

**Jihočeská univerzita v Českých Budějovicích**  
**Pedagogická fakulta**

Katedra fyziky

**Realizace informačního systému**

bakalářská práce

Autor: **Robert Pokorný**  
Vedoucí bakalářské práce: **Ing. Michal Šerý**

České Budějovice 2006

## **Anotace**

Tato bakalářská práce se zabývá realizací informačního systému Telefonní seznam, který je postaven jako třívrstvá webová aplikace typu klient/server. Detailně popisuje hlavní ASP stránky umožňující vyhledávání. Všechny soubory tvořící virtuální adresář aplikace se nacházejí na příloženém kompaktním disku.

## **Synopsis**

This bachelor's thesis is aimed at the realization of the information system called Telephone Directory built as a three-layer client/server web application. The query main ASP pages have been described in detail. All the files that create the virtual directory of the application can be found on the enclosed compact disc.

*Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.*

*Robert Pokorný*

*Děkuji vedoucímu mé bakalářské práce panu Ing. Michalovi Šerému za odborné vedení, cenné rady a připomínky při vypracování mé práce.*

# Obsah

Úvod.....	7
<b>1 Rozbor požadavků zadání.....</b>	<b>8</b>
<b>1.1 Zadání.....</b>	<b>8</b>
<b>1.2 Rozbor.....</b>	<b>8</b>
<b>2 Návrh řešení zadání.....</b>	<b>9</b>
<b>3 Využití programových prostředků a vybavení.....</b>	<b>10</b>
<b>3.1 ASP.....</b>	<b>10</b>
<b>3.2 ADO.....</b>	<b>10</b>
<b>3.3 TSQL.....</b>	<b>10</b>
<b>3.4 VBScript.....</b>	<b>11</b>
<b>3.5 JavaScript, JScript.....</b>	<b>11</b>
<b>3.5.1 JavaScript.....</b>	<b>11</b>
<b>3.5.2 JScript.....</b>	<b>11</b>
<b>4 Praktická realizace zadání.....</b>	<b>12</b>
<b>4.1 Instalace MS SQL Server 2005 Express Edition SP1.....</b>	<b>12</b>
<b>4.2 Vytvoření databáze Telefonny.....</b>	<b>12</b>
<b>4.3 Vytvoření tabulek.....</b>	<b>12</b>
<b>4.3.1 TitutT.....</b>	<b>13</b>
<b>4.3.2 Utvar1T.....</b>	<b>13</b>
<b>4.3.3 Utvar2T.....</b>	<b>13</b>
<b>4.3.4 Utvar3T.....</b>	<b>14</b>
<b>4.3.5 Utvar4T.....</b>	<b>14</b>
<b>4.3.6 FunkceT.....</b>	<b>15</b>

<b>4.3.7</b>	<b>SeznamT.....</b>	<b>15</b>
<b>4.3.8</b>	<b>StatistikaT.....</b>	<b>16</b>
<b>4.3.9</b>	<b>Diagram struktury databáze.....</b>	<b>17</b>
<b>4.4</b>	<b>Instalace IIS.....</b>	<b>18</b>
<b>4.5</b>	<b>Konfigurace IIS.....</b>	<b>18</b>
<b>4.6</b>	<b>Vytvoření a popis hlavních souborů aplikace.....</b>	<b>18</b>
<b>4.6.1</b>	<b>Soubor global.asa.....</b>	<b>18</b>
<b>4.6.2</b>	<b>Souboru default.asp.....</b>	<b>19</b>
<b>4.6.3</b>	<b>Soubor dotaz.asp.....</b>	<b>33</b>
<b>4.6.4</b>	<b>Soubor aktual.asp.....</b>	<b>39</b>
<b>4.6.5</b>	<b>Soubor help.asp.....</b>	<b>39</b>
<b>4.6.6</b>	<b>Soubor styles.css.....</b>	<b>39</b>
<b>4.6.7</b>	<b>Soubor stylesp.css.....</b>	<b>39</b>
<b>4.6.8</b>	<b>Soubor Include\connection.asp.....</b>	<b>40</b>
<b>4.7</b>	<b>Popis souborů pro editaci záznamů.....</b>	<b>40</b>
<b>4.7.1</b>	<b>Soubor default.asp.....</b>	<b>40</b>
<b>4.7.2</b>	<b>Soubor dotaz.asp.....</b>	<b>41</b>
<b>4.7.3</b>	<b>Soubor insert.asp.....</b>	<b>41</b>
<b>4.7.4</b>	<b>Soubor update.asp.....</b>	<b>41</b>
<b>4.7.5</b>	<b>Soubor delete.asp.....</b>	<b>42</b>
<b>5</b>	<b>Parametry vytvořeného systému.....</b>	<b>43</b>
<b>6</b>	<b>Zhodnocení.....</b>	<b>44</b>
	<b>Literatura.....</b>	<b>45</b>

## Úvod

Informace a přístup k nim je jedním z hlavních aspektů dnešní doby. Přístup k potřebným údajům, jejich efektivní využívání a sdílení v rámci organizace může být i v dnešní době problematické a neefektivní.

Aktuálnost dat statických webových stránek nebo telefonního seznamu v tištěné podobě lze jen obtížně zabezpečit.

Proto i telefonní seznam v organizaci s poměrně složitou strukturou, častými personálními změnami a reorganizacemi je vhodné řešit jako moderní informační systém.

Takovýto informační systém by měl být uživatelsky jednoduchý, snadno ovladatelný, bez nutnosti instalace na PC uživatele.

Měl by umožňovat vyhledávání dle různých kritérií bez nutnosti jejich specifikace, prostým vyplněním známých údajů nebo jejich částí.

Zobrazení požadovaných výsledků hledání by mělo být stručné, přehledné a úplné.

Systém by měl být pravidelně aktualizován a centrálně spravován, aby poskytoval uživatelům přesné a aktuální údaje.

# 1 Rozbor požadavků zadání

## 1.1 Zadání

Vytvořit IS Telefonní seznam umožňující vyhledávání záznamů zadáním minimálně jedné z těchto položek:

- příjmení
- jméno
- číslo telefonu nebo faxu
- útvar (ve čtyřech úrovních)
- funkce

Vyhledají se všechny záznamy, jejichž položky současně začínají těmi zadanými a počet takto zobrazených záznamů bude omezen na 200.

Dále bude IS Telefonní seznam kromě výše uvedených obsahovat položky

- titul
- emailová adresa

Součástí IS Telefonní seznam bude možnost centrálně spravovat tento IS pro uživatele s oprávněním provádět změny.

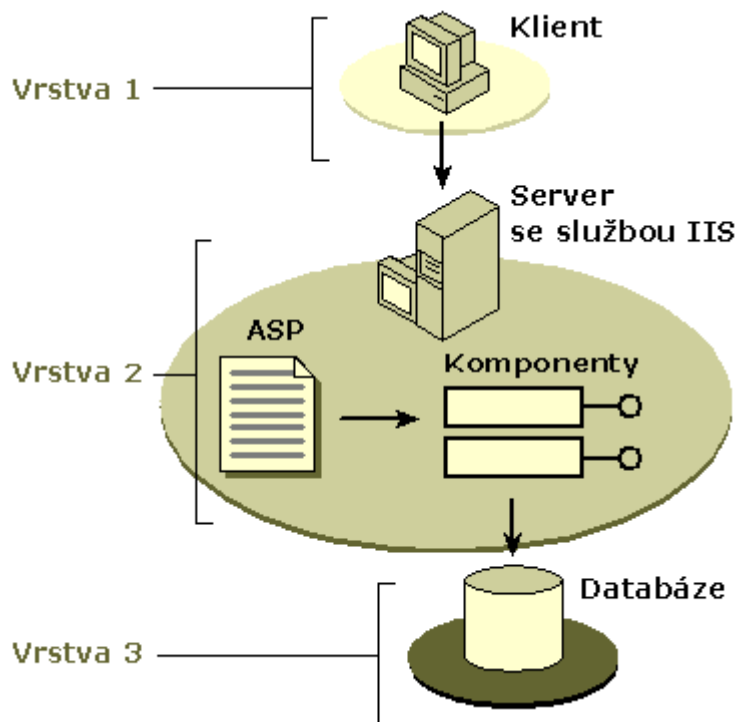
## 1.2 Rozbor

Požadavky na IS Telefonní seznam:

- snadné nasazení bez nutnosti instalace na jednotlivá PC
- snadná obsluha
- zadávání dat pro vyhledávání do formuláře
- kontrola zadaných hodnot na straně uživatele
- položky útvar a funkce realizovat výběrem ze seznamu
- seznam útvarů 2. až 4. úrovně závislý na výběru v předchozí úrovni
- data uložená ve vhodné databázi

## 2 Návrh řešení zadání

Na základě rozboru požadavků jsem se rozhodl vytvořit informační systém postavený na základě technologií firmy Microsoft jako třívrstvou webovou aplikaci typu klient/server viz obrázek [8].



Vrstva 1, uživatelské rozhraní a navigace. Bude realizována prohlížečem Microsoft Internet Explorer, který umožní uživatelům nejen pracovat s aplikací, zadávat data a zobrazovat výsledky požadavků, ale také manipulovat s daty.

Vrstva 2, obchodní logika, spojující uživatele na jedné straně a data na straně druhé. Bude postavena na Microsoft Windows Serveru s Internetovou informační službou (IIS) se skriptovacím prostředím Active Server Pages (ASP) pro generování dynamického obsahu webových stránek.

Vrstva 3, databázové služby a přístup k datům aplikace. Bude postavena na Microsoft SQL Serveru.

## Využití programových prostředků a vybavení

### 2.1 ASP

**ASP** je akronym ze slov **A**ctive **S**erver **P**ages. Jedná se o webové orientovanou technologii firmy Microsoft pro tvorbu dynamických stránek na straně serveru.

Instaluje se automaticky s **IIS** (**I**nternet **I**nformation **S**ervices) na MS Windows NT 4 nebo novějších respektive s **PWS** (**P**ersonal **W**eb **S**erver) na MS Windows 95 nebo novějších. PWS je menší avšak plně funkční verzí IIS.

ASP soubor je textový soubor, který má příponu .asp. Může obsahovat nejen text, HTML, XML ale i příkazy psané ve skriptovacích jazycích. Je prováděn na straně serveru.

Pokud prohlížeč požaduje ASP stránku, IIS předá požadavek ASP (asp.dll). ASP čte soubor řádek po řádku a provádí všechny skriptovací příkazy dané stránky. Nakonec vygeneruje HTML dokument, který je poslán zpět prohlížeči. Poskytuje tedy jistou úroveň zabezpečení, neboť ASP kód nemůže být zobrazen prohlížečem.

ASP umožňuje dynamicky upravovat, měnit obsah webových stránek a přizpůsobovat jejich obsah konkrétním uživatelům. Zpřístupňovat data nebo databáze a odpovídat na dotazy nebo data odeslaná z HTML formulářů.

ASP serverový skript může být vymezen buď dvojicemi znaků `<% %>` nebo elementu `<SCRIPT RUNAT="SERVER"> </SCRIPT>`.

V ASP je možné používat několik skriptovacích jazyků. Automaticky jsou nainstalovány VBScript a JScript. Přednastaveným skriptovacím jazykem je VBScript. Deklarace `<%@LANGUAGE="název jazyka"%>` musí být uvedena na prvním řádku souboru.

Další skriptovací jazyky, jako např. PerlScript nebo Python, je však nutné doinstalovat.

### 2.2 ADO

**ADO** je akronym ze slov **A**ctiveX **D**ata **O**bjects. ADO je Microsoft ActiveX komponenta, umožňující klientské aplikaci přistupovat a manipulovat s daty databázového serveru prostřednictvím poskytovatele **OLE DB**.

Je automaticky instalováno s **IIS**.

### 2.3 TSQL

SQL je akronym ze slov **S**tructured **Q**uery **L**anguage. **Transact-SQL** je implementací tohoto jazyka produktem SQL Server společnosti Microsoft.

Transact-SQL je jazyk, který se používá k administraci databázové jednotky SQL Serveru, k vytváření a správě databázových objektů (jazyk

DDL, Data Definition Language) a ke vkládání, získávání, modifikování a mazání dat (jazyk DML, Data Manipulation Language). Transact-SQL je rozšířením jazyka definovaného v SQL standardech a publikovaného ISO (International Standards Organization) a ANSI (American National Standards Institute).

## 2.4 VBScript

VBScript, Microsoft **V**isual **B**asic **S**cripting Edition, je skriptovací jazyk, který přináší možnost používat VBScript jak na straně webového klienta v Microsoft Internet Exploreru, tak i na straně webového serveru v Microsoft Internet Information Service a dalších prostředích.

VBScript je zjednodušenou verzí programovacího jazyka Microsoft Visual Basic. Je snadné se ho naučit a používat.

Pokud je do HTML dokumentu vložen VBScript, prohlížeč čte HTML a interpretuje VBScript. Ten může být proveden okamžitě nebo při pozdější události.

## 2.5 JavaScript, JScript

### 2.5.1 JavaScript

JavaScript je skriptovací jazyk vyvinutý firmou Netscape Communications určený pro vývoj klientských a serverových internetových aplikací.

Netscape Navigator interpretuje JavaScript vložený do webových stránek. Ten může být proveden okamžitě nebo při pozdější události.

JavaScript je nezávislý na jazyku Java firmy Sun Microsystems.

### 2.5.2 JScript

Microsoft JScript je otevřenou implementací JavaScriptu firmy Netscape Communications. JScript je výkonný skriptovací jazyk určený k vytváření aktivního obsahu pro Internet. JScript umožňuje vývojářům spojovat a automatizovat množství objektů ve webových stránkách včetně ActiveX ovládacích prvků a Java programů.

Microsoft Internet Explorer interpretuje JScript vložený do webových stránek. Ten může být proveden okamžitě nebo při pozdější události.

## 3 Praktická realizace zadání

### 3.1 Instalace MS SQL Server 2005 Express Edition SP1

Instalace MS SQL Server 2005 Express Edition SP1 se provádí spuštěním souboru SQLEXPRESS.EXE, který je volně ke stažení z webových stránek firmy Microsoft.

Ke správě MS SQL Server 2005 Express Edition slouží SQL Server Management Studio Express. Instalace se provádí spuštěním souboru SQLServer2005\_SSMSEE.msi, který je volně ke stažení z webových stránek firmy Microsoft.

### 3.2 Vytvoření databáze Telefoný

Skript vytvoří databázi `Telefony`, určuje umístění souborů, jejich počáteční velikost, automatický přírůstek a kolaci (Case Insensitive and Accent Sensitive).

```
CREATE DATABASE [Telefony] ON PRIMARY
(
    NAME = N'Telefony', FILENAME = N'C:\Program Files\Microsoft SQL
    Server\MSSQL.1\MSSQL\DATA\Telefony.mdf', SIZE = 3072KB, FILEGROWTH =
    1024KB
)
LOG ON
(
    NAME = N'Telefony_log', FILENAME = N'C:\Program Files\Microsoft SQL
    Server\MSSQL.1\MSSQL\DATA\Telefony_log.ldf', SIZE = 1024KB,
    FILEGROWTH = 10%
)
COLLATE SQL_Czech_CP1250_CI_AS
```

### 3.3 Vytvoření tabulek

Tabulky v databázi mohou být vzájemně svázány klíči. Primární klíč je sloupec s unikátní hodnotou pro každý řádek tabulky. Důvodem je propojit data napříč tabulkami bez nutnosti data opakovat.

Tabulky mohou obsahovat indexy. Vytvářejí se proto, aby bylo možné najít řádky rychleji a efektivněji. Aktualizace tabulky obsahující indexy trvá však déle než tabulky bez indexů.

Unikátní index navíc znamená, že dva řádky nemohou mít stejnou indexovou hodnotu.

Identity vlastnost vytváří identity sloupec v tabulce.

Omezení (Constraint) definují způsob, kterým databázová jednotka automaticky vynucuje integritu databáze.

### 3.3.1 TitulT

Skript vytvoří tabulku `TitulT` pro uložení titulů. Tabulka obsahuje `IDENTITY` sloupec `Id` typu `smallint`, který je primárním klíčem tabulky s primárním indexem. Dále obsahuje sloupec `Titul` typu `varchar` pro uložení hodnot titulů.

```
USE [Telefony]
CREATE TABLE [dbo].[TitulT]
(
  [Id] [smallint] IDENTITY(1,1) NOT NULL,
  [Titul] [varchar](50) NOT NULL,
  CONSTRAINT [PK_TitulT] PRIMARY KEY CLUSTERED
(
  [Id] ASC
)
WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

### 3.3.2 Utvar1T

Skript vytvoří tabulku `Utvar1T` pro uložení názvů a kódů 1. úrovně útvarů.

Tabulka obsahuje identity sloupec `Id1`, typu `smallint`, který je primárním klíčem tabulky s primárním indexem. Dále obsahuje sloupec `Utvar1`, typu `varchar`, pro uložení hodnot názvů 1. úrovně útvarů a sloupec `U1`, typu `char`, s kódem 1. úrovně útvarů.

```
CREATE TABLE [dbo].[Utvar1T]
(
  [Id1] [smallint] IDENTITY(1,1) NOT NULL,
  [Utvar1] [varchar](255) NOT NULL,
  [U1] [char](1) NOT NULL,
  CONSTRAINT [PK_Utvar1T] PRIMARY KEY CLUSTERED
(
  [Id1] ASC
) WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

### 3.3.3 Utvar2T

Skript vytvoří tabulku `Utvar2T` pro uložení názvů a kódů 2. úrovně útvarů s vazbou na nadřazený útvar.

Tabulka obsahuje `IDENTITY` sloupec `Id2`, typu `smallint`, který je primárním klíčem tabulky s primárním indexem. Dále obsahuje sloupec `Utvar2`, typu `varchar`, pro uložení hodnot názvů 2. úrovně útvarů, sloupec `Id1`, typu `smallint`, pro vazbu na nadřazený útvar, sloupec `U2`, typu `char`, s kódem 2. úrovně útvarů sloupec a sloupec `U1`, typu `char`, s kódem 1. úrovně útvarů.

```

CREATE TABLE [dbo].[Utvvar2T]
(
  [Id2] [smallint] IDENTITY(1,1) NOT NULL,
  [Id1] [smallint] NOT NULL,
  [Utvvar2] [varchar](255) NOT NULL,
  [U2] [char](1) NULL,
  [U1] [char](1) NULL,
  CONSTRAINT [PK_Utvvar2T] PRIMARY KEY CLUSTERED
  (
    [Id2] ASC
  ) WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

```

### 3.3.4 Utvar3T

Skript vytvoří tabulku `Utvvar3T` pro uložení názvů a kódů 3. úrovně útvarů s vazbou na nadřazený útvar.

Tabulka obsahuje `IDENTITY` sloupec `Id3`, typu `smallint`, který je primárním klíčem tabulky s primárním indexem. Dále obsahuje sloupec `Utvvar3`, typu `varchar`, pro uložení hodnot názvů 3. úrovně útvarů, sloupec `Id2`, typu `smallint`, pro vazbu na nadřazený útvar, sloupec `U3`, typu `char`, s kódem 3. úrovně útvarů sloupec a sloupec `U2`, typu `char`, s kódem 2. úrovně útvarů.

```

CREATE TABLE [dbo].[Utvvar3T]
(
  [Id3] [smallint] IDENTITY(1,1) NOT NULL,
  [Id2] [smallint] NOT NULL,
  [Utvvar3] [varchar](255) NOT NULL,
  [U3] [char](1) NULL,
  [U2] [char](1) NULL,
  CONSTRAINT [PK_Utvvar3T] PRIMARY KEY CLUSTERED
  (
    [Id3] ASC
  ) WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

```

### 3.3.5 Utvar4T

Skript vytvoří tabulku `Utvvar4T` pro uložení názvů a kódů 4. úrovně útvarů s vazbou na nadřazený útvar.

Tabulka obsahuje `IDENTITY` sloupec `Id4`, typu `smallint`, který je primárním klíčem tabulky s primárním indexem. Dále obsahuje sloupec `Utvvar4`, typu `varchar`, pro uložení hodnot názvů 4. úrovně útvarů, sloupec `Id3`, typu `smallint`, pro vazbu na nadřazený útvar, sloupec `U4`, typu `char`, s kódem 4. úrovně útvarů sloupec a sloupec `U3`, typu `char`, s kódem 3. úrovně útvarů.

```

CREATE TABLE [dbo].[Utvar4T]
(
  [Id4] [smallint] IDENTITY(1,1) NOT NULL,
  [Id3] [smallint] NOT NULL,
  [Utvar4] [varchar](255) NOT NULL,
  [U4] [char](1) NULL,
  [U3] [char](1) NULL,
  CONSTRAINT [PK_Utvar4T] PRIMARY KEY CLUSTERED
  (
    [Id4] ASC
  ) WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

```

### 3.3.6 FunkceT

Skript vytvoří tabulku `FunkceT` pro uložení názvů funkcí.

Tabulka obsahuje `IDENTITY` sloupec `Id`, typu `smallint`, který je primárním klíčem tabulky s primárním indexem. Dále obsahuje sloupec `Funkce`, typu `varchar`, pro uložení hodnot názvů funkcí.

```

CREATE TABLE [dbo].[FunkceT]
(
  [Id] [smallint] IDENTITY(1,1) NOT NULL,
  [Funkce] [varchar](255) NOT NULL,
  CONSTRAINT [PK_FunkceT] PRIMARY KEY CLUSTERED
  (
    [Id] ASC
  )
  WITH (PAD_INDEX = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

```

### 3.3.7 SeznamT

Skript vytvoří tabulku `SeznamT` pro uložení záznamů IS Telefonní seznam.

Tabulka obsahuje `IDENTITY` sloupec `Id`, typu `int`, sloupec `Prijmeni`, typu `varchar`, s unikátním indexem, pro uložení hodnot příjmení, sloupec `Jmeno`, typu `varchar` s indexem, pro uložení hodnot jmen, sloupec `TitulId`, typu `smallint`, který je cizím klíčem k propojení s tabulkou `TitulT`, sloupec `Tel`, typu `int` s indexem, pro uložení hodnot telefonních čísel, sloupec `Fax`, typu `int` s indexem, pro uložení hodnot faxových čísel, sloupec `Email`, typu `varchar`, pro uložení hodnot adres elektronické pošty, sloupec `KUIId1`, typu `smallint` s indexem, který je cizím klíčem k propojení s tabulkou `Utvar1T`, sloupec `KUIId2`, typu `smallint` s indexem, který je cizím klíčem k propojení s tabulkou `Utvar2T`, sloupec `KUIId3`, typu `smallint` s indexem, který je cizím klíčem k propojení s tabulkou `Utvar3T`, sloupec `KUIId4`, typu `smallint` s indexem, který je cizím klíčem k propojení s tabulkou `Utvar4T`, sloupec `FunkceId`, typu `smallint` s indexem, který je cizím klíčem k propojení s tabulkou `FunkceT`.

```

CREATE TABLE [dbo].[SeznamT]
(
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Prijmeni] [varchar](255) NULL,
    [Jmeno] [varchar](255) NULL,
    [TitulId] [smallint] NULL,
    [Tel] [int] NOT NULL,
    [Fax] [int] NULL,
    [Email] [varchar](255) NULL,
    [KUIId1] [smallint] NOT NULL,
    [KUIId2] [smallint] NULL,
    [KUIId3] [smallint] NULL,
    [KUIId4] [smallint] NULL,
    [FunkceId] [smallint] NOT NULL
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[SeznamT] WITH NOCHECK ADD CONSTRAINT
[FK_SeznamT_FunkceT] FOREIGN KEY([FunkceId])
REFERENCES [dbo].[FunkceT] ([Id])
GO
ALTER TABLE [dbo].[SeznamT] CHECK CONSTRAINT [FK_SeznamT_FunkceT]
GO
ALTER TABLE [dbo].[SeznamT] WITH NOCHECK ADD CONSTRAINT
[FK_SeznamT_TitulT] FOREIGN KEY([TitulId])
REFERENCES [dbo].[TitulT] ([Id])
GO
ALTER TABLE [dbo].[SeznamT] CHECK CONSTRAINT [FK_SeznamT_TitulT]
GO
ALTER TABLE [dbo].[SeznamT] WITH NOCHECK ADD CONSTRAINT
[FK_SeznamT_Utvar1T] FOREIGN KEY([KUIId1])
REFERENCES [dbo].[Utvar1T] ([Id1])
GO
ALTER TABLE [dbo].[SeznamT] CHECK CONSTRAINT [FK_SeznamT_Utvar1T]
GO
ALTER TABLE [dbo].[SeznamT] WITH NOCHECK ADD CONSTRAINT
[FK_SeznamT_Utvar2T] FOREIGN KEY([KUIId2])
REFERENCES [dbo].[Utvar2T] ([Id2])
GO
ALTER TABLE [dbo].[SeznamT] CHECK CONSTRAINT [FK_SeznamT_Utvar2T]
GO
ALTER TABLE [dbo].[SeznamT] WITH NOCHECK ADD CONSTRAINT
[FK_SeznamT_Utvar3T] FOREIGN KEY([KUIId3])
REFERENCES [dbo].[Utvar3T] ([Id3])
GO
ALTER TABLE [dbo].[SeznamT] CHECK CONSTRAINT [FK_SeznamT_Utvar3T]
GO
ALTER TABLE [dbo].[SeznamT] WITH NOCHECK ADD CONSTRAINT
[FK_SeznamT_Utvar4T] FOREIGN KEY([KUIId4])
REFERENCES [dbo].[Utvar4T] ([Id4])
GO
ALTER TABLE [dbo].[SeznamT] CHECK CONSTRAINT [FK_SeznamT_Utvar4T]

```

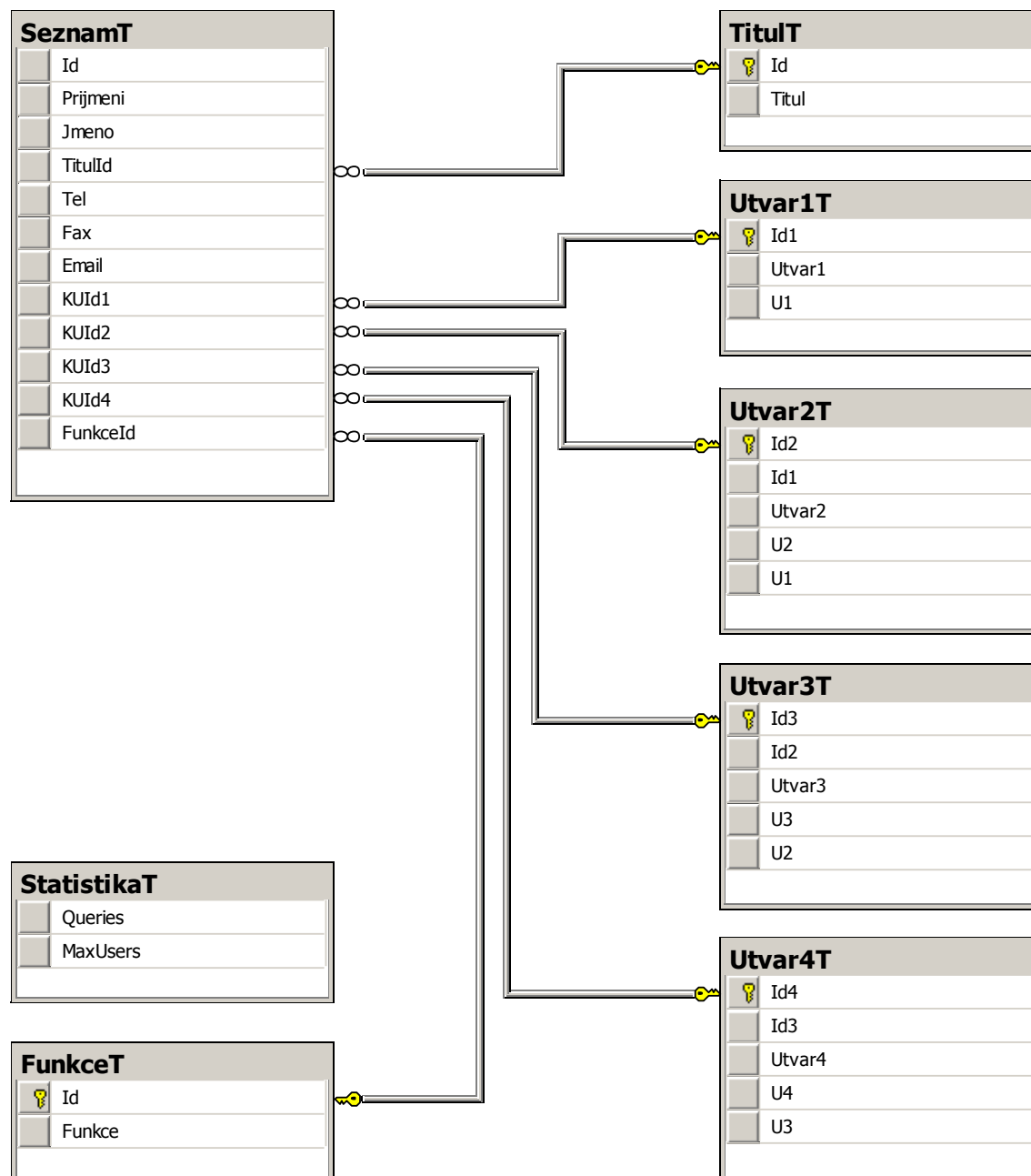
### 3.3.8 StatistikaT

Skript vytvoří tabulku StatistikaT pro uložení statistických údajů.

Tabulka obsahuje sloupec `Queries`, typu `int`, pro uložení hodnoty počtu dotazů a sloupec `MaxUsers`, typu `int`, pro uložení maximální hodnoty současně přihlášených uživatelů.

```
CREATE TABLE [dbo].[StatistikaT]
(
    [Queries] [int] NOT NULL,
    [MaxUsers] [int] NOT NULL
) ON [PRIMARY]
```

### 3.3.9 Diagram struktury databáze



Tabulky `TitulT`, `Utvar1T`, `Utvar2T`, `Utvar3T`, `Utvar4T` a `FunkceT` obsahují primární klíče pomocí kterých jsou tyto tabulky svázány s cizími klíči tabulky `SeznamT` viz diagram.

### 3.4 Instalace IIS

Instalace IIS je nutné provést dle operačního systému. Například ve Windows XP Professional se provede z Ovládacích panelů, Přidat nebo odebrat programy, Přidat nebo odebrat součásti systému a zvolením položky Internetová informační služba.

### 3.5 Konfigurace IIS

Konfigurace IIS se provádí prostřednictvím Microsoft Management Console spuštěním Ovládací panely, Nástroje pro správu, Internetová informační služba.

### 3.6 Vytvoření a popis hlavních souborů aplikace

#### 3.6.1 Soubor `global.asa`

Soubor `global.asa` je volitelný soubor, který může obsahovat deklarace objektů, proměnných a metod, které jsou přístupné každé stránce aplikace.

Událost **`Application_OnStart`** nastane, jestliže první uživatel otevře první stránku ASP aplikace. Tato událost nastane po restartování webového serveru nebo po editaci souboru `global.asa`.

Událost **`Session_OnStart`** nastane vždy, když nový uživatel požaduje svou první stránku ASP aplikace.

Událost **`Session_OnEnd`** nastane vždy, pokud uživatel nepožaduje stránku po určitou dobu (defaultně 20 minut).

Událost **`Application_OnEnd`** nastane jestliže poslední uživatel ukončí session. Tato událost nastane po zastavení webového serveru nebo po editaci souboru `global.asa`.

Metoda objektu `Application` **`Lock`** umožní přístup k ASP objektu `Application` pouze aktuální ASP stránce, aby nedošlo k poškození obsahu při současném přístupu více uživatelů.

Metoda objektu `Application` **`Unlock`** ruší výhradní přístup k ASP objektu `Application` aktuální ASP stránce.

Událost `Application_OnStart` vynuluje `application` proměnnou `ActualUser`. Tato proměnná uchovává hodnotu aktuálního počtu uživatelů. Je inkrementována při události `Session_OnStart` a dekrementována při události `Session_OnEnd`.

Událost `Session_OnStart` také nastavuje session proměnnou `MaxC` na hodnotu `200`. Tato proměnná slouží k omezení počtu zobrazených záznamů při provádění dotazu.

Viz výpis souboru `global.asa`.

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
Sub Application_OnStart
    Application.Lock
    Application("ActualUser")=0
    Application.Unlock
End Sub

Sub Session_OnStart
    Session("MaxC")=200
    Application.Lock
    Application("ActualUser")=Application("ActualUser")+1
    Application.Unlock
End Sub

SUB Session_OnEnd
    Application.Lock
    Application("ActualUser")=Application("ActualUser")-1
    Application.Unlock
End Sub
</script>
```

### 3.6.2 Soubor `default.asp`

Soubor `default.asp` je hlavním souborem IS Telefonní seznam. Jeho základem je v HTML tabulce, v HTML tabulce, vložený formulář s ovládacími tlačítky.

Formulář obsahuje tři vstupní textová pole pro zadání maximálně 25 znaků příjmení, maximálně 25 znaků jména a 9 číslic telefonního nebo faxového čísla. Dále obsahuje pět rozevíracích seznamů, čtyři pro výběr útvaru ve čtyřech úrovních a jeden pro výběr funkce, s pěti skrytými vstupními poli pro uchování jejich hodnot. Pomocí vlastnosti `accesskey` jsou definovány klávesové zkratky pro snadný přístup k jednotlivým prvkům formuláře.

Horní čtveřice tlačítek slouží k přechodu na telefonní seznamy Obvodních ředitelství Praha I – Praha IV.

Dolní čtveřice slouží k zobrazení nápovědy, k zobrazení informací o aktualizaci, k vynulování formuláře a k vyhledání záznamů. Pomocí vlastnosti `accesskey` jsou definovány klávesové zkratky pro snadný přístup k některým tlačítkům.

Při odeslání formuláře jsou vstupní údaje předány stránce `dotaz.asp`.



```

<td>
<select name="U1" size="3" onChange="InitU2();"
onkeypress="CheckKey1();" onmouseup="OnMouseUpU1()"
ondblclick="Submit1();" accesskey="u">
</select>
</td></tr>

<tr><td>
<select name="U2" size="3" onChange="InitU3();"
onkeypress="CheckKey1();" onmouseup="OnMouseUpU2()"
ondblclick="Submit1();" accesskey="u">
</select>
</td></tr>

<tr><td>
<select name="U3" size="3" onChange="InitU4();"
onkeypress="CheckKey1();" onmouseup="OnMouseUpU3()"
ondblclick="Submit1();" accesskey="u">
</select>
</td></tr>

<tr><td>
<select name="U4" size="3" onChange="" onkeypress="CheckKey1();"
onmouseup="OnMouseUpU4()" ondblclick="Submit1();" accesskey="u">
</select>
</td></tr>

<tr><th valign="top"><span style="position: relative; top:
3px;"><u>F</u>unkce:</span></th><td>
<select name="F" size="3" onkeypress="CheckKey1();"
onmouseup="OnMouseUpF()" ondblclick="Submit1();" accesskey="f">
</select>
</td></tr>

<input type="hidden" name="I1" value="-1">
<input type="hidden" name="I2" value="-1">
<input type="hidden" name="I3" value="-1">
<input type="hidden" name="I4" value="-1">
<input type="hidden" name="I5" value="-1">
</table>
</td>

<td bgcolor="white" valign="top">
&nbsp;<span class="cite">Telefonní seznamy:</span><br>
&nbsp;<button
onclick="window.location='http://www.p11.aa/telefony/tel.asp'"
tabindex="-1">OŘ Praha I</button>&nbsp;<br>
&nbsp;<button
onclick="window.location='http://www.p12.aa/telefony/telefon.htm'"
tabindex="-1">OŘ Praha II</button>&nbsp;<br>
&nbsp;<button
onclick="window.location='http://www.p13.aa/index.php?page=telefony'"
tabindex="-1">OŘ Praha III</button>&nbsp;<br>
&nbsp;<button
onclick="window.location='http://www.p14.aa/or_praha14/Telefony/telefo
n_OR.asp'" tabindex="-1">OŘ Praha IV</button>&nbsp;<br>
</td>
</tr>

```

```

<tr>
<td bgcolor="white" valign="bottom">
    <button name="B4" onclick="Help1();" tabindex="-
1">Nápověda</button><br>
    <button name="B3" accesskey="a"
onclick="window.open('aktual.asp');" tabindex="-
1"><u>A</u>ktualizace</button><br>
    <button name="B2" accesskey="n"
onclick="InitF1();">Vy<u>n</u>ulovat</button><br>
    <button name="B1" accesskey="h"
onclick="Submit1();"><b>Vy<u>h</u>ledat</b></button><br>
</td>
</tr>
</table>
</form>

```

Rozevírací seznamy pro výběr útvarů a funkce je potřeba naplnit názvy útvarů a funkcí z příslušných tabulek databáze.

Kód v jazyce JavaScript deklaruje proměnné `U1`, `U2`, `U3` a `U4` typu `Array()` pro jednotlivé úrovně útvarů a proměnnou `F` typu `Array()` pro funkce.

Funkce `I1 (Name, Id1)` a `I2 (Name, Id1, Id2)` slouží k inicializaci těchto proměnných.

```
<script language="JavaScript">
```

```

var U1=new Array();
var U2=new Array();
var U3=new Array();
var U4=new Array();
var F=new Array();

function I1 (Name, Id1)
{
  this.Name=Name;
  this.Id1=Id1;
}

function I2 (Name, Id1, Id2)
{
  this.Name=Name;
  this.Id1=Id1;
  this.Id2=Id2;
}

```

Skript, prováděný na straně serveru, vytvoří ADO objekty `Command` a `Recordset`, naplní proměnnou `sql` SQL příkazem a ten provede. Získané záznamy z tabulky `Utvar1T` uloží v objektu `Recordset`.

Ze získaných záznamů se vytvoří příkazy v jazyce JavaScript, které, na straně klienta, inicializují proměnnou U1. Na závěr je objekt `Recordset` uzavřen.

```
<%
Set cmd=Server.CreateObject("ADODB.Command")
Set rs=Server.CreateObject("ADODB.Recordset")
sql="select utvar1, id1 from utvar1t order by utvar1"
cmd.ActiveConnection=strConn
cmd.CommandText=sql
Set rs=cmd.Execute

i=0
do until rs.EOF
Response.Write("U1[" & i & "]=new I1("" & rs("utvar1") & "", "" &
rs("id1") & "");" & vbCr)
i=i+1
rs.MoveNext
loop
rs.close
```

Skript, prováděný na straně serveru, naplní proměnnou `sql` SQL příkazem a ten provede. Získané záznamy z tabulky `Utvar2T` uloží v objektu `Recordset`.

Ze získaných záznamů se vytvoří příkazy v jazyce JavaScript, které, na straně klienta, inicializují proměnnou U2. Na závěr je objekt `Recordset` uzavřen.

```
sql="select utvar2, id1, id2 from utvar2t order by utvar2"
cmd.ActiveConnection=strConn
cmd.CommandText=sql
Set rs=cmd.Execute

i=0
do until rs.EOF
Response.Write("U2[" & i & "]=new I2("" & rs("utvar2") & "", "" &
rs("id1") & "", "" & rs("id2") & "");" & vbCr)
i=i+1
rs.MoveNext
loop
rs.close
```

Skript, prováděný na straně serveru, naplní proměnnou `sql` SQL příkazem a ten provede. Získané záznamy z tabulky `Utvar3T` uloží v objektu `Recordset`.

Ze získaných záznamů se vytvoří příkazy v jazyce JavaScript, které, na straně klienta, inicializují proměnnou U3. Na závěr je objekt `Recordset` uzavřen.

```

sql="select utvar3, id2, id3 from utvar3t order by utvar3"
cmd.ActiveConnection=strConn
cmd.CommandText=sql
Set rs=cmd.Execute

i=0
do until rs.EOF
Response.Write("U3[" & i & "]=new I2(" & rs("utvar3") & ", " &
rs("id2") & ", " & rs("id3") & ");" & vbCr)
i=i+1
rs.MoveNext
loop
rs.close

```

Skript, prováděný na straně serveru, naplní proměnnou `sql` SQL příkazem a ten provede. Získané záznamy z tabulky `Utvar4T` uloží v objektu `Recordset`.

Ze získaných záznamů se vytvoří příkazy v jazyce JavaScript, které, na straně klienta, inicializují proměnnou `U4`. Na závěr je objekt `Recordset` uzavřen.

```

sql="select utvar4, id3, id4 from utvar4t order by utvar4"
cmd.ActiveConnection=strConn
cmd.CommandText=sql
Set rs=cmd.Execute

i=0
do until rs.EOF
Response.Write("U4[" & i & "]=new I2(" & rs("utvar4") & ", " &
rs("id3") & ", " & rs("id4") & ");" & vbCr)
i=i+1
rs.MoveNext
loop
rs.close

```

Skript, prováděný na straně serveru, naplní proměnnou `sql` SQL příkazem a ten provede. Získané záznamy z tabulky `FunkceT` uloží v objektu `Recordset`.

Ze získaných záznamů se vytvoří příkazy v jazyce JavaScript, které, na straně klienta, inicializují proměnnou `F`. Na závěr je objekt `Recordset` uzavřen.

```

sql="select funkce, id from funkcet order by funkce"
cmd.ActiveConnection=strConn
cmd.CommandText=sql
Set rs=cmd.Execute

i=0
do until rs.EOF

```

```

Response.Write("F[" & i & "]=new I1('' & rs("funkce") & '', '' &
rs("id") & '');"& vbCr)
i=i+1
rs.MoveNext
loop
rs.close

```

Skript, prováděný na straně serveru, naplní proměnnou `sql` SQL příkazem a ten provede. Získaný záznam z tabulky `StatistikaT` uloží v objektu `Recordset`.

Ze získaného záznamu se naplní proměnné `r5` (hodnota počtu dotazů) a `r6` (maximální hodnota současně přihlášených uživatelů). Tyto proměnné jsou později zobrazeny v dolní části stránky.

Na závěr je objekt `Recordset` a `Command` uzavřen.

```

sql="select queries, maxusers from statistikat"
cmd.ActiveConnection=strConn
cmd.CommandText=sql
Set rs=cmd.Execute
r5=rs("queries")
r6=rs("maxusers")
rs.close

Set cmd=Nothing
%>

```

Z inicializovaných proměnných typu pole je potřeba vytvořit a inicializovat rozevírací seznamy útvarů `F1.U1`, `F1.U2`, `F1.U3`, `F1.U4` a funkce `F1.F`.

Funkce `InitU1()` je volána při načítání stránky událostí `onload` objektu `document`.

Funkce v cyklu, dle délky proměnné `U1`, vytvoří a inicializuje jednotlivé nabídky rozevíracího seznamu `F1.U1`. Dále ve třech cyklech odstraní od konce postupně všechny nabídky rozevíracích seznamů `F1.U2`, `F1.U3` a `F1.U4`.

```

function InitU1()
{
  for (var i=0; i<U1.length; i++)
  {
    F1.U1.options[i]=new Option(U1[i].Name);
    F1.U1.options[i].value=U1[i].Id1;
  }
  for (i=F1.U2.options.length; i>0; i--)
    F1.U2.options.remove(i-1);
  for (i=F1.U3.options.length; i>0; i--)
    F1.U3.options.remove(i-1);
}

```

```

for (i=F1.U4.options.length; i>0; i--)
    F1.U4.options.remove(i-1);
}

```

Funkce `InitF()` je volána při načítání stránky událostí `onload` objektu `document`.

Funkce v cyklu, dle délky proměnné `F`, vytvoří a inicializuje jednotlivé nabídky rozevíracího seznamu funkcí `F1.F`.

```

function InitF()
{
    for (var i=0; i<F.length; i++)
    {
        F1.F.options[i]=new Option(F[i].Name);
        F1.F.options[i].value=F[i].Id1;
    }
}

```

Nabídky rozevíracích seznamů útvarů `F1.U2`, `F1.U3` a `F1.U4` jsou závislé na výběru v předchozí úrovni. Tuto funkcionality zabezpečují následující funkce `InitU2()`, `InitU3()` a `InitU4()` volané událostí `onchange`.

Při změně v rozevíracím seznamu `F1.U1` je volána funkce `InitU2()`. Ve třech cyklech odstraní od konce postupně všechny nabídky rozevíracích seznamů `F1.U2`, `F1.U3` a `F1.U4`. Pokud je v rozevíracím seznamu `F1.U1` vybrána nějaká nabídka tak v cyklu prochází proměnnou `U2`. Do rozevíracího seznamu `F1.U2` vkládá jen ty nabídky, které mají stejnou hodnotu `Id1` jako vybraná nabídka.

Při změně v rozevíracím seznamu `F1.U2` je volána funkce `InitU3()`. Ve dvou cyklech odstraní od konce postupně všechny nabídky rozevíracích seznamů `F1.U3` a `F1.U4`. Pokud je v rozevíracím seznamu `F1.U2` vybrána nějaká nabídka, tak v cyklu prochází proměnnou `U3`. Do rozevíracího seznamu `F1.U3` vkládá jen ty nabídky, které mají stejnou hodnotu `Id1` jako vybraná nabídka.

Při změně v rozevíracím seznamu `F1.U3` je volána funkce `InitU4()`. V cyklu odstraní od konce postupně všechny nabídky rozevíracího seznamu `F1.U4`. Pokud je v rozevíracím seznamu `F1.U3` vybrána nějaká nabídka tak v cyklu prochází proměnnou `U4`. Do rozevíracího seznamu `F1.U4` vkládá jen ty nabídky, které mají stejnou hodnotu `Id1` jako vybraná nabídka.

```

function InitU2()
{
    for (i=F1.U2.options.length; i>0; i--)
        F1.U2.options.remove(i-1);
    for (i=F1.U3.options.length; i>0; i--)
        F1.U3.options.remove(i-1);
    for (i=F1.U4.options.length; i>0; i--)
        F1.U4.options.remove(i-1);
    if (F1.U1.selectedIndex!=-1)

```

```

{
  var j=0;
  for (var i=0; i<U2.length; i++)
  {
    if (U2[i].Id1==F1.U1.options[F1.U1.selectedIndex].value)
    {
      F1.U2.options[j]=new Option(U2[i].Name);
      F1.U2.options[j].value=U2[i].Id2;
      j++;
    }
  }
}

function InitU3()
{
  for (i=F1.U3.options.length; i>0; i--)
    F1.U3.options.remove(i-1);
  for (i=F1.U4.options.length; i>0; i--)
    F1.U4.options.remove(i-1);
  if (F1.U2.selectedIndex!=-1)
  {
    var j=0;
    for (var i=0; i<U3.length; i++)
    {
      if (U3[i].Id1==F1.U2.options[F1.U2.selectedIndex].value)
      {
        F1.U3.options[j]=new Option(U3[i].Name);
        F1.U3.options[j].value=U3[i].Id2;
        j++;
      }
    }
  }
}

function InitU4()
{
  for (i=F1.U4.options.length; i>0; i--)
    F1.U4.options.remove(i-1);
  if (F1.U3.selectedIndex!=-1)
  {
    var j=0;
    for (var i=0; i<U4.length; i++)
    {
      if (U4[i].Id1==F1.U3.options[F1.U3.selectedIndex].value)
      {
        F1.U4.options[j]=new Option(U4[i].Name);
        F1.U4.options[j].value=U4[i].Id2;
        j++;
      }
    }
  }
}

```

Pokud je pro přechod na stránku default.asp použito tlačítko *Zpět* na panelu nástrojů Internet Exploreru je nutné obnovit původní vybrané nabídky rozevřacích seznamů.

K tomuto účelu slouží následující funkce `Set1()`, která je volána při načítání stránky událostí `onload` objektu `document`. Příslušné vybrané nabídky se obnoví pomocí dat uložených ve skrytých vstupních polích `I1` až `I5` formuláře `F1`. Následně je volána příslušná funkce `InitU2()`, `InitU3()` respektive `InitU4()`, popsané výše.

```
function Set1 ()
{
  if (F1.I1.value!==-1)
  {
    F1.U1.options[F1.I1.value].selected=true;
    InitU2 ();
  }
  if (F1.I2.value!==-1)
  {
    F1.U2.options[F1.I2.value].selected=true;
    InitU3 ();
  }
  if (F1.I3.value!==-1)
  {
    F1.U3.options[F1.I3.value].selected=true;
    InitU4 ();
  }
  if (F1.I4.value!==-1)
    F1.U4.options[F1.I4.value].selected=true;
  if (F1.I5.value!==-1)
    F1.F.options[F1.I5.value].selected=true;
}
```

Pokud je třeba zrušit již vybranou nabídku rozevíracího seznamu lze tak učinit pomocí pravého tlačítka myši.

Událost `onmouseup` příslušného rozevíracího seznamu formuláře zavolá odpovídající funkci `OnMouseUpU1()`, `OnMouseUpU2()`, `OnMouseUpU3()`, `OnMouseUpU4()` respektive `OnMouseUpF()`. Ta v případě uvolnění pravého tlačítka myši zruší výběr a v případně funkcí `OnMouseUpU1()`, `OnMouseUpU2()`, `OnMouseUpU3()` zavolá odpovídající funkce `InitU2()`, `InitU3()`, `InitU4()` popsané výše. Tím se zruší nabídky i v nižších úrovních.

```
function OnMouseUpU1 ()
{
  if (event.button==2)
  {
    F1.U1.selectedIndex=-1;
    InitU2 ();
  }
}
```

```
function OnMouseUpU2 ()
{
  if (event.button==2)
  {
    F1.U2.selectedIndex=-1;
```

```

    InitU3 ();
}
}

function OnMouseUpU3 ()
{
    if (event.button==2)
    {
        F1.U3.selectedIndex=-1;
        InitU4 ();
    }
}

function OnMouseUpU4 ()
{
    if (event.button==2)
        F1.U4.selectedIndex=-1;
}

function OnMouseUpF ()
{
    if (event.button==2)
        F1.F.selectedIndex=-1;
}

```

Vynulovat formulář `F1` lze stisknutím tlačítka *Vynulovat* nebo klávesovou zkratkou. Událost tlačítka `onclick` formuláře `F1` zavolá funkci `InitF1()`.

Funkce volá metodu `reset()` objektu `form`, funkci `InitF1()`, funkci `InitF()` a metodu `focus()` objektu `text`, která nastaví kurzor do pole pro zadání příjmení.

```

function InitF1 ()
{
    F1.reset ();
    InitU1 ();
    InitF ();
    F1.P2.focus ();
}

```

Před odesláním dotazu je potřeba provést kontrolu vyplněných údajů. K tomuto účelu slouží funkce `Validate1()`.

Funkce vrací hodnotu `false` a zobrazí varování pokud vstupní textové pole pro zadání čísla obsahuje jiné znaky než číslice nebo obsahuje více než 9 číslic. Funkce využívá objekt `RegExp` (regulární výraz) a jeho metodu `test`. Také v případě, že nebylo vyplněno žádné vstupní textové pole a nebyla vybrána žádná nabídka z rozevíracích seznamů, vrací funkce hodnotu `false` a zobrazí varování.

```

function Validate1()
{
  x=F1.P1;
  p1=x.value;
  re=new RegExp("[^0-9^%]");
  if (p1.length>0 && re.test(p1))
  {
    alert("Telefonní číslo.\nPoložka obsahuje nepřípustný znak.");
    x.focus();
    return false;
  }
  if (p1.length>9)
  {
    alert("Telefonní číslo.\nLze zadat maximálně 9 číslic.");
    x.focus();
    return false;
  }
  if (F1.P1.value=="" && F1.P2.value=="" && F1.P3.value=="" &&
F1.U1.value=="" && F1.U2.value=="" && F1.U3.value=="" &&
F1.U4.value=="" && F1.F.value=="")
  {
    alert("Není vyplněna žádná položka.");
    return false;
  }
}
}

```

Odeslat dotaz je možné stisknutím tlačítka *Vyhledat*. Událost `onclick` tohoto objektu zavolá funkci `Submit1()`. Ta zavolá funkci `Validate1()`. Pokud ta vrátí hodnotu `true`, dojde k uložení nastavení vybraných nabídek rozevřacích seznamů skrytých vstupních polí `I1` až `I5` formuláře `F1`. Na závěr odešle formulář pomocí metody `submit()` formuláře `F1`.

Další možností jak odeslat dotaz je dvojkliknout myší v některém vstupním textovém poli nebo rozevřacím seznamu. Událost `ondblclick` jednoho z těchto objektů zavolá funkci `Submit1()`.

```

function Submit1()
{
  if (Validate1() !=false)
  {
    F1.I1.value=F1.U1.selectedIndex;
    F1.I2.value=F1.U2.selectedIndex;
    F1.I3.value=F1.U3.selectedIndex;
    F1.I4.value=F1.U4.selectedIndex;
    F1.I5.value=F1.F.selectedIndex;
    F1.submit();
  }
}

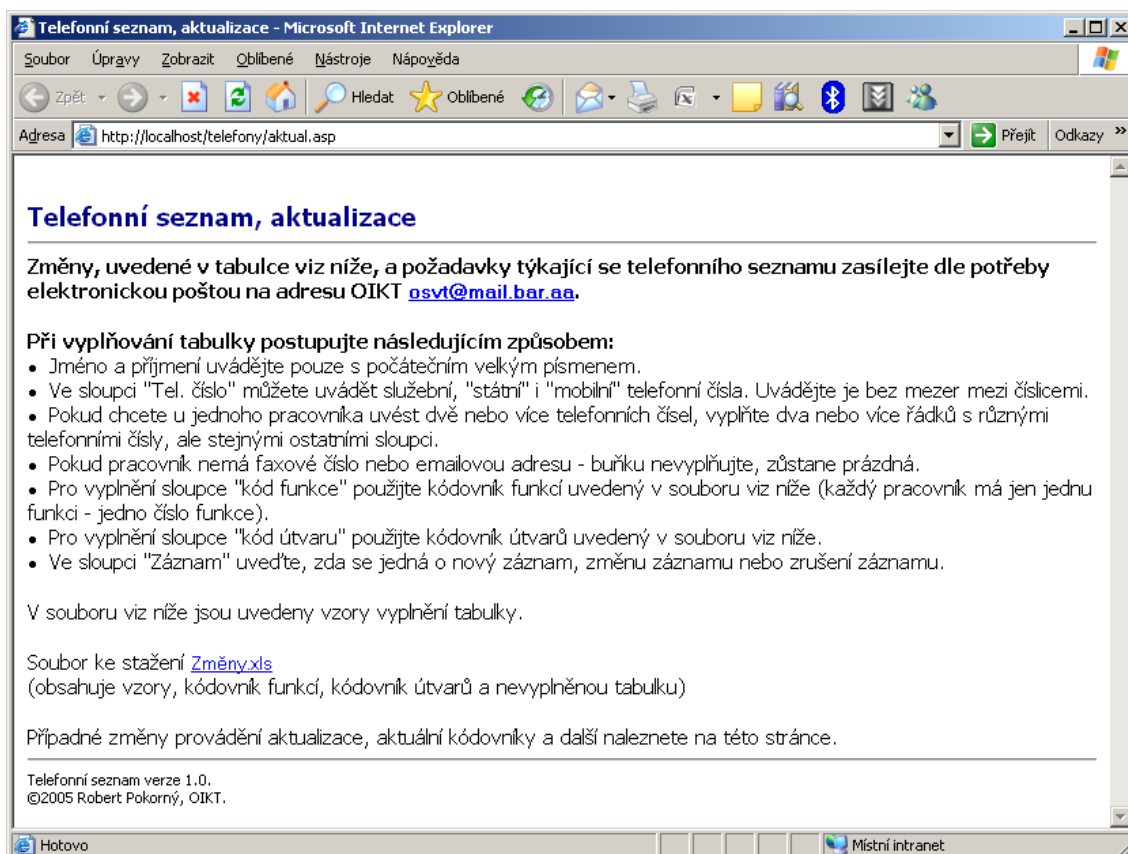
```

Odeslat dotaz je možné také stisknutím klávesy *Enter* v některém vstupním textovém poli nebo rozevřacím seznamu. Událost objektů formuláře `onkeypress` zavolá funkci `CheckKey1()`. Ta, pokud byla stisknuta klávesa *Enter*, zavolá výše popsanou funkci `Submit1()`.

```
function CheckKey1 ()
{
  if (event.keyCode==13)
    Submit1 ();
}
```

Stisknutím tlačítka *Aktualizace* se otevře okno s pokyny pro aktualizaci viz obrázek. Událost tohoto objektu `onclick` zavolá metodu `open` objektu `window`.

Okno je možné zavřít také stisknutím klávesy *Esc*.



Stisknutím tlačítka *Nápověda* se otevře okno nápovědy viz obrázek. Událost tohoto objektu `onclick` zavolá funkci `Help1()`. Ta zavolá metodu `open` objektu `window`.

Další možností jak zobrazit nápovědu je stisknout tlačítko *F1*. Událost `onhelp` objektu `window` prostřednictvím události `onload` objektu `body` také zavolá funkci `Help1()`.

Okno nápovědy je možné zavřít také stisknutím klávesy *Esc*.

Telefonní seznam, nápověda - Microsoft Internet Explorer

Soubor Úpravy Zobrazit Oblíbené Nástroje Nápověda

Zpět Hledat Oblíbené

Adresa http://localhost/telefonny/help.asp

---

## Telefonní seznam, nápověda

---

Telefonní seznam slouží k vyhledávání záznamů zadáním minimálně jedné z těchto položek:

- příjmení
- jméno
- číslo telefonu nebo faxu
- útvar (ve čtyřech úrovních, krok 1-4)
- funkce

**Vyhledávání zadáním položek:**

- minimálně jedné
- libovolné kombinace
- vyhledávají se všechny záznamy, jejichž položky současně začínají těmi zadanými (a)
- počet takto zobrazených záznamů je omezen na 200

**Položky příjmení, jméno:**

- lze zadat 0 až maximálně 25 znaků
- vyhledávají se položky, které těmito znaky začínají

**Položka číslo:**

- lze zadat 0 až maximálně 9 číslic
- vyhledávají se položky, které těmito číslicemi začínají

**Položka útvar:**

- lze zadat 0 až 4 úrovně
- rozevírací pole slouží k zadání útvaru ve čtyřech úrovních
- při výběru se automaticky aktualizuje rozevírací pole v nižší úrovni
- výběr lze v kterékoli úrovni zrušit kliknutím pravého tlačítka myši

**Položka funkce:**

- rozevírací pole slouží k zadání funkce
- rozevírací pole slouží k zadání pseudo funkce např. sekretariát, vrátnice...
- výběr lze v kterékoli úrovni zrušit kliknutím pravého tlačítka myši

**Práce s formulářem:**

- vynulovat formulář lze tlačítkem [Vynulovat](#)
- vyhledat lze z kterékoli položky klávesou [Enter](#), [dvojklikem](#) nebo tlačítkem [Vyhledat](#)
- k další položce lze přejít klávesou [Tab](#)
- k předchozí položce lze přejít klávesami [Shift+Tab](#)
- v rozevíracím poli je možno se pohybovat pomocí kláves počátečního písmene výběru
- k určité položce či tlačítku lze přejít pomocí klávesových zkratk [Alt+klávesa](#) (levý Alt + podtržený znak názvu položky) např. pro položku funkce Alt+f

**Tisk:**

- stlačením tlačítka [Tisk](#) se výstup převede na černobílý a spustí se tisk
- následné automatické spuštění tisku vyžaduje Internet Explorer verze 5 a vyšší
- standardním způsobem z Internet Exploreru (z menu Soubor položkou Tisk...)

**Nápověda:**

- nápovědu lze zobrazit pomocí klávesy [F1](#) nebo tlačítka [Nápověda](#)
- nápovědu lze zavřít pomocí klávesy [Esc](#) nebo zavřením okna

---

Microsoft Internet Explorer; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; InfoPath.2; .NET CLR 2.0.50727)  
Telefonní seznam verze 1.0.  
©2005 Robert Pokorný, OIKT.

Hotovo Místní intranet

```

function Help1 ()
{
    window.open("help.asp");
    event.returnValue=false;
}

```

### 3.6.3 Soubor dotaz.asp

Odesláním formuláře jsou vstupní údaje formuláře předány stránce dotaz.asp.

Následující skript, prováděný na straně serveru, vytvoří ADO objekty `Command` a `Recordset`. Nastaví vlastnosti objektu `Command` připojení, typ uložená procedura, její jméno a parametry a příkaz provede. Získaný záznam z tabulky `StatistikaT` uloží v objektu `Recordset`. Ze získaného záznamu naplní proměnné `r5` (hodnota počtu dotazů) a `r6` (maximální hodnota současně přihlášených uživatelů). Tyto proměnné jsou později zobrazeny v dolní části stránky. Na závěr objekt `Command` uzavře.

Z dat formuláře zaslaných metodou `post` vytvoří SQL dotaz, bez propojení tabulek, který zjistí počet záznamů vyhovujících danému dotazu. Vytvoří ADO objekt `Command`. Nastaví vlastnosti objektu `Command` připojení, typ SQL dotaz a příkaz provede. Získaný záznam z tabulky `SeznamT` uloží v objektu `Recordset`. Hodnotou ze získaného záznamu naplní proměnnou `c`. Do proměnné `mc` vloží hodnotu `session` proměnné `MaxC`.

```

<%
set cmd=Server.CreateObject ("ADODB.Command")

cmd.ActiveConnection=strConn
cmd.CommandText="usp_TelStat"
cmd.CommandType=adCmdStoredProc

cmd.Parameters.Append cmd.CreateParameter ("@p2", adVarChar,
adParamInput, 5, Application("ActualUser"))
cmd.Parameters.Append cmd.CreateParameter ("@r5", adInteger,
adParamOutput)
cmd.Parameters.Append cmd.CreateParameter ("@r6", adInteger,
adParamOutput)

set rs=Server.CreateObject ("ADODB.Recordset")
cmd.Execute

r5=cmd.Parameters ("@r5").Value
r6=cmd.Parameters ("@r6").Value
set cmd=Nothing

sql1="select count (*) as c from seznamt "
sql2="select prijmeni, jmeno, titult.titul as t, tel, fax, email,
funkcet.funkce as f, utvar1, utvar2, utvar3, utvar4 from seznamt "
sql3="left outer join titult on seznamt.titulid=titult.id "
sql4="inner join funkcet on seznamt.funkceid=funkcet.id "
sql5="left outer join utvar1t on kuid1=id1 "
sql6="left outer join utvar2t on kuid2=id2 "

```

```

sql7="left outer join utvar3t on kuid3=id3 "
sql8="left outer join utvar4t on kuid4=id4 "
sql9="where "

i=1
p1=request.form("p1")
if p1<>"" then
  if i>1 then
    sql9=sql9 & "and "
  else
    i=i+1
  end if
  sql9= sql9 & "(tel like '" & p1 & "%' or fax like '" & p1 & "%') "
end if
p2=request.form("p2")
if p2<>"" then
  if i>1 then
    sql9=sql9 & "and "
  else
    i=i+1
  end if
  sql9= sql9 & "prijmeni like '" & p2 & "%' "
end if
p3=request.form("p3")
if p3<>"" then
  if i>1 then
    sql9=sql9 & "and "
  else
    i=i+1
  end if
  sql9= sql9 & "jmeno like '" & p3 & "%' "
end if
u1=request.form("u1")
if u1<>"" then
  if i>1 then
    sql9=sql9 & "and "
  else
    i=i+1
  end if
  sql9= sql9 & "kuid1=" & u1 & " "
end if
u2=request.form("u2")
if u2<>"" then
  if i>1 then
    sql9=sql9 & "and "
  else
    i=i+1
  end if
  sql9= sql9 & "kuid2=" & u2 & " "
end if
u3=request.form("u3")
if u3<>"" then
  if i>1 then
    sql9=sql9 & "and "
  else
    i=i+1
  end if
  sql9= sql9 & "kuid3=" & u3 & " "

```

```

end if
u4=request.form("u4")
if u4<>" " then
  if i>1 then
    sql9=sql9 & "and "
  else
    i=i+1
  end if
  sql9= sql9 & "kuid4=" & u4 & " "
end if
f=request.form("f")
if f<>" " then
  if i>1 then
    sql9=sql9 & "and "
  else
    i=i+1
  end if
  sql9= sql9 & "funkceid=" & f & " "
end if

sql= sql1 & sql9
set cmd=Server.CreateObject("ADODB.Command")
cmd.ActiveConnection=strConn
cmd.CommandText=sql
Set rs=cmd.Execute
c=rs("c")
mc=Session("MaxC")
%>

```

Pokud nebyl nalezen žádný záznam, nebo pokud počet nalezených záznamů je větší než povolené maximum, skript, prováděný na straně serveru, ve start tagu `body` pomocí události `onload` objektu `body` nastaví vlastnost `disabled` tlačítka *Tisk* a tím tlačítko zakáže viz níže.

Dále vypíše text s počtem nalezených záznamů a, v případě, že počet nalezených záznamů je větší než povolené maximum, červeně text o překročení maxima a jeho hodnoty s pokynem k upřesnění dotazu.

```

<body<% if (c=0 or mc<c) then Response.Write ("
onload=""F1.B3.disabled=true"";") %> onhelp="Help1()" ">

<span class="heading">Telefonní seznam</span><hr>
<%

Response.Write ("Počet záznamů odpovídajících dotazu: <b>" & c &
"</b>.<br><br>")
if mc<c then
  Response.Write("<p style=""color:red""><b>Bylo překročeno maximum " &
CStr(mc) & " záznamů.")
  Response.Write("<br>Je třeba upřesnit dotaz.</b></p>")

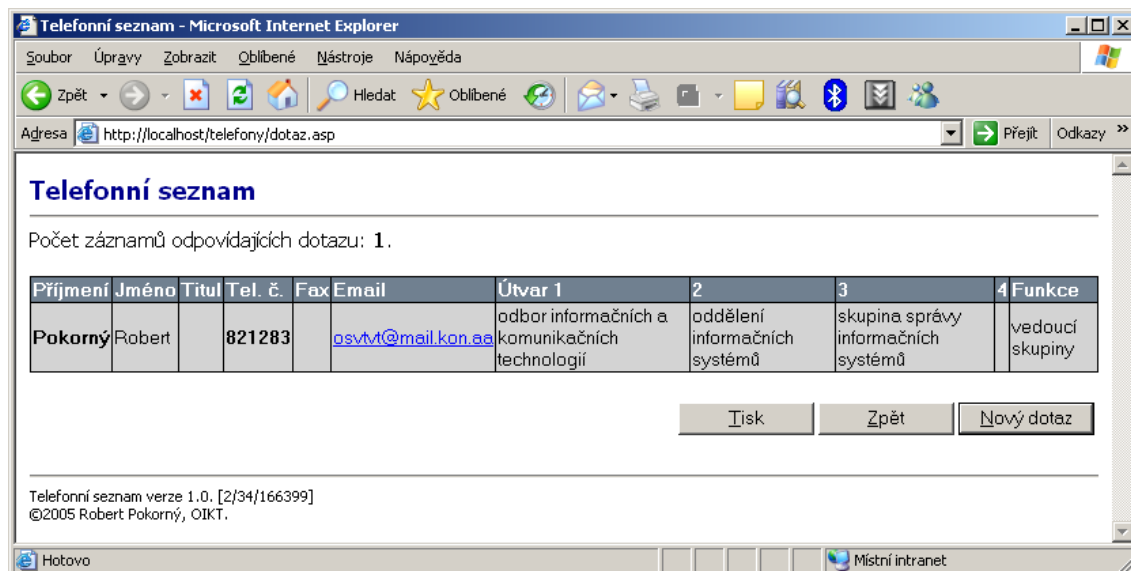
```

Jinak, pokud byly nalezeny nějaké záznamy, tedy počet nalezených záznamů je větší než nula, sestaví SQL dotaz s propojením tabulek, nastaví

vlastnosti objektu `Command` připojení, typ SQL dotaz a příkaz provede. Získané záznamy uloží v objektu typu `Recordset`.

Ze získaných záznamů v cyklu vytvoří HTML tabulku a naplní ji nalezenými záznamy viz obrázek.

Na závěr objekt `Command` uzavře.



```
elseif c>0 then
    sql10="order by prijmeni, jmeno, utvar1, tel"
    sql= sql2 & sql3 & sql4 & sql5 & sql6 & sql7 & sql8 & sql9 & sql10

    cmd.ActiveConnection=strConn
    cmd.CommandText=sql
    Set rs=cmd.Execute

    Response.Write "<table width=""100%"" id=""ot1"" cellpadding=1
style=""font-family: ms sans serif; font-size: 14px""
bgcolor=""Black"">"
    Response.Write "<tr>"
    Response.Write "<th>"
    Response.Write "Příjmení"
    Response.Write "</th>"
    Response.Write "<th>"
    Response.Write "Jméno"
    Response.Write "</th>"
    Response.Write "<th>"
    Response.Write "Titul"
    Response.Write "</th>"
    Response.Write "<th nowrap>"
    Response.Write "Tel. č."
    Response.Write "</th>"
    Response.Write "<th nowrap>"
    Response.Write "Fax"
    Response.Write "</th>"
    Response.Write "<th>"
    Response.Write "Email"
```

```

Response.Write "</th>"
Response.Write "<th>"
Response.Write "Útvar 1"
Response.Write "</th>"
Response.Write "<th>"
Response.Write "2"
Response.Write "</th>"
Response.Write "<th>"
Response.Write "3"
Response.Write "</th>"
Response.Write "<th>"
Response.Write "4"
Response.Write "</th>"
Response.Write "<th>"
Response.Write "Funkce"
Response.Write "</th>"
Response.Write "</tr>"
While Not rs.EOF
  Response.Write "<tr>"
  Response.Write "<td>"
  Response.Write "<b>" & rs("prijmeni") & "</b>"
  Response.Write "</td>"
  Response.Write "<td>"
  Response.Write rs("jmeno")
  Response.Write "</td>"
  Response.Write "<td>"
  Response.Write rs("t")
  Response.Write "</td>"
  Response.Write "<td>"
  Response.Write "<b>" & rs("tel") & "</b>"
  Response.Write "</td>"
  Response.Write "<td>"
  Response.Write rs("fax")
  Response.Write "</td>"
  Response.Write "<td>"
  Response.Write "<a href=""mailto:" & rs("email") & """>" &
rs("email") & "</a>"
  Response.Write "</td>"
  Response.Write "<td>"
  Response.Write rs("utvar1")
  Response.Write "</td>"
  Response.Write "<td>"
  Response.Write rs("utvar2")
  Response.Write "</td>"
  Response.Write "<td>"
  Response.Write rs("utvar3")
  Response.Write "</td>"
  Response.Write "<td>"
  Response.Write rs("utvar4")
  Response.Write "</td>"
  Response.Write "<td>"
  Response.Write rs("f")
  Response.Write "</td>"
  Response.Write "</tr>"
  rs.MoveNext
Wend
Response.Write "</tr></table>"
end if

```

```
Set cmd=Nothing
%>
```

Formulář obsahuje v HTML tabulce ovládací tlačítka *Zpět*, *Nový dotaz* a *Tisk*. Pomocí vlastnosti `accesskey` jsou definovány klávesové zkratky pro snadný přístup k těmto tlačítkům viz kód.

Tlačítko *Zpět* při události `onclick` využívá metodu `back()` objektu `history` objektu `window`. Tlačítko *Nový dotaz* otevírá stránku `default.asp` prostřednictvím metody `submit` objektu `form`.

```
<form action="default.asp" method="post" name="F1">
<table border="0" width="100%">
<tr><td align="right" bgcolor="white">
<button name="B3" accesskey="t"
onclick="print1()"><u>T</u>isk</button>&nbsp;
<button name="B1" accesskey="z"
onclick="history.back();"><u>Z</u>pět</button>&nbsp;
<button name="B2" type="submit" accesskey="n"><u>N</u>ový
dotaz</button>
</td></tr>
</table>
</form>
```

Stisknutím tlačítka *Tisk* událost `onclick` zavolá funkci `Print1()`. Ta skryje respektive zobrazí tlačítka *Zpět* a *Nový dotaz* pomocí vlastnosti `visibility` objektu `style` objektu `button`. Dále změní nápis tlačítka *Tisk* na *Zpět* respektive *Zpět* na *Tisk* vlastností `innerHTML` objektu `button`, nastaví vlastnost `href` objektu `link` a tím „zamění“ soubor `styles.css` za `stylesp.css` (pro černobílý tisk) a naopak. Pokud je detekován Internet Explorer verze 5 a vyšší, dokument vytiskne pomocí metody `print()` objektu `window`.

```
function Print1()
{
  if (pr==0)
  {
    F1.B1.style.visibility="hidden";
    F1.B2.style.visibility="hidden";
    F1.B3.innerHTML="Zpě<u>t</u>";
    L1.href="stylesp.css";
    pr=1;
    if (ie>=5)
      window.print();
  }
  else
  {
    F1.B1.style.visibility="visible";
    F1.B2.style.visibility="visible";
    F1.B3.innerHTML="<u>T</u>isk";
  }
}
```

```
L1.href="styles.css";  
pr=0;  
}  
}
```

Stisknutím klávesy *F1* se otevře okno nápovědy viz obrázek výše. Událost `onhelp` objektu `window` prostřednictvím události `onload` objektu `body` zavolá funkci `Help1()`. Ta otevře okno metodou `open()` objektu `window`.

Okno nápovědy je možné zavřít také stisknutím klávesy *Esc*.

```
function Help1()  
{  
  window.open("help.asp");  
  event.returnValue=false;  
}
```

### 3.6.4 Soubor aktual.asp

Soubor `aktual.asp` zobrazí informace o aktualizaci IS Telefonní seznam po stisknutí tlačítka *Aktualizace* ze stránky `default.asp` viz obrázek a popis výše.

### 3.6.5 Soubor help.asp

Soubor `help.asp` zobrazí nápovědu k IS Telefonní seznam po stisknutí tlačítka *Nápověda* nebo klávesy *F1* ze stránek `default.asp` a `dotaz.asp` viz obrázek a popis výše.

V dolní části okna pod čarou jsou pro servisní účely navíc zobrazeny informace o verzi Internet Exploreru a verzi operačního systému klienta.

### 3.6.6 Soubor styles.css

Soubor `styles.css` obsahuje externí definice stylů. Každá stránka, která používá tento externí style sheet musí obsahovat `link` tag v `head` sekci.

```
<head>  
<link rel="stylesheet" href="Styles.css" type="text/css">
```

### 3.6.7 Soubor stylesp.css

Soubor `stylesp.css` obsahuje externí definice stylů pro tisk ze stránky `dotaz.asp`.

### 3.6.8 Soubor Include\connection.asp

Soubor connection.asp nastavuje vlastnost ActiveConnection objektu Command.

Obsahuje údaje uživatelské jméno, heslo, název serveru a databáze a další pro připojení k datovému zdroji.

Pro vložení obsahu tohoto souboru do jiné asp stránky je použita direktiva #include.

```
<!-- #include file="Include/connection.asp" -->
```

## 3.7 Popis souborů pro editaci záznamů

Složka Edit obsahuje soubory sloužící ke správě záznamů databáze pro uživatele s oprávněním provádět změny.

### 3.7.1 Soubor default.asp

Soubor default.asp vychází z výše popsaného souboru default.asp.

Telefonní seznam PČR správy hl. m. Prahy


Příjmení, jméno:	<input type="text"/>	Záznam:	Nový
Titul:	<input type="text"/>		Upravit
Tel. číslo, fax:	<input type="text"/>		Smazat
Email:	<input type="text"/>		
Útvar:	[Krok1] a kancelář ředitele správy b oddělení vnitřní kontroly c školní policejní středisko		
	[Krok2]		
	[Krok3]		
	[Krok4]		
Funkce:	01 administrativní a spisový pracovník 02 analytik inform. a komunik. technologií 03 archivář		Nápověda
			Aktualizace
			Výhledat
			Vyhledat

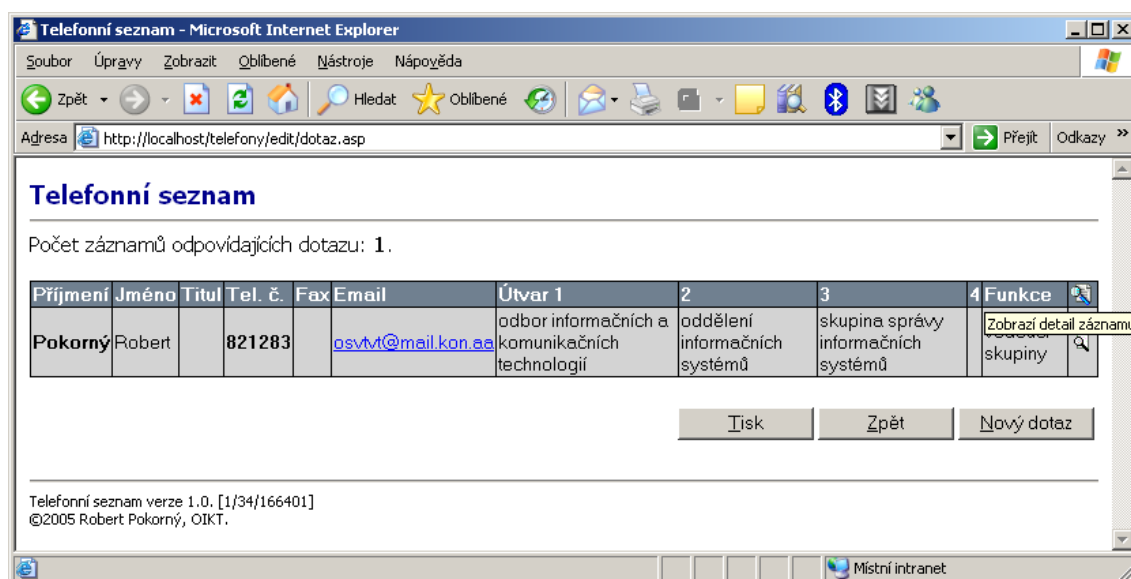
Telefonní seznam verze 1.0. [1/34/166400]  
Určeno pro Microsoft Internet Explorer verze 4 a vyšší.  
©2005 Robert Pokorný, OIKT.

Formulář je doplněn o rozevírací seznam pro výběr titulu, obsahuje samostatná textová pole pro zadání 9 číslic telefonního a faxového čísla. Rozevírací seznamy pro výběr útvaru a pro výběr funkce obsahují navíc před názvem kód.

Místo horní čtveřice tlačítek se nachází tlačítka pro práci se záznamy *Nový*, *Upravit* a *Smazat* viz obrázek.

### 3.7.2 Soubor dotaz.asp

Soubor dotaz.asp vychází z výše popsaného souboru default.asp. Výsledná tabulka nalezených záznamů je doplněna o sloupec s ikonou  umožňující zobrazení detailu konkrétního záznamu prostřednictvím souboru default.asp viz obrázek.



### 3.7.3 Soubor insert.asp

Po stisknutí tlačítka *Nový* jsou data formuláře zkontrolována a předána stránce insert.asp. Ta prostřednictvím ADO a SQL příkazu Insert vloží nový záznam do databáze.

### 3.7.4 Soubor update.asp

Po stisknutí tlačítka *Upravit* jsou data formuláře zkontrolována a předána stránce update.asp. Ta prostřednictvím ADO a SQL příkazu Update upraví daný záznam.

### 3.7.5 Soubor delete.asp

Po stisknutí tlačítka *Smazat* jsou data formuláře předána stránce delete.asp. Ta prostřednictvím ADO a SQL příkazu Delete smaže daný záznam.

## 4 Parametry vytvořeného systému

Parametry vytvořeného systému beze zbytku splňují všechny požadavky zadání.

Informační systém Telefonní seznam umožňuje vyhledávání záznamů zadáním minimálně jedné z požadovaných položek pro vyhledávání (příjmení, jméno, číslo telefonu nebo faxu, útvar a funkce).

Seznam útvarů 2. až 4. úrovně je závislý na výběru v předchozí úrovni a nelze tak vybírat nesmyslné kombinace. Tato vlastnost je realizována na straně klienta a tím odpadá nutnost načítání stránky při každé změně.

Vyhledávají se všechny záznamy, jejichž položky současně začínají zadanými a počet takto zobrazených záznamů je omezen na 200.

Záznamy kromě položek pro vyhledávání obsahují položky titul a emailová adresa.

Aplikace obsahuje stručnou nápovědu k práci s formulářem, vyhledávání, k tisku a dalším. Dále je možné zobrazit stránku s informacemi k aktualizaci tohoto systému.

IS Telefonní seznam umožňuje centrálně spravovat tento IS pro uživatele s oprávněním provádět změny.

## 5 Zhodnocení

Praktickou realizací tohoto informačního systému jsem si ověřil, že třívrstvá webová aplikace typu klient/server je vhodná pro vytvoření telefonního seznamu jako moderního informačního systému.

Studiem odborné literatury při tvorbě aplikace jsem si prohloubil a rozšířil vědomosti z oblastí HTML, DHTML, CSS, JavaScript (JScript), VBScript, ADO, T-SQL a dalších.

Organizace touto prací získala uživatelsky jednoduchý, snadno ovladatelný telefonní seznam, umožňující vyhledávání dle známých položek, bez nutnosti jejich specifikace, prostým vyplněním známých údajů nebo jejich částí, včetně organizační struktury a funkcí. Pravidelné aktualizace a centrální správa umožňují organizaci poskytovat uživatelům přesné a aktuální údaje.

Uvedená literatura sloužila především pro studijní účely.

## Literatura

- [1] Alex Homer, David Sussman, Brian Francis, George Reilly, Dino Esposito, Craig McQueen, Simon Robinson, Richard Anderson, Andrea Chiarelli, Chris Blexrud, Bill Kropog, John Schenken, Matthew Gibbs, Dean Sonderegger, Dan Denault: *Active Server Pages 3.0 PROFESIONÁLNĚ*. Vydání první, Praha, Vydavatelství a nakladatelství Computer Press, 2000. Autorizovaný překlad z originálního anglického vydání Professional Active Pages 3.0. ISBN 80-6097-47-1.
  
- [2] Kenneth Henderson: *Mistrovství v Transact-SQL*. Vydání první, Praha, Vydavatelství a nakladatelství Computer Press, 2000. Autorizovaný překlad z originálního anglického vydání The Guru's Guide to Transact-SQL. ISBN 80-7226-393-5.
  
- [3] Microsoft HTML Help: *Microsoft ActiveX Data Objects 2.5 (ADO)*. Microsoft Corporation, 1998-2003.
  
- [4] Books Online: *Microsoft SQL Server 2005 Books Online*. Microsoft Corporation, 2006.
  
- [5] Microsoft HTML Help: *Visual Basic Scripting Edition*. Microsoft Corporation.
  
- [6] Microsoft HTML Help: *JScript*. Microsoft Corporation.
  
- [7] Microsoft HTML Help: *Web Workshop: DHTML, HTML & CSS*. Microsoft Corporation, 1999.
  
- [8] Microsoft: *Dokumentace ke službě Microsoft IIS 5.1*. Microsoft Corporation, 1997-2001.