

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

GENEALOGICKÝ INFORMAČNÍ SYSTÉM  
V ASP.NET 2.0

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

MARTIN REPTA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# GENEALOGICKÝ INFORMAČNÍ SYSTÉM V ASP.NET 2.0

GENEALOGY INFORMATION SYSTEM IN ASP.NET 2.0

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN REPTA

VEDOUCÍ PRÁCE

SUPERVISOR

ING. ZBYNĚK KŘIVKA PH.D.

BRNO 2008

## **Abstrakt**

Táto bakalárska práca sa venuje vývoju genealogického informačného systému. Čitateľ má možnosť dozvedieť sa zaujímavé informácie o genealógii ako vednom obore. Vysvetľuje problematiku hľadania našich predkov. Ďalej sa venuje ASP.NET 2.0 a práci s jeho pokročilými možnosťami tvorby webových aplikácií. Je tu vysvetlená problematika generovania grafického znázornenia rodokmeňa, PDF súborov za použitia open-source knižnice. V závere sú popísané možné vylepšenia tohto systému.

## **Klíčová slova**

ASP.NET 2.0, Microsoft .NET Framework, Genealógia, Generovanie grafov, Generovanie obrázkov, Generovanie PDF.

## **Abstract**

This bachelor thesis is engaged in evolution on genealogy information system. The reader is in a position to learn interesting information about genealogy as disciplinary division. It is explaining the question of searching our ancestry. Further, it is addicted to ASP.NET 2.0 and it's advanced capability of the creating web applications. The question of the generation of the genealogical tree, PDF files using the open-source library is also explained there. In the end, there are described possible upgrades of this system.

## **Keywords**

ASP.NET 2.0, Microsoft .NET Framework, Genealogy, Graf generation, Picture generation, PDF generation.

## **Citace**

Repta Martin: Genealogický informační systém v ASP.NET 2.0. Brno, 2008, bakalárska práca, FIT VUT v Brně.

# Genealogický informační systém v ASP.NET 2.0

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Zbyňka Křivky Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Martin Repta  
24.4.2008

## Poděkování

Ďakujem Ing. Zbyňkovi Křivkovi Ph.D., za odborné vedenie, rady a podnety, ktoré mi počas práce poskytoval.

© Martin Repta, 2008

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

Obsah .....	1
1 Úvod.....	3
1.1 Genealógia.....	4
1.1.1 Získavanie informácií .....	4
2 Použité technológie.....	5
2.1 Microsoft .NET Framework.....	5
2.2 ASP.NET 2.0.....	6
2.2.1 Stavové prostredie nad bezstavovým protokolom .....	7
2.3 Microsoft SQL Server .....	7
2.4 Vývojové prostredie .....	8
2.4.1 Microsoft Visual Studio.....	8
3 Analýza a návrh .....	9
3.1 Špecifikácia požiadaviek.....	9
3.2 Dátový model .....	9
3.2.1 Popis objektov.....	10
4 Implementácia.....	11
4.1 Uživateľské účty.....	11
4.1.1 Membership API.....	11
4.1.2 Správa rolí.....	12
4.2 Rodokmeň .....	15
4.2.1 Založenie rodokmeňa.....	16
4.2.2 Tvorba pozvánok k rodokmeňu .....	17
4.2.3 Generovanie obrázka rodokmeňa.....	17
4.3 Generovanie PDF .....	19
4.3.1 Knižnica iTextSharp .....	19
4.4 Generovanie XML.....	20
4.4.1 Namespace System.XML .....	20
4.4.2 XML výstup .....	20
4.5 Vývoj viacjazyčných informačných systémov v ASP.NET 2.0.....	21
4.5.1 Namespace System.Resources .....	21
4.5.2 Namespace System.Globalization.....	21
4.5.3 Lokalizácia a zdroje .....	21
4.5.4 Globalizácia a zdroje.....	22
4.6 Trieda BasePage.....	23

5	Záver .....	25
5.1	Vylepšenia systému.....	25
	Literatúra .....	26
	Zoznam príloh.....	27
	Súborová štruktúra CD .....	27

# 1 Úvod

Témou tejto bakalárskej práce je vytvorenie genealogického informačného systému, ktorého hlavnou úlohou bude uchovávať informácie o osobách a zaznamenávať vzťahy medzi nimi, z hľadiska genealógie. Tieto vzťahy budú zobrazované vo forme grafov a údaje generované do PDF dokumentov.

Prvá podkapitola sa zaoberá samotnou genealógiou a heraldikou, ako vedou a popisuje jednotlivé typy rodokmeňov a spôsoby hľadania údajov.

V druhej kapitole sú spomenuté technológie, ktoré som pri tvorbe tohto informačného systému využil. Venuje sa a stručne popisuje Microsoft .NET Framework a jeho využitie pri tvorbe webových informačných systémov v ASP.NET.

Tretia kapitola je venovaná analýze a samotnému návrhu informačného systému. Venuje sa dátovému modelu a popisom jednotlivých objektov.

V štvrtej kapitole sú spomenuté postupy použité pri implementácii a programovaní. Poukazuje na využitie aplikačného rozhrania Membership, ktoré je súčasťou ASP.NET 2.0. Podrobnejšie sa kapitola zaoberá i generovaním grafu rodokmeňa a exportu dát do PDF súborov využitím open-source knižnice *iTextSharp*. Zahŕňa informácie o vývoji a spôsobe vývoja viacjazyčných informačných systémov. V závere štvrtej kapitoly popisujem export do XML súboru.

Posledná kapitola je zhrnutím dosiahnutých výsledkov a vlastných dojmov. Popisuje budúce využitie a navrhuje možné vylepšenia tohto systému.

## 1.1 Genealógia

Pátranie po vlastných predkoch je neopísateľné dobrodružstvo. Amatérska genealógia láka čoraz viac ľudí, ktorí zatúžili spoznať vlastné korene. Rodopis v mnohých prípadoch dokáže poskytnúť netušené odpovede, ktoré zaskočia aj samotných pátračov.

Genealógia je historická veda skúmajúca vývoj rodov a vzťahy medzi rodovo príbuznými jedincami. Genealogický výskum začína od jedinca zo skúmaného rodu buď ako rodiča alebo potomka a podľa toho skúma jeho rodinné vzťahy k predkom (vývod) tzv. ascendentom alebo k potomkom (rozrod) tzv. descendentom.

*Vývod* je súhrn predkov jedinca v priamej línii. Zaznamenávajú sa do neho všetci priamy pokrvní predkovia danej osoby po otcovej aj matkinej línii. Vývod sa využíval hlavne v oblasti dokazovania vlastníckych, prípadne nástupníckych práv. Najčastejšie sa vypracúval z troch alebo štyroch generácií predkov. V súčasnosti je dôležitou pomôckou pri biologickom a sociologickom skúmaní, ukazuje vplyvy dedičnosti a znázorňuje vzájomné prestupovanie najrôznejších spoločenských vrstiev v jednotlivých rodoch.

*Rozrod* je opis potomstva najstaršieho zistiteľného páru. Ide o všetkých potomkov po meči a po praslici. Rozrod je súpis, prípadne grafické znázornenie pokrvných potomkov určitej osoby. Má dve základné formy. V prvej sú uvedení všetci potomkovia synov a dcér, ktoré sa nevydali, prípadne adoptované osoby. V minulosti bol najčastejší spôsob grafického znázornenia rozrodu vo forme stromu, ktorého jednotlivé vetvy znázorňovali jednotlivé rodové línie.

*Rodokmeň (strom života)* vychádza taktiež od najstaršieho zistiteľného predka – zakladateľa rodu. No je to opis potomstva iba „po meči“. Zaznamenávajú sa síce všetci potomkovia, no do ďalšej generácie sa sledujú len rodiny synov, teda osoby s rovnakým rodovým menom. Používal sa najmä na dokazovanie šľachtictva.

### 1.1.1 Získavanie informácií

Genealogický výskum je veľmi náročný. Okrem trpezlivosti sú pri výskume potrebné aj financie. Štúdium matrik je totiž spoplatnené. Za každý zapožičaný zväzok sa platí 50-korunovým kolkom. Originálne matriky však archívy neposkytujú. Niekde sú dispozícii čítačky s mikrofilmami. Voľne prístupné sú šľachtické písomnosti ako aj ďalšie sekundárne zdroje.

Najväčšou genealogickou databázou na svete disponujú mormoni čiže Cirkev Ježiša Krista Svätých Neskorších Dní (CJKSND). Genealogické databázy zhromaždené členmi Cirkvi Ježiša Krista Svätých Neskorších Dní sú voľne prístupné na internetovej stránke [www.familysearch.org](http://www.familysearch.org).

Záujemci o zostavenie rodokmeňa môžu osloviť firmy zaoberajúce sa genealógiou na profesionálnej úrovni. Je to však finančne veľmi nákladné.

Zdroj vid' literatúra [7].



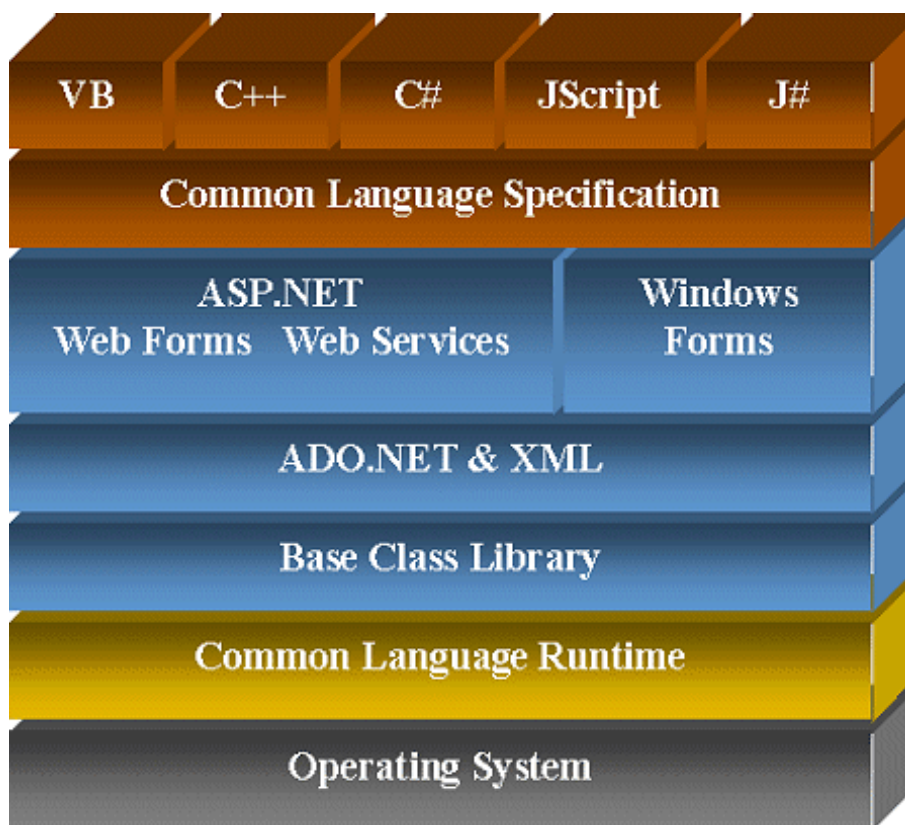
## 2 Použité technológie

Táto kapitola sa zaoberá technológiami, ktoré som pre vývoj informačného systému použil. Stručne popisuje samotný Microsoft .NET Framework, na ktorom informačný systém beží. Ďalej popisuje možnosti vývoja webových informačných systémov v ASP.NET 2.0.

### 2.1 Microsoft .NET Framework

Informačný systém je možno vytvoriť ako stolnú alebo webovú aplikáciu. Výber technológie padol na Microsoft .NET Framework, pretože prináša zaujímavé a efektívne riešenia, tak ako pre stolné, tak i pre webové aplikácie.

Microsoft .NET Framework podporuje tzv. Common Type Specification (CTS), čím programátorovi umožňuje výber programovacieho jazyka, ktorý mu najviac vyhovuje. Môžete si napísať triedy v C# a vytvárať inštancie z nich vo VB.NET. Môžete vyhodit' výnimku napísanú vo VB.NET a odchytiť ju v C# triede. Teda výber programovacieho jazyka je čisto na programátorovi, keď samozrejme odmyslíme limitujúce faktory niektorých programovacích jazykov pre nami vyvíjanú aplikáciu. Je možno si zvoliť z niektorých z jazykov: C#, C++, VB, J#, JScript.



Obr.1: Architektúra Microsoft .NET Framework – zdroj vid' literatúra [10]

Microsoft .NET Framework pracuje na vrchnej vrstve operačného systému, ktorým môže byť akýkoľvek z operačných systémov firmy Microsoft, a obsahuje množstvo komponent s predprogramovanými riešeniami obsiahnuté v tzv. Base Class Library (BCL). Knižnica pokrýva širokú škálu potrieb v oblasti užívateľského rozhrania, prístupu k dátam, pripojenie k databázam, kryptografie, sieťovej komunikácie, numerických algoritmov, XML či vývoja v oblasti webových aplikácií. Knižnica je využívaná programátorom, ktorý ju kombinuje s vlastným kódom.

Common Language Runtime (CLR) sa postará o radu nízko úrovňových povinností, ktoré v minulosti museli programátori riešiť sami. Sú to najme správa pamäti, vytváranie a rušenie objektov, bezpečnosť a kontrola oprávnení k vykonávaným operáciám, spúšťanie a zastavovanie vlákien kódu, naťahovanie potrebných knižníc a komponent do pamäte.

Podrobnejšie informácie možno nájsť v literatúre [1].

## **2.2 ASP.NET 2.0**

Ak chceme vyvíjať webové aplikácie pod ASP.NET 2.0, je nutné, aby sme si nainštalovali už spomínanú komponentu Microsoft .NET Framework 2.0. Existujú aj staršie verzie 1.0 a 1.1, ktoré ale neobsahujú mnohé veci obsiahnuté práve vo verzii 2.0. Medzi novinky v tejto verzii patria MasterPages, Membership, Themes a mnohé ďalšie, ktoré zrýchľujú a zjednodušujú vývoj webových aplikácií. V súčasnosti už vyšla verzia .NET Framework 3.5 formálne nazývaná aj ako WinFix s ďalšími novinkami hlavne pre Windows Vista a Windows Server 2008.

Aplikácie založené na ASP.NET sú rýchlejšie od čisto skriptovacích jazykov, pretože sú predkompilované do tzv. DLL súborov. ASP.NET uľahčuje prechod od programovania klasických aplikácií pre Windows k programovaniu webových aplikácií. Stránky sú tvorené z objektov, a serverových ovládacích prvkov (Controls). Je teda možné používať ovládacie prvky ako tlačidlo (Button), popisok (Label) a ďalšie. Týmto prvkom sa dajú nastavovať určité vlastnosti, zachytávať a ošetrovať udalosti. Pri preklade sa produkuje čistý HTML prípadne XHTML kód tvoriaci výslednú stránku poslanú na klientský prehliadač. Programátori pritom majú na výber veľké množstvo programovacích jazykov, ktoré podporuje už samotný Microsoft .NET Framework. Dajú sa prevádzkovať na rôznych webových serveroch napr. IIS (Windows), Apache (Windows, Linux s open-source implementáciou .NET Mono). Podrobnejšie informácie možno nájsť v literatúre [2] alebo na Internete.

## 2.2.1 Stavové prostredie nad bezstavovým protokolom

Webový protokol http je sám o sebe bezstavový. Programovanie riadené udalosťami však zachovanie stavu vyžaduje. ASP.NET rieši tento problém kombináciou HTML a JavaScriptu a to pomocou dvoch základných techník: ViewState a Session State.

### 2.2.1.1 ViewState

ViewState dokáže uchovať informácie medzi opakovaným odoslaním formulára na server (postback) v zakódovanom tvare v skrytých formulárových poliach. Jeho hlavnou výhodou je, že využíva iba HTML a nevyžaduje žiadnu špeciálnu podporu na strane servera ani klienta. Nevýhodou ale zostáva fakt, že sa medzi serverom a klientom prenáša väčší objem dát, hlavne ak je ViewState využívané nesprávne.

### 2.2.1.2 Session State

Session State na rozdiel od ViewState ukladá všetky informácie na strane servera a predáva ich vo forme cookies, alebo ako súčasť URL. To síce znižuje objem prenášaných dát, ale na druhej strane zase ikladie vyššie nároky na výkon serveru. ASP.NET umožňuje ukladanie Session State do samostatného procesu alebo na SQL server. To zjednodušuje jeho použitie vo webových farmách, zvyšuje výkon a umožňuje stav zachovať i pri reštarte serveru.

## 2.3 Microsoft SQL Server

Pri vývoji dynamických webových stránok je nutné informácie niekde uschovávať a zase ich zobrazovať. Riešení je viacero. Dáta môžeme uschovávať v rôznych externých súboroch ako XML, XLS a ďalších, alebo na nejakom databázovom serveri.

Microsoft ponúka riešenie pre oba spôsoby a to buď možnosť využiť jeden z plnohodnotných databázových serverov ako Microsoft SQL Server 2005 (MS SQL), alebo jeho Express Edition, ktorá umožňuje vytváranie databáz vo forme externých súborov s príponou MDF.

MS SQL server je komplexný programový relačný databázový systém. Skladá sa z viacerých častí, ktoré úzko spolupracujú navzájom i s ďalšími systémami (webový server, mailový server apod.). Možno ho prevádzkovať v jednoduchých inštaláciách, ale aj v serverových farmách (clustering) a distribuovaných aplikáciách s replikáciou dát.

Podrobnejšie informácie možno vyhľadať vid' literatúra [4].

## 2.4 Vývojové prostredie

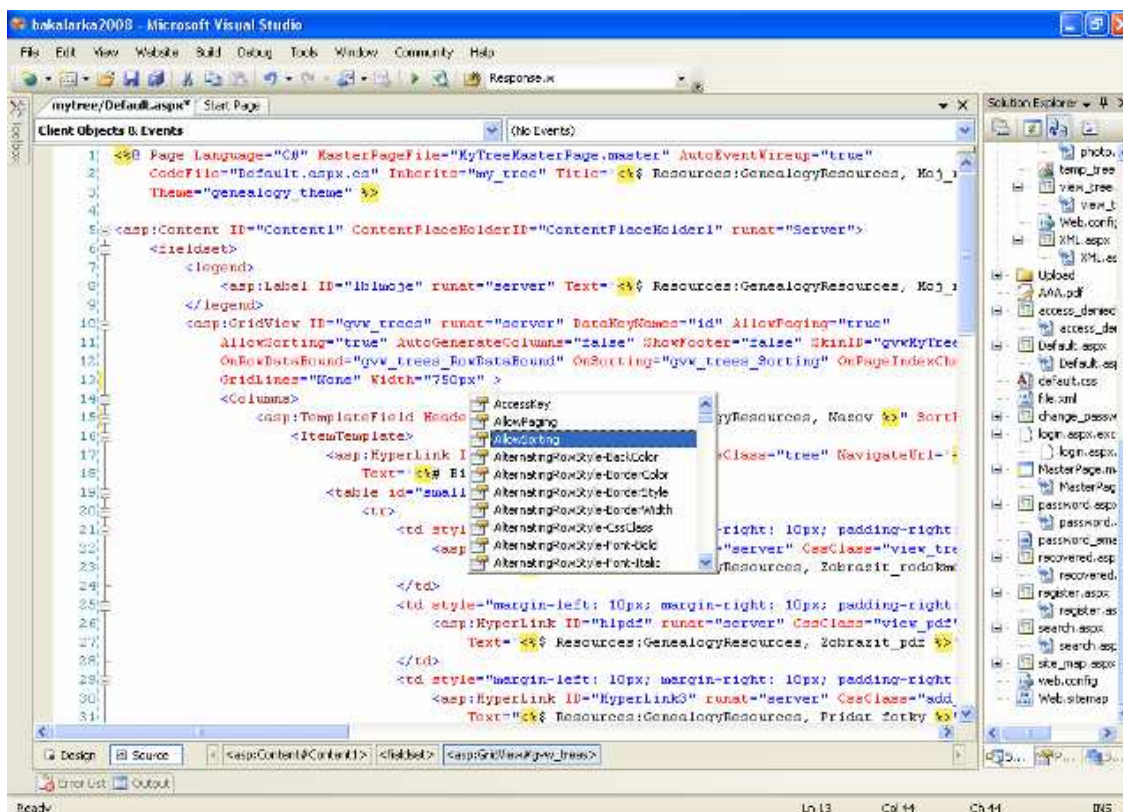
Pre zefektívnenie vývoja aplikácií či už webových alebo klasických GUI aplikácií, zohráva veľkú úlohu aj výber vývojového prostredia. Firma Microsoft navrhla vývojové prostredie Microsoft Visual Studio, ktoré je priam stvorené na vývoj aplikácií v Microsoft .NET.

Verzia Microsoft Visual Studio 2005 pri inštalácii umožňuje priamo nainštalovať aj databázový server Microsoft SQL Server 2005 Express Edition. Existuje aj edícia priamo určená pre vývoj webových aplikácií – Visual Web Developer 2005.

### 2.4.1 Microsoft Visual Studio

Microsoft Visual Studio je hlavným integrovaným vývojovým prostredím firmy Microsoft. Dá sa využiť na vývoj klasických konzolových, GUI i webových aplikácií pre všetky podporované platformy Microsoft Windows, Windows Mobile, Microsoft .NET Framework, Microsoft .NET Compact Framework a Microsoft Silverlight.

Visual Studio obsahuje editor kódu podporujúci IntelliSense a Code Refactoring, ktorý sprehradňuje kód vhodným formátovaním, debugger a ďalšie vstavané utility pre návrh GUI aplikácií, návrh webdesign-u či databázovej schémy. Užívateľ si môže mnohé ďalšie doplnky stiahnuť na Internete vo forme plug-in súborov a jednoducho nainštalovať.



Obr. 2: Ukážka IntelliSense

### 3 Analýza a návrh

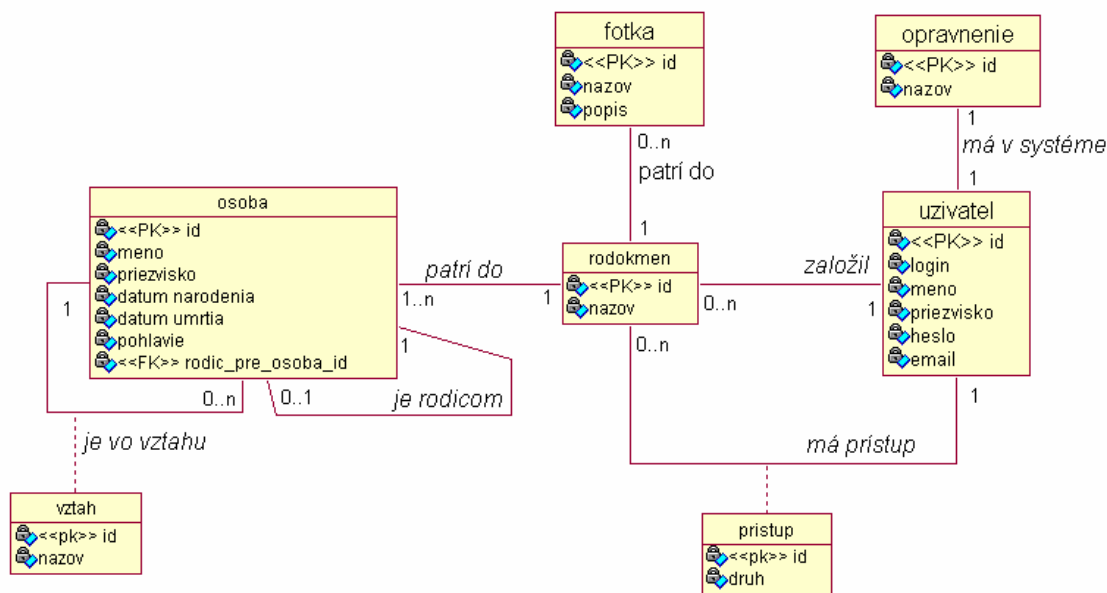
Táto kapitola popisuje návrh a vývoj informačného systému na základe požiadaviek. Podrobnejšie sa zaoberá dátovým modelom.

#### 3.1 Špecifikácia požiadaviek

Úlohou bolo vytvoriť genealogický webový informačný systém s viac vrstvou architektúrou pre správu rodinných genealogických informácií vrátane správy užívateľov, ich rolí a oprávnení v systéme. Informačný systém má umožňovať generovanie výstupu vo forme obrázkov a PDF dokumentov. Tento informačný systém bude umožňovať tiež uchovávať fotografie k jednotlivým rodokmeňom, vytvárať a spravovať pozvánky pre jednotlivé rodokmene. Takto bude môcť zakladateľ rodokmeňa prizvať ďalších ľudí a udeliť im rôzne úrovne oprávnení k budovaniu rodokmeňa.

#### 3.2 Dátový model

Pri návrhu dátového modelu som musel zvážiť veľa aspektov a navrhnuť ho tak, aby systém, ktorý ho bude využívať, dokázal splniť požiadavky. Tento dátový model možno vidieť na obrázku 3.



Obr. 3.: Dátový model

### 3.2.1 Popis objektov

Medzi najhlavnejšie a najdôležitejšie objekty tohto modelu patrí objekt *osoba*.

Objekt *osoba* uchováva informácie o členoch jednotlivých rodokmeňov. Obsahuje základné, v rodokmeňoch najbežnejšie uvádzané informácie o konkrétnej osobe, teda meno, priezvisko, dátum a miesto narodenia a úmrtia. Každá osoba môže byť v určitom vzťahu k inej osobe. Systém bude uchovávať základné rodičovské vzťahy typu matka, otec a vzťahy vytvorené samotnými užívateľmi. Každá osoba patrí do nejakého rodokmeňa.

Objekt *rodokmen* uchováva informácie o vytvorených rodokmeňoch. Ku každému rodokmeňu možno definovať prístup. Toto je úlohou objektu *prístup*.

Objekt *prístup* špecifikuje úrovne prístupu k rodokmeňu. Definoval som štyri úrovne prístupu, ktoré definujú rozsah operácií, ktoré môže konkrétny užívateľ s daným rodokmeňom vykonávať.

	prezeranie, generovanie obrázkov, PDF, XML	správa a pridávanie fotografií	vytváranie nových pozvánok	správa rodokmeňa, pridávanie a editácia osôb	mazanie rodokmeňa, mazanie osôb z rodokmeňa
Host'	áno	áno	nie	nie	nie
Prispievateľ	áno	áno	áno	nie	nie
Editor	áno	áno	áno	áno	nie
Vlastník	áno	áno	áno	áno	áno

Tab.1: Rozdelenie sprístupnených operácií k rodokmeňu podľa druhu oprávnenia

Objekt *fotka* uchováva informácie o fotkách priradených k jednotlivým rodokmeňom.

## 4 Implementácia

Táto kapitola sa zaoberá transformáciou ER diagramu a riešením správy užívateľských účtov využitím API Membership. Ďalej popisuje generovanie obrázkov vývodu, tvorbu pozvánok a export do PDF súboru využitím open-source knižnice iTextSharp.

### 4.1 Užívateľské účty

Pre správu užívateľských účtov a rolí v samotnom informačnom systéme som zvolil aplikačné rozhranie Membership, ktoré je súčasťou Microsoft.NET Framework-u práve od verzie 2.0.

#### 4.1.1 Membership API

Jednou z novinek v ASP.NET 2.0 je aplikačné rozhranie na správu užívateľských účtov a rolí nazývané Membership. Tento model prináša množstvo aplikačných služieb a ovládacích prvkov, ktoré Membership priamo využívajú. Poskytujú bežné a často programátormi vytvárané služby pre správu užívateľov a ich oprávnení.

Hoci Membership prichádza s prepracovanou funkčnosťou pripravenou ihneď k použitiu, programátor si ju môže doladiť podľa vlastných potrieb. Funguje v súčinnosti s ďalšou novou vlastnosťou nazývanou Role Manager.

Role Manager poskytuje infraštruktúru tvorby rolí prideľovania rolí užívateľom. Keď sú Membership a Role Manager použité v súčinnosti s autentifikáciou založenou na formulároch, ASP.NET 2.0 poskytuje priamu podporu pre tvorbu, autentifikáciu a autorizáciu používateľov.

Membership aj Role Manager boli navrhnuté pre použitie modelu poskytovateľov služieb. Títo poskytovatelia abstrahujú biznis-logiku konkrétnej vlastnosti a fyzický dátový zdroj pre triedy. Role Manager i Membership sú dodané s poskytovateľmi pre Microsoft SQL Server. Membership je dodaný spolu s poskytovateľom pre Active Directory a pre Active Directory Application Mode (ADAM). Role Manager je dodaný spolu s poskytovateľom pracujúcim s vlastnosťou Authorization Manager, dostupnou vo Windows Server 2003.

Prihlasovacie ovládacie prvky tvoria množinu ovládacích prvkov, ktoré poskytujú zjednotené používateľské rozhranie pre autentifikáciu a autorizáciu. Tieto ovládacie prvky používajú funkcionality vlastností Membership, Role Manager a autentifikáciu založenú na formulároch.

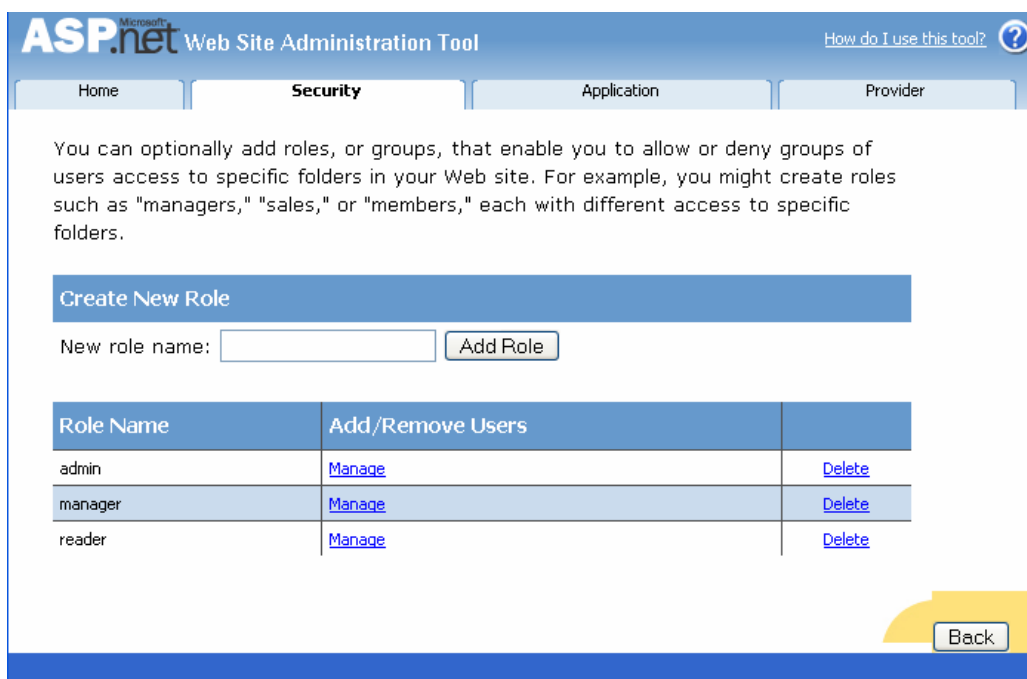
ASP.NET Membership vytvára a rozhranie pre:

- Vytváranie užívateľov a hesiel.
- Ukladanie informácií o užívateľoch v Microsoft SQL Serveri, Active Directory, ale aj v iných alternatívnych formách (napr. XML).
- Môžete tvoriť svojich vlastných poskytovateľov a nastaviť ich, aby pracovali buď s vlastnosťou Membership, alebo s vlastnosťou Role Manager. Stránky, ktoré používajú vlastnosti Membership a Role Manager, budú pracovať bez zmeny aj s Vašimi vlastnými poskytovateľmi.
- Autentifikáciu a autorizáciu užívateľov programátorsky alebo využitím predpripravených ovládacích prvkov, ktoré nepotrebujú ďalšie nastavovanie parametrov, alebo len minimálne.
- Kompletnú správu hesiel, teda vytváranie, zmenu prípadnú obnovu. V závislosti od nastavení vlastností Membership-u, Membership môže automaticky resetovať, deje sa tak v prípade, keď užívateľ správne odpovie na otázku, ktorú zadal pri registrácii. Heslo je obnovené a zaslané formou emailu na adresu zadanú pri registrácii.

Údaje sú čerpané z publikácie vid' literatúra [5] a zo stránok Microsoft, vid' literatúra [6].

## 4.1.2 Správa rolí

Pri návrhu informačného systému som pomocou nástroja VAT, ktorý je súčasťou Visual Studia 2005, definoval 3 druhy oprávnení – rolí.



The screenshot shows the 'ASP.NET Web Site Administration Tool' interface. The 'Security' tab is active, displaying instructions on adding roles and a 'Create New Role' form. Below the form is a table listing existing roles: 'admin', 'manager', and 'reader'. Each role has 'Manage' and 'Delete' links. A 'Back' button is visible in the bottom right corner.

Role Name	Add/Remove Users	
admin	<a href="#">Manage</a>	<a href="#">Delete</a>
manager	<a href="#">Manage</a>	<a href="#">Delete</a>
reader	<a href="#">Manage</a>	<a href="#">Delete</a>

Obr. 4: Nástroj VAT



Na základe týchto rolí sú registrovaným užívateľom sprístupnené rôzne možnosti práce s informačným systémom. Podrobnejšie to popisuje nasledujúca tabuľka.

názov role	Popis
reader	každému novo zaregistrovanému užívateľovi je automaticky pridelená rola "reader", čím sa užívateľovi sprístupní možnosť vytvárania vlastných rodokmeňov.
manager	títo užívatelia majú okrem iného prístup a možnosť editovania a mazania rodokmeňov a osôb v rodokmeňoch
admin	najvyššia úroveň oprávnenia, užívateľ v tejto roli môže okrem vytvárania, spravovania, mazania rodokmeňov a osôb i spravovať užívateľov a meniť ich role v systéme a odblokovat' z nejakého dôvodu zablokovaných užívateľov

Tab. 2: Rozdelenie rolí

Každému novo zaregistrovanému užívateľovi informačného systému je automaticky pridelená rola *reader*. Týmto je sprístupnená možnosť tvorby vlastných rodokmeňov.

Nasledujúca časť kódu demonštruje ako Membership umožňuje programátorsky dotvárať funkčnosť svojho API. Samotné pridelovanie role sa deje v poslednej fázy registrácie nového užívateľa.

```

/// <summary>
/// add the role "reader" to each new registered user
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void CreateUserWizard1_CreatedUser(object sender, EventArgs e)
{
    Membership.FindUsersByName(CreateUserWizard1.UserName);
    if (Roles.RoleExists("reader"))
    {
        Roles.AddUserToRole(CreateUserWizard1.UserName, "reader");
    }
}

```

Zdroj. kód 1: Ukážka pridelenia role pri registrácii užívateľa

Administrácia užívateľov je podobná ako v rozhraní VAT. Umožňuje rýchle vyhľadanie konkrétneho užívateľa. Vyhľadávať možno podľa reťazca obsiahnutého v logine, alebo podľa reťazca obsiahnutom v emailovej adrese.

The screenshot shows the 'UŽIVATELIA' (Users) section of the 'GENEALOGICKÝ INFORMAČNÝ SYSTÉM'. At the top, there is a navigation menu with 'ÚVOD', 'MOJE RODOKMENE', 'RODOKMENE', 'OSOBY', and 'UŽIVATELIA'. A language dropdown is set to 'Slovak'. A user is logged in as 'admin!', with links for 'Zmena hesla' and 'Odhlásiť'. The main content area shows 'Užívateľia' with statistics: 'Všetkých užívateľov: 9' and 'Online: 1'. A search bar is set to search by 'Login' with the value 'martin'. Below this is a table of users.

Login	E-mail	Založený	Naposledy prihlásený	Online	Blokovaný
martin xrepta00@stud.fit.vutbr.cz		22. 4. 2008	22. 4. 2008 16:41:47	<input type="checkbox"/>	<input type="checkbox"/>

Copyright © 2008 Martin Repta

Obr. 5: Ukážka vyhľadávania osôb v administrácii

Každý užívateľ má v systéme pridelenú len a práve len jednu rolu.

## 4.2 Rodokmeň

Mnou navrhnutý a vytvorený informačný systém umožňuje tvorbu rodokmeňov typu vývod, teda je určený k hľadaniu a zaznamenávaniu predkov danej osoby. Pri budovaní vývodu sa postupuje smerom do minulosti. Hľadáme rodičov svojich predkov a postupujeme do minulosti.

Každému registrovanému užívateľovi je po prihlásení sa do systému sprístupnená časť „Moje rodokmene“. Po vstupe do tejto sekcie sa užívateľovi zobrazí zoznam rodokmeňov, ku ktorým má prístup. Sú to rodokmene vytvorené samotným užívateľom alebo tie, ku ktorým ho niekto pozval.

Ak užívateľ doposiaľ nezaložil žiadny rodokmeň a ani nebol pozvaný do nejakého rodokmeňa od iného užívateľa, zobrazí sa mu správa oznamujúca, že si môže rodokmeň založiť.

The screenshot shows the main interface of the 'GENEALOGICKÝ INFORMAČNÝ SYSTÉM'. At the top, there is a navigation bar with links: ÚVOD, ZALOŽIŤ NOVÝ RODOKMEŇ, MOJE RODOKMENE, ADMINISTRÁCIA, and HĽADANIE. A language dropdown menu is set to 'Slovak'. A welcome message 'Vitajte admin!' is displayed with links for 'Zmena hesla' and 'Odhlásiť'.

The 'Moje rodokmene' section is active, showing a table of family trees. The table has three columns: 'Názov', 'Oprávnenie', and 'Posledná modifikácia'. Each row includes a star icon, a name, and a set of action icons (Zobrazit rodokmeň, Zobrazit PDF, Pridať fotky, Pozvať rodinu, Spravovať rodokmeň).

Názov	Oprávnenie	Posledná modifikácia
★ Moj prvý rodokmeň Zobrazit rodokmeň   Zobrazit PDF   Pridať fotky   Pozvať rodinu   Spravovať rodokmeň	Vlastník	17. 4. 2008 21:30:53
★ Test Zobrazit rodokmeň   Zobrazit PDF   Pridať fotky   Pozvať rodinu   Spravovať rodokmeň	Editor	14. 4. 2008 20:23:13
★ testovací Zobrazit rodokmeň   Zobrazit PDF   Pridať fotky   Pozvať rodinu	Prispievateľ	16. 4. 2008 12:59:03
★ Martinov rodokmeň Zobrazit rodokmeň   Zobrazit PDF   Pridať fotky	Host	22. 4. 2008 20:45:42

Copyright © 2008 Martin Repta

Obr. 6: Ukážka hlavného rozhrania pre zakladanie a správu rodokmeňov

## 4.2.1 Založenie rodokmeňa

Pri zakladaní rodokmeňa napomáha ovládací prvok sprievodca, tzv. *Wizard*, ktorému som definoval jednotlivé kroky. Použitím validačných prvkov zabezpečujem, aby užívateľ zadal aspoň základné údaje ako meno, priezvisko počiatocnej osoby a tiež matky i otca, po prejdení prvých troch krokov, musí užívateľ zadať názov rodokmeňa a má možnosť zvoliť si, či bude rodokmeň verejný. Táto vlastnosť rodokmeňa znamená, že údaje osôb v ňom figurujúcich budú sprístupnené k vyhľadávaniu registrovaným i neregistrovaným užívateľom v časti „Hľadanie“.

The screenshot shows the 'GENEALOGICKÝ INFORMAČNÝ SYSTÉM' website. At the top, there is a navigation bar with links: ÚVOD, ZALOŽIŤ NOVÝ RODOKMEŇ, MOJE RODOKMENE, ADMINISTRÁCIA, and HĽADANIE. A language dropdown is set to 'Slovak'. A user login area says 'Vitajte manager!' with links for 'Zmena hesla' and 'Odhlasit'. Below the navigation bar, a breadcrumb trail reads 'Úvod > Moje rodokmene > Založiť rodokmeň'. The main content area features a 'Založiť nový rodokmeň' button and a form with the following fields: 'Meno:' (Martin), 'Priezvisko:' (Repta), 'Pohlavie:' (Male), 'Dátum narodenia:', 'Miesto narodenia:', 'Dátum úmrtia:', and 'Miesto úmrtia:'. A 'Next' button is located at the bottom right of the form. The footer contains the text: 'Copyright © 2008 Genealogy. Designed by Free CSS Templates. Programmed by Martin Repta'.

Obr. 7: Sprievodca pre založenie rodokmeňa

Po prejdení celého sprievodcu je užívateľovi zobrazené rozhranie, cez ktoré môže rodokmeň ďalej rozširovať definovaním ďalších rodičovských vzťahov prípadne má možnosť definovať si vzťahy vlastné. Koreňová osoba (tá pre ktorú vývod robíme) nemôže byť zmananá. Užívateľ však môže zmazať celý rodokmeň naraz. Užívateľ si taktiež môže dať rodokmeň priamo vykresliť prípadne vyexportovať osoby do XML súboru či PDF dokumentu.

## 4.2.2 Tvorba pozvánok k rodokmeňu

Vlastník rodokmeňa môže vytvoriť akoby „pozvánku“ pre svoj rodokmeň. Na základe registračnej emailovej adresy udelí oprávnenie nejakému inému užívateľovi. Užívateľ podľa druhu prideleného oprávnenia môže s rodokmeňom vykonávať rôzne operácie. Tieto operácie som popisoval podrobnejšie v tab. 1.

Pozvánku s oprávnením „Vlastník“ má len zakladateľ rodokmeňa. Nie je možné takúto pozvánku vytvoriť pre iného užívateľa a ani ju zmazať. Pozvaný užívatelia s oprávnením „Prispievateľ“ a „Editor“ môžu pozvánky vytvárať a pridelať aj sami, ale maximálne s oprávnením aké majú oni sami. Teda napr. užívateľ s pozvánkou „Prispievateľ“, nemôže vytvoriť pozvánku s vyšším oprávnením („Editor“).

## 4.2.3 Generovanie obrázka rodokmeňa

Na začiatku generovania je nutné určiť rozstup prvých dvoch uzlov tak, aby najvrchnejšia úroveň, teda tá historicky najstaršia, bola stále v dostatočnom rozstupe. Definoval som si preto rozstup 50 pixelov medzi jednotlivými uzlami najstaršej línie. Tento rozstup je dostatočne široký nato aby boli mená osôb čitateľné a neprekrývali sa. Bolo teda nutné vymyslieť vzťah, ktorý vypočíta rozstup prvých dvoch uzlov tak, aby sa nasledujúce línie neprekrývali a pritom v najvrchnejšej z nich bol rozstup medzi jednotlivými uzlami 50 px. Tento vzťah teda závisí od hĺbky rodokmeňa, alebo inak povedané od počtu generácií. Rozstup som nazval „delta“. Vzťah pre výpočet je nasledovný:

$$\text{delta} = 2^{(\text{deep} - 1)} * \text{width}$$

Vysvetlivky:

delta – šírka rozstupu uzlov prvej línie

deep – hĺbka (počet generácií) rodokmeňa

width – šírka rozstupu medzi uzlami v najvrchnejšej línii

Pri generovaní každej ďalšej úrovne sa delta vždy znižuje o polovinu.

### 4.2.3.1 Výpočet hĺbky (počtu generácií)

Algoritmus pre výpočet hĺbky rodokmeňa je rekurzívnou funkciou, pri ktorej sa zisťuje, ktorá z vetiev je hlbšia. Z tabuľky *rodicia* sa vyhľadajú matka a otec pre aktuálnu osobu. Ak takýto údaj existuje, výsledok sa zvýši o 1 a znova sa zisťuje maximum z hĺbky stromu pre matku i otca. Po prejdení všetkých generácií sa ako výsledok vráti hĺbka rodokmeňa. Detailnejšie vid' zdroj. kód 2.

```

/// <summary>
/// This method returns deep of tree
/// </summary>
/// <param name="oid">id of person</param>
/// <param name="rid">id of family tree</param>
/// <returns></returns>
protected int Deep(int oid, int rid)
{
    DataTable t = new DataTable(); //create new data table
    t = rodicia.GetData(oid, rid); //get id of mother and father
    if (t.Rows.Count == 0)
    {
        t.Dispose();

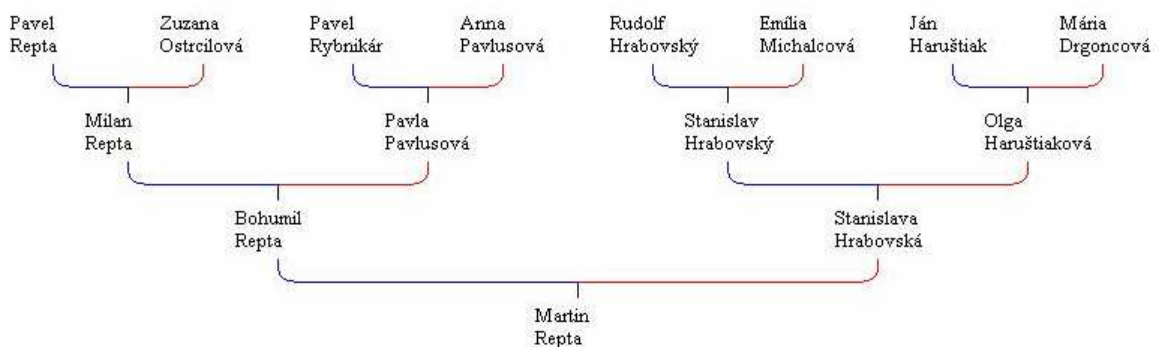
        return -1;
    }
    else
    {
        int id_o = Convert.ToInt32(t.Rows[0]["otec_id"].ToString());
        int id_m = Convert.ToInt32(t.Rows[0]["matka_id"].ToString());
        t.Dispose(); //dispose allocated resources

        return (1 + Max(Deep(id_o, rid), Deep(id_m, rid)));
    }
}

```

Zdroj. kód 2: Výpočet hĺbky rodokmeňa

Na generovanie rodokmeňa som si vytvoril rekurzívnu metódu, ktorej na počiatku predám id koreňovej osoby, ktoré zaznamenávam pri vytváraní rodokmeňa do tabuľky rodokmen. Pre túto osobu vyhlľadám id otca i matky z tabuľky rodicia. Pre každú z nájdených osôb sa metóda rekurzívne zavolá znova. Prebieha vykresľovanie a generovanie obrázka. K uzlu matky, je pre lepšie rozpoznanie, vedená krivka červenej farby, k uzlu otca farby modrej (viď obr. 8).



Obr. 8: Vygenerovaný rodokmeň

## 4.3 Generovanie PDF

Súčasťou zadania bolo, aby systém umožňoval generovanie PDF dokumentov. Hľadal som preto dostupný a pritom open-source nástroj, ktorý dokáže on-line generovať PDF dokumenty. Na stránkach portálu Sourceforge (<http://sourceforge.net/>) som našiel open-source knižnicu iTextSharp.

### 4.3.1 Knižnica iTextSharp

Táto knižnica na generovanie on-line PDF dokumentov je odnožou knižnice iText, ktorá je pôvodne určená pre Javu. iTextSharp je celá naprogramovaná v C#.

Umožňuje napr.:

- Dynamické generovanie PDF i RTF súborov z XML súboru či databáze
- Číslovanie strán
- Pridávania bookmark-ov, watermark-ov
- Spájať a rozdeľovať strany
- Automatické vyplňovanie formulárov
- Digitálne podpisy

Je implementovaná ako assembly knižnica, teda jednoduchým vložením do adresára *bin* sa sprístupní jej API.

Jej API je intuitívne a na jej domovských stránkach je sprístupnený rozsiahly tutoriál, po ktorého preštudovaní som bol schopný vytvoriť požadovanú časť. Problém ale nastal s diakritikou.

#### 4.3.1.1 Fonty

Pretože fonty prístupné priamo v knižnici nepodporujú slovenskú či českú diakritiku, musel som tento problém vyriešiť. Vytvoril som si vlastný font (viď zdroj. kód 3), ktorého základ som pôvodne skopíroval z priečinka *Fonts* vo Windows. Použil som font *Times New Roman*.

```
//create new font from external file
string fontpath = Server.MapPath("../images/times.ttf");
string fontbdpath = Server.MapPath("../images/timesbd.ttf");

BaseFont TNR = BaseFont.CreateFont(fontpath, BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
BaseFont TNRB = BaseFont.CreateFont(fontbdpath, BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);

Font times = new Font(TNR, 10);
Font timesHB = new Font(TNRB, 10);
```

Zdroj. kód 3: Časť kódu z generovania PDF - vytvorenie vlastného fontu z ext. súboru

## 4.4 Generovanie XML

Do informačného systému som pridal i možnosť, aby si užívatelia mohli informácie o osobách v ich vlastnom alebo sprístupnenom rodokmeni vygenerovať do XML súboru. Týmto bude možné s informáciami prípadne pracovať i v iných aplikáciách.

Na prácu s XML súbormi je v Microsoft .NET samostatný namespace System.XML.

### 4.4.1 Namespace System.XML

Je to skupina tried reprezentujúcich praktickosť v každom aspekte práce s XML. Tieto triedy poskytujú metódy na manipuláciu, vytváranie a úpravu XML dokumentov a dát.

Namespace System.XML podporuje prácu s XML súbormi založenú na štandardoch. Bližšie informácie o podporovaných štandardoch nájdete v literatúre [8]

### 4.4.2 XML výstup

Pri generovaní XML dokumentu do atribútov uzlu `person` vkladám všetky atribúty objektu osoba diskutovaného v kapitole 3.2.1. Do atribútov je vkladané aj `id` z databázy pre možné využitie v ďalších aplikáciách, tak aby sa dali jednotlivé osoby jednoznačne rozoznať napríklad pri generovaní grafov.

Výstup generovania demonštruje obr. 9.

```
<tree name="Moj prvý rodokmen" public="True">
  <persons>
    <person id="30" firstname="Martin" surname="Repta" gender_male="True" dob="30.8.1985" pob="Trencín" dod="" pod=""/>
    <person id="31" firstname="Bohumil" surname="Repta" gender_male="True" dob="16.9.1951" pob="Nová Lehota" dod="" pod=""/>
    <person id="32" firstname="Stanislava" surname="Hrabovská" gender_male="False" dob="30.1.1963" pob="" dod="" pod=""/>
    <person id="33" firstname="Stanislav" surname="Hrabovský" gender_male="True" dob="30.12.1963" pob="" dod="" pod=""/>
    <person id="34" firstname="Olga" surname="Haruštiaková" gender_male="False" dob="24.5.1939" pob="" dod="" pod=""/>
    <person id="35" firstname="Rudolf" surname="Hrabovský" gender_male="True" dob="17.4.1908" pob="" dod="26.4.1947" pod=""/>
    <person id="36" firstname="Emília" surname="Michalcová" gender_male="False" dob="3.8.1912" pob="" dod="16.10.1990" pod=""/>
    <person id="37" firstname="Michal" surname="Michalec" gender_male="True" dob="25.9.1879" pob="" dod="12.7.1961" pod=""/>
    <person id="132" firstname="Katarína" surname="Kucerová" gender_male="False" dob="" pob="" dod="" pod=""/>
    <person id="139" firstname="Pavel" surname="Rybníkář" gender_male="True" dob="" pob="" dod="" pod=""/>
    <person id="140" firstname="Anna" surname="Pavlusová" gender_male="False" dob="" pob="" dod="" pod=""/>
    <person id="143" firstname="Ján" surname="Repta" gender_male="True" dob="" pob="" dod="25.1.1925" pod="Modrovka"/>
    <person id="144" firstname="Katarína" surname="Pagáčová" gender_male="False" dob="" pob="" dod="" pod="Nová Lehota"/>
  </persons>
</tree>
```

Obr. 9: Ukážka vygenerovaného XML výstupu



## 4.5 Vývoj viacjazyčných informačných systémov v ASP.NET 2.0

Ak chceme budovať webové informačné systémy na vysokej úrovni, ktoré budú prístupné pre širokú verejnosť, podpora rôznych kultúr je základom úspechu. Microsoft .NET Framework 2.0, a tiež už aj jeho predchodca 1.x prichádza s integrovanou štruktúrou pre tvorbu viacjazyčných multikultúrnych aplikácií. CLR podporuje mechanizmus pre vývoja vo všetkých typoch aplikácií, nielen stolných, ale i mobilných a webových. CLR prináša niekoľko tried pre správu a prístup ku zdrojom (tzv. Resources) vo vnútri aplikácie. Tieto triedy sa nachádzajú v mennom priestore System.Resources a System.Globalization.

### 4.5.1 Namespace System.Resources

Obsahuje triedy a rozhrania, ktoré programátorovi pri vývoji aplikácie dovoľujú vytváranie, ukladanie a prácu s rozličnými zdrojmi (Resources). Zdroje sú uložené ako hlavné assembly, alebo ako tzv. satellite assembly.

### 4.5.2 Namespace System.Globalization

Tento menný priestor obsahuje triedy, ktoré definujú informácie závislé na kultúre danej oblasti, teda jazyk, formát dátumu, meny, čísiel atď.

### 4.5.3 Lokalizácia a zdroje

Podpora lokalizácie v ASP.NET 2.0 sa stala pre vývoj aplikácií jednoduchšou ako predtým. Zdroje sú zvyčajne vytvárané pre každú z kultúr, ktoré chceme podporovať, zvlášť. Napr. ak plánujeme podporovať jazykové a kultúrne zvyklosti krajín Slovensko, Anglicko a Tunis (arabsky), mali by sme pre každú kultúru vytvoriť svoj vlastný zdroj. CLR definuje chovanie pre nájdenie zdroja pre špecifickú kultúru. Ak sa zdroj pre danú kultúru nenájde je možno zdefinovať defaultný zdroj.

Visual Studio 2005 podporuje a zjednodušuje vytváranie zdrojov pre jazykové kultúry. V prípade, že chceme lokalizovať len určitú časť svojho webu, pre vytvorenie zdrojov použijeme menu. Po prepnutí do dizajnu stránky, kliknutím na Tools - Generate Local Resources sa do projektu pridá adresár *App\_LocalResources* a slovník pre danú stránku (viď obr. 10).

The screenshot shows a window titled 'admin/App\_Loc...ons.aspx.resx'. The window has a menu bar with 'Strings', 'Add Resource', and 'Remove Resource'. Below the menu bar is a table with two columns: 'Name' and 'Value'.

Name	Value
ListItemResource3.Text	Male
ListItemResource3.Value	True
ListItemResource4.Text	Female
ListItemResource4.Value	False
InkAddResource1.Text	Insert
InkAddResource1.ToolTip	
InkCancelResource1.Text	Cancel
InkCancelResource1.ToolTip	
InkDeleteResource1.Text	Delete
InkDeleteResource1.ToolTip	
InkEditResource1.Text	Edit

Obr. 10: Vytvorený slovník

#### 4.5.3.1 Použitie lokalizácie

Lokalizačné výrazy sú identifikované podľa meta:resourceKey atribútu pre daný ovládací prvok.

```
<asp:Label ID="LegendFirstname" runat="server"
Text="Firstname:" meta:resourcekey="LabelResource1" />
```

Zdroj. kód 4: Použitie meta:resourceKey

### 4.5.4 Globalizácia a zdroje

Ak chceme globalizovať nami vyvíjaný informačný systém, musíme si do projektu pridať adresár App\_GlobalResources. Do neho umiestnime zdroje – slovníky, z ktorých bude CLR čerpať. V Solution Explorer-i klikneme pravým tlačítkom na adresár App\_GlobalResources a zvolíme možnosť pridať nový súbor, z ponúkaných možností vyberieme „Resource file“. Takto nám vznikne akoby slovník, do ktorého budeme pridávať slová pre lokalizáciu. Každý z týchto Resource súborov je pomenovaný rovnakým názvom, po ktorom nasleduje “.” a prípona podľa kultúry, pre ktorú slovník vytvárame. Teda napr. pre Slovensko je to ”sk-SK“, pre USA je to ”en-US“. Kompletný zoznam týchto prípon možno nájsť v literatúre [9]. Názov súboru slovníka potom vyzerá napr. takto: GenealogyResources.sk-SK.resx

## 4.6 Trieda BasePage

Kvôli zefektívneniu práce a opakujúcim sa častiam rovnakého programového kódu vo viacerých súboroch, som si vytvoril triedu BasePage. Táto programová trieda obsahuje najviac používané metódy a vlastnosti v mojom informačnom systéme. Preto aby ostatné triedy mohli dediť jej chovanie, umiestnil som ju do adresára *App\_code*. Všetky triedy, ktoré dedia jej chovanie, potom môžu pracovať s protected metódami a vlastnosťami (properties) v BasePage. Táto trieda obsahuje najmä metódy pre globalizáciu.

V hornej časti užívateľského rozhrania sa nachádza ovládací prvok na výber jazyka tzv. DropDownList. Užívateľ si môže zvoliť z ponúkaných jazykov a podľa toho sa globalizácia nastaví. Tento proces sa vykoná vždy po zovaní *postback* udalosti. Bolo potrebné prepísať chovanie metódy *InitializeCulture* (viď zdroj. kód 5) tak, aby sa kultúrne zvyklosti nastavovali podľa výberu užívateľa, teda vždy po zovaní *postback*-u. Problémom však je, že pred samotným zobrazením stránky ešte nepoznáme názov samotného ovládacieho prvku, takže nemôžeme zistiť vybraný jazyk.

Túto vlastnosť možno však vyriešiť použitím prvkov z Form collection. Po zvolení jazyka je do Session uložený parameter definujúci vybraný jazyk, nato aby sa nastavenie zachovalo i po prechode na inú časť informačného systému.

```
/// <summary>
/// BasePage for the common functionality in all
/// the web pages of the site.
/// </summary>
public class BasePage: Page
{
    // The id of the dropdownlist for change of language
    public const string ddlLang = "ctl100$ddlLang";

    //name of postback event target
    public const string PostBackEventTarget = "__EVENTTARGET";

    /// <SUMMARY>
    /// Overriding the InitializeCulture method to set the user selected language,
    /// this method is called much earlier in the Page lifecycle
    /// </SUMMARY>
    protected override void InitializeCulture()
    {
        //check if Postback
        if (Request[PostBackEventTarget] != null)
        {
            //get control id of control that caused postback
            string controlId = Request[PostBackEventTarget];

            if (controlID.Equals(ddlLang))
            {
                //save user selection of language
                if (Request.Form[Request[PostBackEventTarget]].ToString() != "-1")
                {
                    Session["MyUICulture"] = Request.Form[Request[PostBackEventTarget]].ToString();
                }
            }
        }
        if (Session["MyUICulture"] != null)
        {
            //set culture info into thread
            SetCultureThread(Session["MyUICulture"].ToString());
        }
    }
}
```

Zdroj. kód 5: Ukážka uloženia parametra pre nastavenie jazykovej kultúry do Session

Z metódy InitializeCulture je po nastavení Session volaná metóda SetCultureThread, ktorou uloží a nastavím jazykové a kultúrne zvyklosti do vlákna stránky.

```
/// <Summary>
/// set current CurrentUICulture and CurrentCulture into thread
/// </Summary>
/// <PARAM name="name"></PARAM>
protected void SetCultureThread(string lang)
{
    Thread.CurrentThread.CurrentUICulture = new CultureInfo(lang);
    Thread.CurrentThread.CurrentCulture = new CultureInfo(lang);

    //save culture info into session for use across the pages
    Session["MyUICulture"] = lang;
}
```

Zdroj. kód 6: Nastavenie CurrentUICulture a CurrentCulture

K fungovaniu globalizácie je potrebné, aby všetky stránky, ktoré chceme globalizovať, dedili od samotnej BasePage. Potom je ešte potrebné prepísať chovanie metódy OnInit stránok tak, aby volali OnInit metódu svojej rodičovskej triedy. Tým sa vyvolá nastavenie kultúrnych a jazykových zvyklostí.

## 5 Záver

Cieľom tejto bakalárskej práce bolo vytvoriť genealogický informačný systém v ASP.NET 2.0. Preštudoval som si spôsob tvorby rodokmeňov a skúmanie pôvodu osôb z genealogického hľadiska. Skúmanie a hľadanie svojich predkov bola veľmi zaujímavá úloha, ale časovo náročná, pretože informácie sú na matrikách väčšinou uložené vo forme knižných zápiskov a údaje sú mnohokrát zle čitateľné. Nájst' informácie dalo veľa námahy, ale výsledný efekt priniesol uspokojenie. Preštudoval som stovky strán, väčšinou anglickej literatúry, a podrobnejšie som sa zoznámil s tvorbou viacvrstvových webových aplikácií v ASP.NET 2.0. Spoznal som jazyk C#, ktorý ma veľmi zaujal a chcel by som sa v ňom aj naďalej zlepšovať a zdokonaľovať. Informačný systém splnil požadované ciele a dokáže generovať rodokmene či PDF dokumenty. Systém som dôkladne otestoval na tvorbe niekoľkých rodokmeňoch zo všetkých oblastí jeho funkcionality. V budúcnosti by som chcel naďalej pracovať na jeho vylepšeniach.

### 5.1 Vylepšenia systému

V budúcnosti by som chcel funkcionality systému naďalej rozširovať. Veľmi ma nadchla možnosť pripojiť tento systém s niektorou s verejne prístupných databáz. Najme v USA a Kanade je možno nájsť veľa informácií o svojich predkoch, pretože veľa ľudí emigrovalo do zámoria za prácou s nádejou lepšieho života. Ak by sa systém dokázal pripojiť na takéto databáze a vyhľadávať v nich. Myslím že by to veľmi pomohlo a uľahčilo hľadanie svojich predkov.

Chcel by som pridať i import dát z externých súborov ako XML príp. CSV. Formát XML sa stáva čoraz viac populárnejším a práca s ním nie je náročná.

Ďalšie možnosti vylepšovania by som videl využitím AJAX, čím by sa ešte zmenšil tok prenášaných dát na klientský prehliadač a tým by sa systém zrýchlil.

Vylepšil by som a rozšíril schopnosti generovania rodokmeňa tak, aby užívateľ si mohol zvoliť vrstvy svojich vlastných vzťahov, ktoré chce v rodokmeni zobrazovať.

# Literatúra

- [1] *Microsoft .NET Framework* [online]. 2008 [cit. 2008-05-03]. Dostupný z WWW: <<http://www.microsoft.com/net/default.aspx>>..
- [2] Homer, A., Sussman, D. *ASP.NET 2.0 Illustrated*. Addison Wesley Professional 2006.
- [3] Duthie, G., MacDonald, M. *ASP.NET in a Nutshell, 2nd Edition*. O'Reilly 2003.
- [4] *Microsoft SQL Server* [online]. 2008 [cit. 2008-05-03]. Dostupný z WWW: <<http://www.microsoft.com/sql/default.msp>>.
- [5] Schackow, S., *Professional ASP.NET 2.0 Security, Membership and Role Management*. Wrox 2006.
- [6] *MSDN: Membership* [online]. 2008 [cit. 2008-05-03]. Dostupný z WWW: <<http://msdn.microsoft.com/en-us/library/yh26yfzy.aspx>>.
- [7] REPA, Marián. Po meči a po praslici. *IŽurnál: Naj zo Žurnálu* [online]. 2008 [cit. 2008-05-03].
- [8] *MSDN: System.Xml Namespace* [online]. 2008 [cit. 2008-05-03]. Dostupný z WWW: <[http://msdn.microsoft.com/en-us/library/system.xml\(vs.71\).aspx](http://msdn.microsoft.com/en-us/library/system.xml(vs.71).aspx)>.
- [9] *MSDN: CultureInfo Class* [online]. 2008 [cit. 2008-05-03]. Dostupný z WWW: <[http://msdn.microsoft.com/en-us/library/system.globalization.cultureinfo\(vs.71\).aspx](http://msdn.microsoft.com/en-us/library/system.globalization.cultureinfo(vs.71).aspx)>.
- [10] *Microsoft .NET Framework 3.5 Beta 2* [online]. 2007 [cit. 2008-05-03]. Dostupný z WWW: <<http://palprince.com/software/microsoft-net-framework-35-beta-2/>>.

# Zoznam príloh

Príloha 1. CD so zdrojovými textami

## Súborová štruktúra CD

ebook/xrepta00.pdf	elektronická verzia textu bakalárskej práce
admin/AdminMasterPage.master	obsahuje dizajn administratívneho rozhrania
admin/Default.aspx	hlavná stránka administrácie
admin/edit_user.aspx	umožňuje úpravy užívateľov
admin/persons.aspx	rozhranie pre správu osôb pridaných do rodokmeňov
admin/trees.aspx	rozhranie pre správu rodokmeňov
admin/users.aspx	rozhranie pre vyhľadávanie užívateľov
admin/Web.config	konfiguračné nastavenia
App_code/BasePage.cs	obsahuje najčastejšie používané metódy a vlastnosti
App_code/genealogy.xsd	obsahuje dátové objekty pre prácu s databázou
App_GlobalResources	obsahuje slovníky pre globalizáciu
App_Themes	obsahuje vytvorené témy, ktoré využívajú niektoré použité ovládacie prvky
Bin/itextsharp.dll	open-source knižnica využívaná na generovanie PDF dokumentov
images/	obsahuje obrázky použité v užívateľskom rozhraní
mytree/create_tree.aspx	umožňuje vytvorenie rodokmeňa, pridávanie osôb i generovanie PDF dokumentu
mytree/Default.aspx	hlavné rozhranie časti “Moje rodokmene”
mytree/define_relation.aspx	rozhranie na tvorbu a správu užívateľských vzťahov
mytree/invite_to_tree.aspx	rozhranie pre tvorbu a správu pozvánok
mytree/manage_tree.aspx	umožňuje užívateľovi zmeniť základné parametre rodokmeňa

mytree/MyTreeMasterPage.master	obsahuje dizajn rozhrania pre správu a tvorbu rodokmeňov
mytree/photo.aspx	zabezpečuje upload a správu fotografií k rodokmeňu
mytree/view_tree.aspx	generuje obrázok rodokmeňa
mytree/Web.config	konfiguračné nastavenia pre tento adresár
mytree/XML.aspx	zabezpečuje export do XML súboru
Upload/	priečinkov uploadovaných fotografií
access_denied.aspx	zobrazuje bezpečnostné hlásenia v prípade pokusu o neoprávnené preniknutie do systému
Default.aspx	úvodná stránka informačného systému
default.css	kaskádové štýly definujúce vzhľad informačného systému
change_password.aspx	umožňuje užívateľovi zmenu hesla
MasterPage.master	definuje vzhľad úvodného užívateľského rozhrania
password.aspx	umožňuje zaslanie prístupových údajov na registračnú emailovú adresu
register.aspx	zabezpečuje registráciu nových užívateľov
search.aspx	umožňuje vyhľadávanie genealogických informácií vo verejných rodokmeňoch
Web.config	obsahuje konfiguračné nastavenia pre Membership a pripojenie k databáze
Web.sitemap	XML štruktúra celého informačného systému
db/genealogydb.mdf	obsahuje databázový súbor