem na tabulku s detailním rozpisem. Po stisknutí tlačítka **Uložit** se provede kontrola vyplnění povinných polí. Při chybném vyplnění se vedle textového pole objeví chybová ikona . Po najetí nad ikonu kurzorem myši se zobrazí chybové hlášení. Tlačítkem **Storno** zavřeme formulář bez uložení změn.



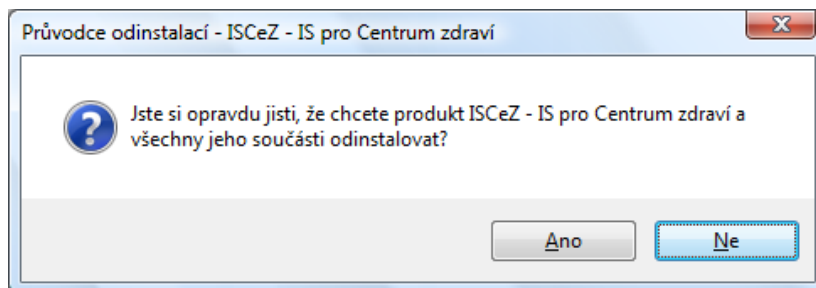
Obrázek 14. Návrh šablony jídelníčku

### 3.3.13. Součtová sestava

Součtová sestava je dostupná uživateli, který má přiřazenou roli Manažer. Formulář slouží k zobrazení součtové sestavy za daný měsíc a rok. Tuto sestavu pak manažer používá k vyhodnocení hospodaření. Po zobrazení formuláře je hodnota v rozbalovacím seznamu „Měsíc“ nastavena na aktuální měsíc. Hodnota v rozbalovacím seznamu „Rok“ je nastavena na aktuální rok. Pomocí tlačítka  Tisk součtovou sestavu vytiskneme. Součtová sestava se nejdříve zobrazí v okně webového prohlížeče z něhož je možné provést fyzicky tisk.

## 3.4. Odinstalování aplikace

Aplikaci můžete odinstalovat pomocí volby v nabídce **Start > Programy > ISCeZ > Odinstalovat aplikaci IS pro Centrum zdraví**. Následně se spustí průvodce odinstalací.



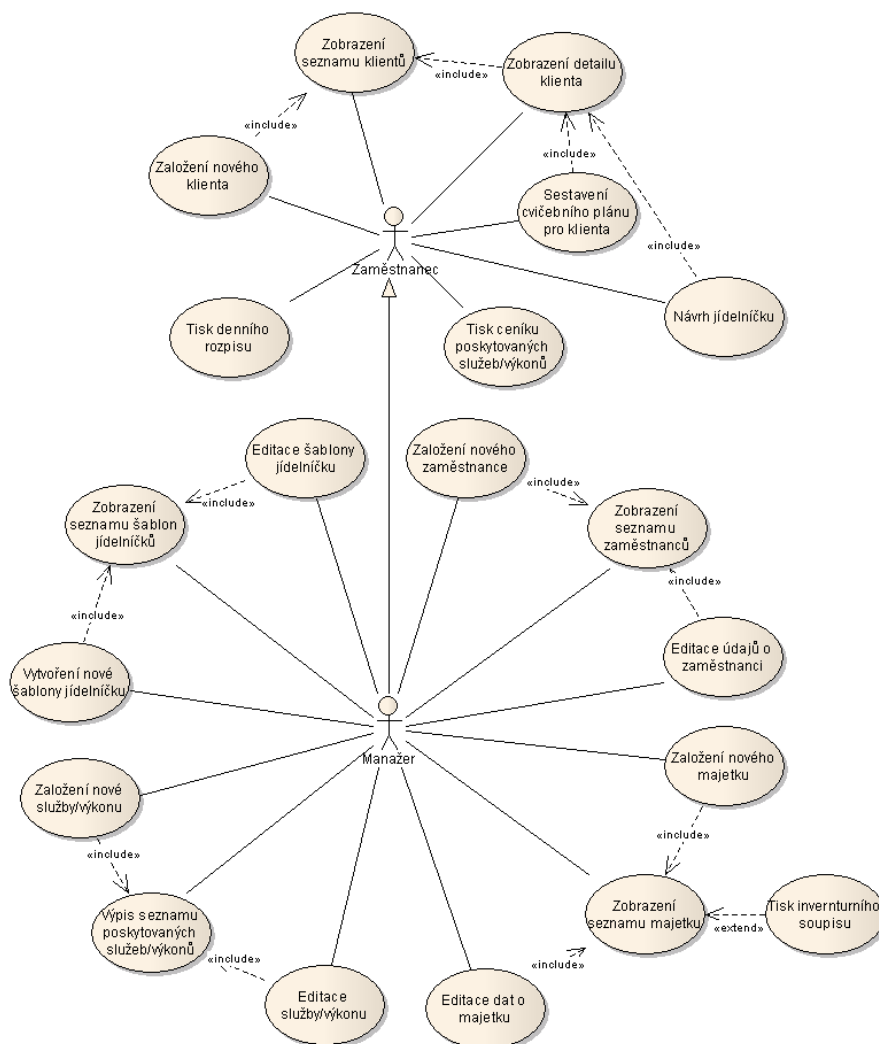
Klikněte na tlačítko **Ano**. Průvodce odinstalací spustí vlastní proces odinstalace. O úspěšném odinstalování aplikace Vás bude průvodce informovat.

## 4. Programátorská část

### 4.1. Použité technologie

Aplikace je vytvořena v prostředí Microsoft Visual Studia 2008 [6] a jazyku C#, za použití Microsoft .NET Frameworku 3.5. Pro ukládání dat používá databázový stroj Microsoft SQL Server Compact 3.5 [4]. Pro přístup k datům využívá integrovaného jazyka pro dotazování LINQ pro SQL [2]. Tiskové výstupy jsou realizovány pomocí XSLT transformace [5].

### 4.2. Základní use case diagram

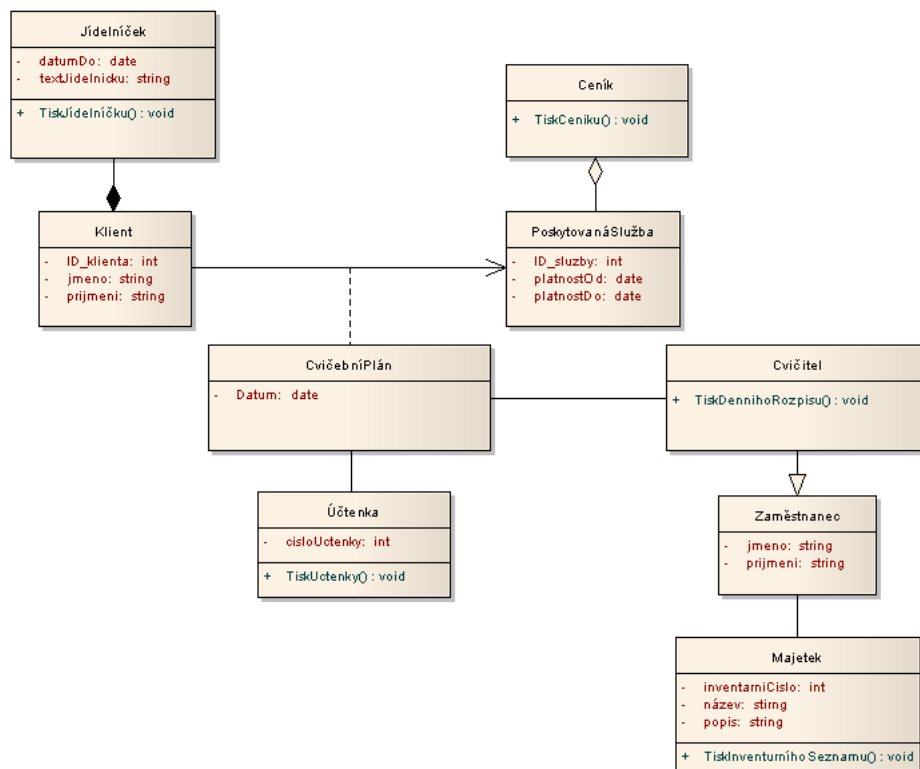


Obrázek 15. Základní use case diagram



### 4.3. Analytický model tříd

Pro jednoduchost analytického modelu tříd (obr.16.) jsou uvedeny pouze nejdůležitější atributy a operace.



Obrázek 16. Analytický model tříd

### 4.4. Architektura aplikace

Architektura aplikace je založena na třívrstevém modelu složeném z datové vrstvy (Data Access Layer), logické vrstvy (Business Logic Layer) a prezentační vrstvy (Presentation Layer). Rozdělení do vrstev umožňuje měnit a spravovat danou vrstvu, aniž by se to dotklo ostatních vrstev. Tento model je často využíván v aplikacích pracujících s databází.

### 4.5. Datová vrstva (Data Access Layer)

Datová vrstva je uložena v projektu ISCeZ.DAL. Obsahuje třídu `IscezDataContext`, která obstarává komunikaci mezi LINQ a databází. Dále ob-

sahuje třídy jednotlivých entit, které poskytují metadata pro dotazovací stroj LINQ a vytváří úložiště pro data načtená z databáze. Pro definici mapování je použit soubor DBML vytvořený pomocí nástroje SQLMetal.

## 4.6. Logická vrstva (Business Logic Layer)

Logická vrstva je uložena v projektu ISCeZ.BLL. Třídy této vrstvy zajišťují komunikaci mezi vrstvami a obsahují veškerou aplikační logiku. Ve všech třídách odvozených od třídy `BaseManager` je implementován návrhový vzor Singleton [3], který zajistí, aby třída měla pouze jednu instanci. K metodám této instance pak přistupujeme pomocí statické vlastnosti `Instance`. Následuje popis tříd této vrstvy a popis nejdůležitějších metod.

- `BaseManager`  
Abstraktní třída, která poskytuje všem odvozeným třídám datový člen `ISCeZ.DAL.IscezDataContext dc` a metodu `void SubmitChanges()`, která provádí aktualizaci dat v databázi.
- `AssetManager`  
Odvozená třída od abstraktní třídy `BaseManager`. Poskytuje metody pro práci s entitou `Majetek (ISCeZ.DAL.Asset)`.

```
public IQueryable<Asset> GetAssets()
```

Vrátí seznam majetku.

```
public IQueryable<Asset> GetAssets(bool discarded)
```

Vrátí seznam majetku. Parametr `discarded` určuje, zda se má do seznamu zahrnout i vyřazený majetek.

```
public IQueryable<Asset> GetAssetsByID(int assetID)
```

Vrátí seznam majetku, u kterého začátek inventárního čísla odpovídá parametru `assetID`.

```
public IQueryable<Asset> GetAssetsByID(int assetID, bool discarded)
```

Vrátí seznam majetku, u kterého začátek inventárního čísla odpovídá parametru `assetID`. Parametr `discarded` určuje, zda se má do seznamu zahrnout i vyřazený majetek.

```
public IQueryable<Asset> GetAssetsByRoomNumber(string roomNumber)
```

Vrátí seznam majetku, u kterého začátek čísla místnosti odpovídá parametru `roomNumber`.

```
public IQueryable<Asset> GetAssetsByRoomNumber(string roomNumber, bool discarded)
```

Vrátí seznam majetku, u kterého začátek čísla místnosti odpovídá

parametru `roomNumber`. Parametr `discarded` určuje, zda se má do seznamu zahrnout i vyřazený majetek.

```
public IQueryable<Asset> GetAssetsByEmployeeName(string name)
```

Vrátí seznam majetku, u kterého začátek příjmení odpovědné osoby odpovídá parametru `name`.

```
public IQueryable<Asset> GetAssetsByEmployeeName(string name, bool discarded)
```

Vrátí seznam majetku, u kterého začátek příjmení odpovědné osoby odpovídá parametru `name`. Parametr `discarded` určuje, zda se má do seznamu zahrnout i vyřazený majetek.

```
public Asset GetAsset(int assetID)
```

Vrátí majetek, jehož inventární číslo odpovídá parametru `assetID`.

```
public XDocument GetAssetsAsXML(object assets)
```

Vrátí seznam majetku jako XML dokument.

```
public int GetNewAssetID()
```

Vygeneruje nové inventární číslo.

```
public void InsertOnSubmit(Asset asset)
```

Vytvoří nový záznam v tabulce `Asset`.

```
public void DeleteOnSubmit(Asset asset)
```

Odstraní záznam z tabulky `Asset`.

- **BillOfFareManager**

Odvozená třída od abstraktní třídy `BaseManager`. Poskytuje metody pro práci s entitami `Jídelníček (ISCeZ.DAL.BillOfFare)` a `Šablona jídelníčku (ISCeZ.DAL.BillOfFareTemplate)`.

```
public IQueryable<BillOfFare> GetBillOfFareByClientID(int clientID)
```

Vrátí seznam jídelníčků pro klienta, jehož identifikační číslo odpovídá parametru `clientID`.

```
public BillOfFare GetBillOfFare(int id)
```

Vrátí jídelníček, jehož `BillOfFareID` odpovídá parametru `id`.

```
public IQueryable<BillOfFareTemplate> GetBillOfFareTemplates()
```

Vrátí seznam všech šablon jídelníčků.

```
public BillOfFareTemplate GetBillOfFareTemplate(int id)
```

Vrátí šablonu jídelníčku, jejíž `BillOfFareTemplateID` odpovídá parametru `id`.

```
public BillOfFareTemplate GetBillOfFareTemplate(string name)
```

Vrátí šablonu jídelníčku, jejíž název odpovídá parametru name.

```
public XDocument GetBillOfFareAsXML(BillOfFare billOfFare)
```

Vrátí jídelníček jako XML dokument.

```
public void InsertBillOfFareOnSubmit(BillOfFare billOfFare)
```

Založí nový záznam do tabulky BillOfFare.

```
public void InsertBillOfFareItemOnSubmit(BillOfFareItem item)
```

Založí nový záznam do tabulky BillOfFareItem.

```
public void DeleteBillOfFareOnSubmit(BillOfFare billOfFare)
```

Odstraní záznam z tabulky BillOfFare a záznamy z tabulky BillOfFareItem.

```
public void InsertBillOfFareTemplateOnSubmit(BillOfFareTemplate template)
```

Založí nový záznam do tabulky BillOfFareTemplate.

```
public void InsertBillOfFareTemplateItemOnSubmit(BillOfFareTemplateItem item)
```

Založí nový záznam do tabulky BillOfFareTemplateItem.

```
public void DeleteBillOfFareTemplateOnSubmit(BillOfFareTemplate template)
```

Odstraní záznam z tabulky BillOfFareTemplate a záznamy z tabulky BillOfFareTemplateItem.

- **ClientManager**

Odvozená třída od abstraktní třídy `BaseManager`. Poskytuje metody pro práci s entitami Klient (`ISCeZ.DAL.Client`), Plán čerpání služeb/výkonů (`ISCeZ.DAL.ClientPlan`) a Účtenka (`ISCeZ.DAL.Bill`).

```
public IQueryable<Client> GetClients()
```

Vrátí seznam všech klientů.

```
public IQueryable<Client> GetClientsByLastName(string lastName)
```

Vrátí seznam klientů, jejichž začátek příjmení odpovídá parametru `lastName`.

```
public IQueryable<Client> GetClientsByPIN(string pin)
```

Vrátí seznam klientů, jejichž začátek rodného čísla odpovídá parametru `pin`.

```

public IQueryable<Client> GetClientsByID(int id)
Vrátí seznam klientů, jejichž začátek identifikačního čísla odpovídá
parametru id.

public IQueryable<ClientPlan> GetClientPlanByClientID(int
clientID)
Vrátí seznam plánu čerpání služeb/výkonů pro klienta, jehož
identifikační číslo odpovídá parametru clientID.

public IQueryable<ClientPlan> GetClientPlanByPlanDate(DateTime
planDate)
Vrátí seznam plánu čerpání služeb/výkonů pro datum v parametru
planDate.

public IQueryable<ClientPlan> GetClientPlanByPlanDate(DateTime
planDate, int serviceID)
Vrátí seznam plánu čerpání služeb/výkonů pro datum v parametru
planDate a službu, jejíž ServiceID odpovídá parametru serviceID.

public IQueryable<ClientPlan> GetClientPlanByEmployee(DateTime
planDate, int employeeID)
Vrátí seznam plánu čerpání služeb/výkonů pro datum v parametru
planDate a cvičitele, jehož EmployeeID odpovídá parametru
employeeID.

public int GetNewClientID()
Vygeneruje nové identifikační číslo klienta.

public Bill GetBill(int billID)
Vrátí účtenku jejíž BillID odpovídá parametru billID.

public XDocument GetBillAsXML(Bill bill, Client client,
List<ClientPlan> items)
Vrátí účtenku jako XML dokument.

public int GetNewBillID()
Vygeneruje nové číslo účtenky.

public void InsertOnSubmit(Client client)
Založí nový záznam do tabulky Client.

public void DeleteOnSubmit(Client client)
Odstraní záznam z tabulky Client.

public void InsertClientPlanOnSubmit(ClientPlan
clientPlan)
Založí nový záznam do tabulky ClientPlan.

public void DeleteClientPlanOnSubmit(ClientPlan
clientPlan)
Odstraní záznam z tabulky ClientPlan.

```

```
public void InsertBillOnSubmit(Bill bill)
Založí nový záznam do tabulky Bill.
```

- **EmployeeManager**

Odvozená třída od abstraktní třídy `BaseManager`. Poskytuje metody pro práci s entitou `Zaměstnanec (ISCeZ.DAL.Employee)`.

```
public IQueryable<Employee> GetEmployees()
Vrátí seznam všech zaměstnanců.
```

```
public Employee GetEmployee(int employeeID)
Vrátí zaměstnance, jehož EmployeeID odpovídá parametru employeeID.
```

```
public Employee GetEmployee(string login)
Vrátí zaměstnance, jehož uživatelské jméno odpovídá parametru login.
```

```
public void InsertOnSubmit(Employee employee)
Vytvoří nový záznam v tabulce Employee.
```

```
public void DeleteOnSubmit(Employee employee)
Odstraní záznam z tabulky Employee. Nejprve provede kontrolu, zda není zaměstnanec evidován jako odpovědná osoba majetku nebo není odpovědná osoba služby/výkonu nebo není evidován jako cvičitel. Při chybě vyvolá výjimku.
```

```
public bool ValidateLogin(string login, string password)
Ověří přihlašovací údaje.
```

```
public static string GetSHA1Hash(string input)
Vrátí SHA1 hash parametru input.
```

```
public void ChangePassword(string login, string password)
Nastaví heslo předané v parametru password u zaměstnance s uživatelským jménem login.
```

```
public bool CheckOtherManager(int excludeID)
Zkontroluje existenci jiného zaměstnance s rolí Manažer. Číslo v parametru excludeID určuje, který zaměstnanec má být z kontroly vyjmut.
```

Třída má vlastnost `public static Dictionary<string, string> Roles`, která vrátí seznam rolí používaných v aplikaci.

- **ServiceManager**

Odvozená třída od abstraktní třídy `BaseManager`. Poskytuje metody pro práci s entitou `Služba/výkon (ISCeZ.DAL.Service)`.

```
public IQueryable<Service> GetServices()
Vrátí seznam všech služeb/výkonů.
```

```

public IQueryable<Service> GetServices(DateTime date)
Vrátí seznam všech služeb/výkonů, které jsou platné k datu parametru
date.

public Service GetService(int serviceID)
Vrátí službu/výkon, jejíž ServiceID odpovídá parametru serviceID.

public static XDocument GetServicesAsXML(object services)
Vrátí seznam služeb/výkonů jako XML dokument.

public IQueryable<ServiceGroup> GetServiceGroup(int
serviceID)
Vrátí skupinu služeb/výkonů pro službu/výkon, jejíž ServiceID
odpovídá parametru serviceID.

public IQueryable<ServiceGroup> GetParentServices(int
childServiceID)
Vrátí seznam skupin služeb/výkonů, do kterých je zařazena sku-
pina/výkon jejíž serviceID odpovídá parametru childServiceID.

public IQueryable<Employee> GetGarants()
Vrátí seznam zaměstnanců, kteří mají přiřazenou roli Cvičitel.

public void InsertOnSubmit(Service service)
Vytvoří nový záznam v tabulce Service.

public void DeleteOnSubmit(Service service)
Odstraní záznam z tabulky Service. Nejprve provede kontrolu, zda
již nebyla služba/výkon naplánována nebo nepatří do skupiny slu-
žeb/výkonů. Při chybě vyvolá výjimku.

```

- ISCeZUser  
Instance této třídy představuje aktuálně přihlášeného uživatele. Pro kontrolu přidělených rolí obsahuje metodu `public bool HasRole(string role)`, která vrací `true/false` podle toho, zda má/nemá uživatel přidělenou roli odpovídající parametru `role`.

## 4.7. Prezentační vrstva (Presentation Layer)

Prezentační vrstva obsahuje třídy formulářů používaných v aplikaci. Třída `MainForm` představuje hlavní formulář aplikace. Je zde implementován návrhový vzor Singleton [3], který zajistí, aby třída měla pouze jednu instanci. Následuje abecední seznam tříd s popisem jejich významu:

- `AboutForm` - *O aplikaci*
- `AssetDetailForm` - *Majetková karta*
- `AssetsForm` - *Evidence majetku*

- `BillOfFareDetailForm` - *Návrh jídelníčku*
- `BillOfFareTemplateDetailForm` - *Návrh šablony jídelníčku*
- `BillOfFareTemplatesForm` - *Šablony jídelníčků*
- `ClientDetailForm` - *Údaje o klientovi*
- `ClientPlanAddForm` - *Plánovací kalendář*
- `ClientPlanBillForm` - *Účtenka*
- `ClientRegisterForm` - *Registr klientů*
- `ClientsForm` - *Evidence klientů*
- `DayPlanForm` - *Denní rozpis*
- `EmployeeDetailForm` - *Údaje o zaměstnanci*
- `EmployeesForm` - *Evidence zaměstnanců*
- `ChangePasswordForm` - *Změna hesla*
- `LoginForm` - *Přihlášení*
- `ServiceDetailForm` - *Údaje o službě/výkonu*
- `ServicesForm` - *Evidence služeb/výkonů*
- `TotalReportForm` - *Součtová sestava*

Třídy `AssetsForm`, `BillOfFareTemplatesForm`, `ClientsForm`, `EmployeesForm` a `ServicesForm` jsou odvozeny od třídy `ISCeZ.Controls.BaseListingForm`.

Třídy `AssetDetailForm`, `BillOfFareDetailForm`, `BillOfFareTemplateDetailForm`, `ClientDetailForm`, `ClientPlanAddForm`, `ClientPlanBillForm`, `EmployeeDetailForm` a `ServiceDetailForm` jsou odvozeny od třídy `ISCeZ.Controls.BaseDetailForm`.

V třídě `ClientPlanAddForm` je použit ovládací prvek `ISCeZ.Controls.ExtendedMonthCalendar`, který rozšiřuje standardní `MonthCalendar` o obsluhu události `DoubleClick`. Zdrojový kód si můžeme stáhnout na adrese <http://dotnet.mvps.org/dotnet/faqs/?id=monthcalendardoubleclick&lang=en>.



V třídě `AssetDetailForm` je použit ovládací prvek `ISCeZ.Controls.NullableDateTimePicker`, který rozšiřuje standardní `DateTimePicker` o možnost zobrazit prázdnou hodnotu. Zdrojový kód si můžeme stáhnout na adrese [http://www.codeproject.com/kb/selection/nullable\\_datetimepicker.aspx](http://www.codeproject.com/kb/selection/nullable_datetimepicker.aspx).

Ikony, použité na formulářích, byly staženy se serveru <http://download.chip.eu/cz>.

## 4.8. Databáze

Aplikace používá pro ukládání dat databázový stroj Microsoft SQL Server Compact 3.5. Soubor s databází `iscez.sdf` je uložen v adresáři `Data`. Uživatel, který aplikaci spustil, musí mít nastaveno oprávnění zápisu do tohoto souboru.

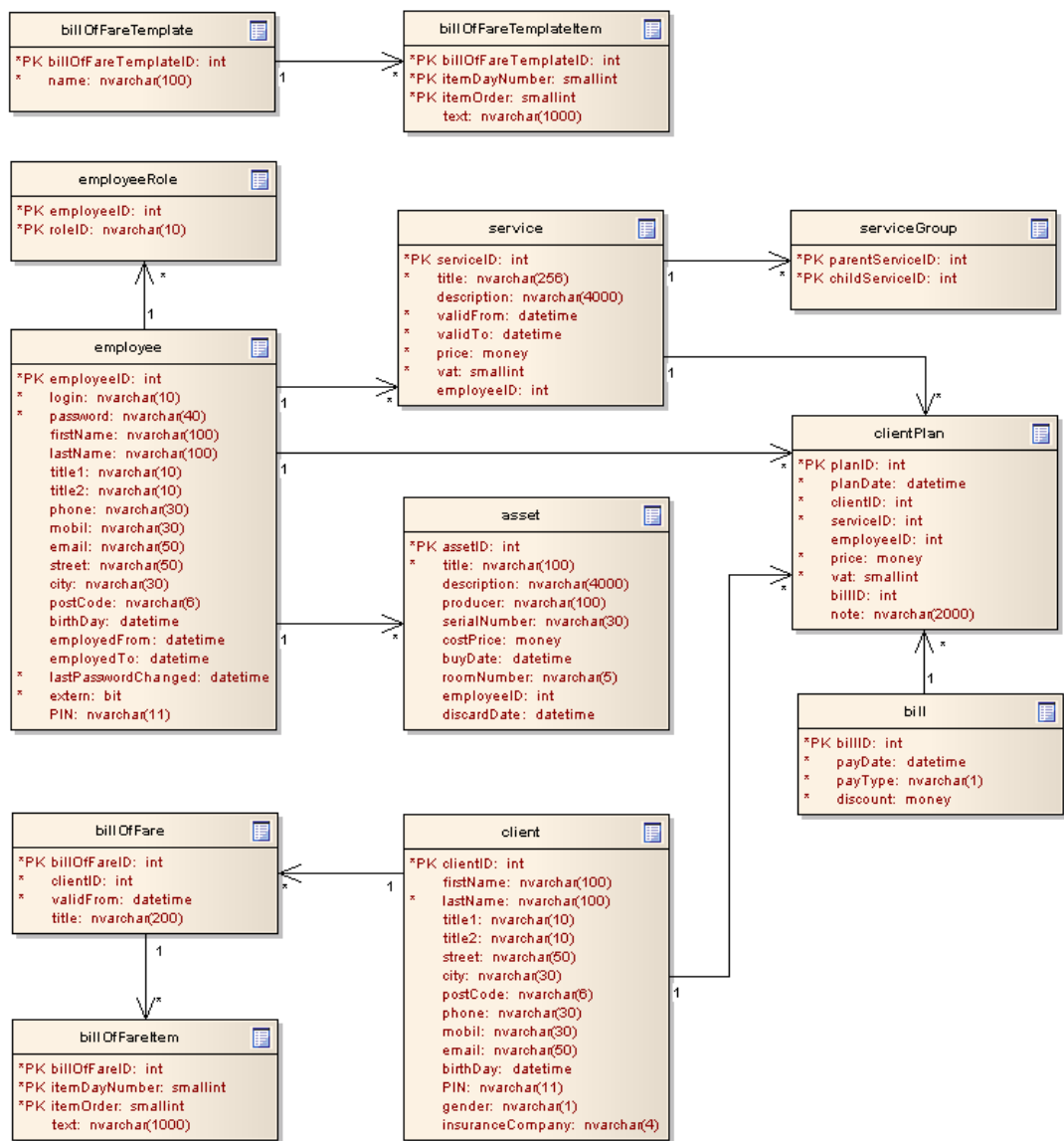
Na obrázku č.17. je zobrazen datový model včetně relací mezi tabulkami.

## 4.9. Tiskové výstupy

Všechny tiskové výstupy jsou nejdříve zobrazeny v okně webového prohlížeče. HTML stránka je vygenerována z XML dokumentu dodaného logickou vrstvou a z XSL šablony pomocí XSLT transformace. Šablony jsou uloženy v adresáři `Data`. Tento princip nám umožňuje jednoduše měnit vzhled tiskového výstupu bez nutnosti zásahu do kódu aplikace.

## 4.10. Instalátor

Instalační soubor `ISCeZ_setup.exe` byl vytvořen pomocí programu Inno Setup Compiler 5.3.10. [7] Pro kontrolu, zda-li je nainstalován Microsoft .NET Framework, ověřuje hodnotu `Install` v klíči registru `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NET Framework Setup\NDP\v3.5`. Pro kontrolu, zda-li je nainstalován Microsoft SQL Server Compact 3.5, ověřuje existenci klíče registru `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server Compact Edition\v3.5`.



Obrázek 17. Datový model

## Závěr

Výsledkem této bakalářské práce je nová aplikace sloužící soukromému subjektu, který poskytuje služby v oblasti zdravého životního stylu a volnočasových aktivit. Plně podporuje proces plánování čerpání služeb/výkonů klientům. Manažerovi umožňuje vést evidenci zaměstnanců, poskytovaných služeb/výkonů a majetku. Pro kontrolu hospodaření může použít součtovou sestavu.

Při návrhu uživatelského rozhraní jsem se snažil o jeho jednoduchost a snadné ovládání. Dle vyjádření uživatelů, kteří aplikaci testovali, je ovládání intuitivní a po přečtení uživatelské příručky snadno pochopitelné.

Nabízí se další možnosti pro rozšíření funkcí aplikace. Např. automatické rozesílání denních rozpisů cvičitelům elektronickou poštou nebo přidání detailnějších reportů pro manažera. Předpokládám další rozvoj aplikace na základě požadavků uživatelů.

Pro tvorbu aplikace jsem využil některé své dosavadní zkušenosti z praxe a rozšířil je o nové poznatky programování v jazyku C# a integrovaného jazyka pro dotazování LINQ.

## Reference

- [1] John Sharp: Microsoft Visual C# 2005 Krok za krokem, Computer Press, a.s., 2006, ISBN 80-251-1156-3
- [2] Paolo Pialorsi, Marco Russo: Microsoft LINQ Kompletní průvodce programátora, Computer Press, a.s., 2009, ISBN 978-80-251-2735-3
- [3] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Návrh programů pomocí vzorů, GRADA, 2003, ISBN 80-247-0302-5
- [4] Microsoft SQL Server Compact 3.5  
<http://www.microsoft.com/sqlserver/2005/en/us/compact.aspx>
- [5] XSLT Tutorial <http://www.w3schools.com/xsl/default.asp>
- [6] Microsoft Developer Network (MSDN) <http://www.msdn.com>
- [7] Inno Setup Compiler <http://www.innosetup.com>

## A. Popis obsahu příloženého CD

Součástí této práce je i CD. Jeho obsahem jsou tři adresáře a jeden soubor.

- Adresář `bin` obsahuje instalační soubor `ISCeZ_setup.exe`.
- Adresář `doc` obsahuje dokumentaci této práce ve spustitelné podobě a zdrojové kódy dokumentace v `TEXu`.
- Adresář `src` obsahuje zdrojový kód programu.
- V souboru `readme.txt` je popsáno, jak lze aplikaci spustit.