

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

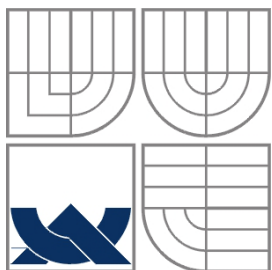
SPISOVÁ SLUŽBA ADVOKÁTNÍ KANCELÁŘE S
NAPOJENÍM NA VEŘEJNÉ DATABÁZE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

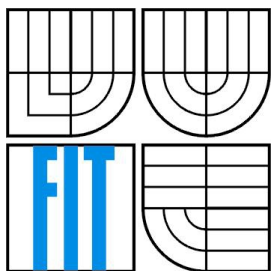
AUTOR PRÁCE
AUTHOR

BC. JIŘÍ JANDA

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SPISOVÁ SLUŽBA ADVOKÁTNÍ KANCELÁŘE S NAPOJENÍM NA VEŘEJNÉ DATABÁZE

FILE MANAGEMENT IN LEGAL COMPANY WITH CONNECTION TO PUBLIC DATABASES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

BC. JIŘÍ JANDA

VEDOUCÍ PRÁCE

SUPERVISOR

ING. PAVEL OČENÁŠEK, PH.D.

BRNO 2014

Abstrakt

Tato práce se zabývá popisem systému pro podporu administrativních procesů v advokátních kancelářích. Zaměřuje se na zjištění procesů v kancelářích, analyzuje možnosti zjednodušení práce a celkově její automatizace. Důraz je kladen zejména na možnosti automatizovaného získávání informací z veřejných databází. Dále práce popisuje a porovnává již existující řešení, která jsou běžně dostupná na trhu. V další části práce je řešen návrh samotného systému a volba vhodných technologií pro praktickou realizaci navrženého systému. Cílem práce je pak implementace systému dle vytvořeného návrhu a jeho otestování na reálných datech.

Abstract

This Master's thesis concerns the description of the systems used for the support of the administrative processes in legal offices. It focuses on finding the processes in offices, analyzes the possibilities for their simplification and automation. The great emphasis is placed especially on possibilities of automatic acquisition of information from public databases. Furthermore, the thesis describes and compares already existing solutions that are commonly available on the market. In another part of this thesis is being solved proposal of system itself and choice of suitable technologies for its practical implementation. The main goal of this paper is to implement the system according to generated proposal and its testing on real data.

Klíčová slova

Advokátní kancelář, spisy, získávání dat, web, ASP MVC, veřejné databáze.

Keywords

Legal company, files, data mining, web, ASP MVC, public databases.

Citace

Jiří Janda: Spisová služba advokátní kanceláře s napojením na veřejné databáze, diplomová práce, Brno, FIT VUT v Brně, 2014

Spisová služba advokátní kanceláře s napojením na veřejné databáze

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Pavla Očenáška, Ph.D.

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

.....
Jiří Janda
16. května 2014

Poděkování

Rád bych poděkoval mému vedoucímu Ing. Pavlu Očenáškoví, Ph.D. za pomoc a rady při vypracování této práce.

© Jiří Janda, 2014

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 Struktura práce.....	3
2 Procesy v advokátní kanceláři.....	4
2.1 Civilní a trestní proces.....	4
2.2 Vedení spisu.....	4
2.3 Vedení klientů / osob.....	5
3 Existující řešení.....	6
3.1 InfoSoud.....	6
3.2 Kardex PowerPick Office.....	6
3.3 Acta Safe.....	7
3.4 Advokátní spis.....	8
3.5 Jurisdix AK.....	8
3.6 SynopsIS.....	9
3.7 Kleos.....	9
3.8 Porovnání.....	10
4 Analýza požadavků.....	12
4.1 Správa spisů.....	12
4.2 Správa dokumentů.....	12
4.3 Správa pošty.....	12
4.4 Činnosti a fakturování.....	12
4.5 Statistiky.....	13
4.6 Získávání informací.....	13
4.7 Způsob realizace.....	13
4.8 Uživatelské role.....	13
4.8.1 Pracovník.....	14
4.8.2 Manažer.....	14
4.9 Diagram případů užití.....	14
5 Návrh systému.....	16
5.1 ER diagram.....	16
5.2 Popis tabulek databáze.....	17
5.3 Návrhové vzory.....	18
4.7.1 Repository.....	18
6 Použité technologie.....	19
6.1 Volba platformy.....	19
6.2 ASP.NET.....	19
6.3 MVC.....	20
6.4 Entity Framework.....	21
6.5 HTML5.....	22
6.6 Responzivní design.....	22
6.7 Licence a cena technologií.....	23
6.7.1 Vlastní provoz.....	24
6.7.2 Hosting.....	24
7 Napojení na veřejné databáze.....	25
7.1 InfoSoud.....	25
7.2 InfoJednání.....	25
7.3 ARES.....	25
7.4 Insolvenční rejstřík.....	26
7.5 Způsob získávání dat.....	26

8 Implementace systému.....	27
8.1 Vytvoření nového projektu.....	27
8.2 Struktura projektu.....	28
8.3 Adresní struktura.....	30
8.4 Modely.....	31
8.5 Práce s databází.....	32
8.6 Rozložení stránky.....	33
8.7 Responzivní zobrazení.....	34
8.8 Autentizace a autorizace uživatelů.....	34
8.9 Správa zaměstnanců.....	35
8.10 Správa spisů.....	35
8.11 Získávání dat.....	36
8.11.1 Systém ARES.....	36
8.11.2 Systémy eJustice.....	36
8.12 Generování faktur.....	37
8.13 Oznamování nových událostí.....	39
9 Testování.....	41
9.1 Testování zobrazení.....	41
9.2 Testování funkcionality.....	41
9.3 Testování výpočtů a získávání dat.....	42
10 Závěr.....	43
10.1 Dosažené výsledky.....	43
10.2 Možnosti rozšíření.....	43
Literatura.....	45
Seznam příloh.....	46
Příloha A.....	47
Příloha B.....	48
Příloha C.....	51

1 Úvod

Přestože informační technologie hrají v současnosti významnou roli v každém oboru lidského podnikání, stále se velmi často stává, že nejsou využívány dostatečně a zejména efektivně. V běžném provozu advokátní kanceláře je nutné evidovat velké množství informací. Právě evidence informací a co nejsnadnější přístup k těmto datům je problém, který velmi efektivně řeší informační systémy. V advokátních kancelářích se administrativa stále často vede v papírové podobě, a to i přestože je na našem trhu k dostání velké množství různorodých komerčních systémů, které tuto práci automatizují, zpřehledňují a zjednodušují.

Úkolem v první části této diplomové práce je shromáždit požadavky a navrhnout systém určený zejména pro advokátní kanceláře, který by zautomatizoval určité každodenní procesy zaměstnanců a celkově zjednodušil nutnou administrativu, se kterou se každá advokátní kancelář potýká. Práce se dále věnuje analýze již existujících řešení od komerčních firem v České republice.

Druhou, praktickou částí této práce je výběr vhodných technologií a implementace navrženého systému v těchto technologiích. V závěru by měl být tento vytvořený systém otestován a vyzkoušen na reálných datech.

1.1 Struktura práce

Kapitola 2 této práce řeší administrativní procesy, které fungují v advokátních kancelářích. Tyto procesy začínají již tím, když nový klient vstoupí do kanceláře. Končí tehdy, když je klientův případ (i neúspěšně) vyřešen a je vystavena a zaplacená faktura. Tyto procesy je nutné dobře zanalyzovat, aby na jejich základě poté mohli vzniknout ucelené požadavky pro systém.

V kapitole 3 jsou popsány existující komerční softwarová řešení na českém trhu, které lze využít pro administrativu advokátní kanceláře. V této kapitole jsou detailněji popsány funkce, které jednotlivá řešení nabízí a v závěru je uvedeno tabulkové porovnání všech zkoumaných aplikací.

Kapitola 4 se zabývá analýzou požadavků na navrhovaný systém. Tyto požadavky vychází z kapitoly 2, ve které jsou řešeny procesy advokátní kanceláře. Na základě prozkoumání těchto procesů jsou dále vytvořeny požadavky na funkce, které by měla aplikace nabízet. Tyto funkce jsou vyobrazeny pomocí diagramu případů užití.

Kapitola 5 řeší návrh systému. Je zde detailněji rozebírána struktura ukládaných dat a vhodně prezentována pomocí ER diagramu.

Kapitola 6, která nese název Použité technologie, rozebírá jaké technologie budou použity pro tvorbu aplikace a důvody jejich výběru. V závěru kapitoly jsou rozebrány licence a náklady na jednotlivé technologie.

Kapitola 7 popisuje spolupráci aplikace a již existujících systémů veřejných databází. Zejména pak systémy eJustice - InfoSoud a InfoJednání, které provozuje Ministerstvo spravedlnosti.

V kapitole 8 je popsána samotná implementace aplikace. Detailněji je popsán konkrétní postup a řešení složitějších částí aplikace.

Kapitola 9 se zabývá testováním výsledné aplikace pomocí reálných dat. Testování probíhá jak uživatelským způsobem, tak automatizovanými testy.

Kapitola 10 obsahuje závěr této práce. Práce je zde zhodnocena, je porovnána s dalšími existujícími komerčními řešeními. V závěru kapitoly jsou také uvedeny možnosti případného budoucího rozšíření funkcionality.

Kapitoly 1 – 6 byly převzaty ze semestrálního projektu, který předcházel této diplomové práci.

2 Procesy v advokátní kanceláři

Hlavní proces každého případu začíná příchodem nového klienta do advokátní kanceláře. Klient přednese svůj problém a pokud se s některým pracovníkem advokátní kanceláře dohodne na spolupráci, je mu založen nový spis.

2.1 Civilní a trestní proces

Tento nový klient může přijít do advokátní kanceláře ze dvou hlavních důvodů. V prvním případě přichází klient preventivně, kvůli případným právním sporům, které očekává v budoucnu, nebo již nějaký právní spor existuje a klient potřebuje pomoci s jeho řešením. Tato varianta se nazývá **civilní proces**. Po domluvě s advokátní kanceláří je založen nový spis. Prvním, přiloženým dokumentem do spisu, je plná moc, na jejímž základě může advokát zastupovat svého klienta u soudu. V průběhu civilního procesu do spisu mohou přibývat další dokumenty (například znalecké posudky, vyjádření či důkazní materiál). Když je proces ukončen a v případě, že se protistrana již dále neodvolává je ukončen i vedený spis. Posledním vkládaným dokumentem je doklad o zaplacení advokátních služeb klientem.

Druhým případem může být tzv. **trestní proces**. V takovém případě se klientem stává osoba, která je podezřelá ze spáchání trestného činu. Klient si může vybrat advokáta dobrovolně, nebo je mu přidělen z úřední povinnosti. V trestním procesu je opět založen spis, do kterého je vložena plná moc. Dále může být vloženo například usnesení o zahájení trestního stíhání či obžaloba, to zaleží na tom, v jakém stupni se již proces nachází.

2.2 Vedení spisu

Pokud advokátní kancelář nevyužívá žádný software pro ulehčení spisové administrativy, je tento spis v podstatě fyzická složka, která v sobě zahrnuje veškerou papírovou dokumentaci k případu. Obsahuje tedy například uzavřenou smlouvu mezi právníkem a jeho klientem, sepsanou plnou moc pro právního zástupce, veškerou dokumentaci k dané kauze či založenou fyzickou korespondenci vazánou k tomuto případu. Do spisu jsou průběžně přikládány nové dokumenty (či pouze nové verze již existujících dokumentů), až dokud spis není uzavřen.

Pokud je daný případ řešen soudem, je mu přidělena určitá soudní spisová značka. Tu obdrží každé soudní řízení (zahájené například žalobou). Tato spisová značka je unikátní identifikace pro každý spis vedený u soudu v České republice. Tato spisová značka je poté uváděna na všech dokumentech, které soud v souvislosti s daným případem vydá. U nás je tento způsob identifikace užíván již od konce 19. století.

Pro tvar spisové značky platí pevně formulovaná pravidla. Každá značka se skládá ze čtyř částí:

- číslo soudního senátu
- druh věci podle soudní agendy
- běžné číslo
- lomítko a rok vzniku

Spisová značka může tedy vypadat například takto: 54 D 2222 / 2013. Pokud se jedná o insolvenční řízení, je před číslo ještě přidána zkratka krajského soudu, kde je daný případ veden.

Ne každý spis je řešen soudem a nemusí tedy mít nutně přidělenou spisovou značku. Je proto nutné, aby advokátní kancelář každý spis již od založení evidovala pomocí své interní spisové značky, která bude také unikátní pro každý spis vedený v rámci kanceláře.

Tento spis může být obecně ve dvou stavech. Otevřený, což znamená, že věc stále trvá a není vyřešena. Po ukončení daného případu se spis uzavírá (stav Uzavřený).

2.3 Vedení klientů / osob

Advokátní kancelář musí v rámci spisu (nebo i zvlášt mimo spis) evidovat údaje o různých osobách. Nejdůležitější osoby jsou samozřejmě klienti dané kanceláře. Nejsou to ale jediné osoby, které je nutné evidovat, dále může jít například o protistrany v případech klientů. K jednomu klientovi může náležet i více než jeden spis (nebo teoreticky nemusí žádný). Vedení těchto informací v rámci každého spisu vede ke zbytečnému a redundantnímu ukládání údajů o klientech v případě, že k jednomu klientovi náleží více než jeden spis. To je problém, který se v papírově vedené evidenci řeší těžko, ale pro počítačovou aplikaci je takové řešení velmi snadné. Údaje o klientovi se uloží pouze jedinkrát a ke každému spisu se pouze uloží vazba k danému klientovi, čímž se minimalizuje velikost ukládaných informací.

3 Existující řešení

Na českém trhu existuje již několik komerčních řešení na ulehčení administrativy pro advokátní kanceláře. Některá tato řešení nejsou určena přímo pro advokátní kanceláře, ale lze je využít i pro tento účel. Jednotlivá řešení se liší jak svojí celou koncepcí (webový informační systém, desktopová aplikace) tak i zaměřením svých doplňkových funkcí (například důraz na práci v týmu či rozšířené možnosti finanční evidence).

V jednotlivých částech této kapitoly je uveden přehled některých z existujících řešení. V závěrečné podkapitole je potom uvedena tabulka, která slouží pro porovnání funkcionality, kterou jednotlivá řešení nabízí. Údaje pro jednotlivé popisy systémů jsem získal z informací o produktech uvedených na stránkách jejich výrobců.

3.1 InfoSoud

Aplikace hlídá změny zadaných spisů, které se objeví na oficiálních webových stránkách Justice – InfoSoud. Aplikace přes internet stahuje data ze serveru justice.cz, které dále zpracovává. Na všechny nové informace upozorňuje svého uživatele. Aplikaci lze mít také spuštěnou jako službu běžící na pozadí počítače, která uživatele informuje o změnách pomocí zasílání e-mailových zpráv.

Mezi další nabízené funkce patří například hromadné zpracování záznamů, snadné zobrazení dalších informací (obchodní rejstřík, insolvenční rejstřík) klienta či odpůrce, hromadné zpracovávání záznamů, automatické zasílání emailu při změně či možnost ukládání poznámek ke spisům. Tvůrcem aplikace InfoSoud je Ing. Petr Holubec [1].

3.2 Kardex PowerPick Office

Software Kardex PowerPick Office slouží pro správu archivovaných spisů. Ne však ve smyslu právních spisů, ale obecně libovolných dokumentů. Nejde tedy o specializovaný software pro advokátní kanceláře, ale pro jakýkoliv obor, kde je potřeba efektivně spravovat větší množství uložených dokumentů. Aplikace umožňuje rychlý přístup k uloženým dokumentům v prostředí Microsoft Windows a dle popisu výrobce je možné ji jednoduše integrovat do stávající IT-infrastruktury. Výrobce slibuje rychlé získání rutiny a vysokou produktivitu uživatelů tohoto softwaru.

Mezi další nabízené funkce patří rychlé vyhledávání pomocí volitelných kritérií, evidence o pohybech spisů, redukce chyb při ukládání spisů, redukce doby archivace či přehled doby archivace. Tvůrcem této aplikace je firma Kardex s.r.o. [2].

3.3 Acta Safe

Acta Safe je aplikace, která má dle popisu firmy spojovat principy projektového managementu spolu s nástroji pro správu dokumentů. umožňuje vedení jednotlivých spisů (obecně zakázek) a dokumentů. Ke každému dokumentu i spisu lze nastavit hlídání termínů s napojením na aplikaci MS Outlook (e-mailový klient). V návaznosti na evidované spisy umožňuje aplikace Acta Safe vystavit faktury se sazbou zvlášť pro každého klienta. Každému klientovi lze také nastavit limit záloh, čímž se omezuje práce advokátů bez jistoty jejího finančního pokrytí. Aplikace je napojena na datové schránky a e-mailový klient, čímž zjednodušuje veškerou elektronickou komunikaci.

Další funkce, které tato aplikace nabízí, jsou plánování práce pro týmy, hodnocení jednotlivých členů pracovních týmů či částečné uzpůsobení systému. Tvůrcem aplikace Acta Safe je firma AiP Safe s.r.o. [3]

The screenshot displays the Acta Safe web application interface. At the top, there is a navigation bar with 'Hlavní stránka', 'Fakturace', and 'Můj výběr'. The user is identified as 'Petr Melčík (partner)'. The main content area shows details for a case 'PM-C-0088/2011'. The case information includes: Rejstřík: C, Věc: Založení společnosti Tremor Group, Jazyk: Čeština, Stav: aktivní, Gen: ne. The client is 'Tremor Group'. Other fields include 'Protistrana', 'Zástupce protistrany', 'Rozpočet: 23,5', 'Odpovědnost: Petr Melčík (partner)', 'Tým spisu: Klára Faiblová, Rytych David (advokát)', 'Uložení: archiv D, složka 12', 'Zapůjčen: ne', and 'Datum vzniku'. There is also a 'Literatura' section with 'Klíčová slova: vznik právnické osoby' and 'Související spisy:'. Below this, there are sections for 'Termíny, lhůty, události (0)', 'Dokumenty (3)', and 'Požadavky (4)'. The 'Požadavky' table is as follows:

Od	Do	Zpracovatel	Stav	Požadavek	Odpracováno
10.8.2010		Klára Faiblová	Splněný	příprava podkladů	3
2.8.2010		Klára Faiblová	Splněný	zajištění úředně ověřeného překladu	1
28.7.2010		Klára Faiblová	Splněný	revize smlouvy, vyjádření k návrhu smlouvy	1
16.7.2010		Rytych David (advokát)	Splněný	osobní jednání s klientem	2

The total 'Odpracováno' is 7,00. At the bottom, there is a section for 'Činnosti (11)' with columns for 'Datum', 'Vytvořil', 'Stav', 'Činnost', 'Skutečná pracnost', and 'K fakturaci'.

Obrázek č. 3.1 – Detail spisu v programu Acta Safe. Zdroj [3]

3.4 Advokátní spis

Tento produkt jako většina ostatních umožňuje vedení spisů, údajů o klientech i protistranách. Ke spisům lze vkládat lhůty, které jsou propojené s programem MS Outlook. Aplikace je dále také propojena s insolvenčním rejstříkem. Výhodou tohoto produktu je zaměření na jednotlivé zaměstnance advokátní kanceláře. Každému pracovníkovi lze vytvořit vlastní přístupový účet, kterému lze nastavit individuální přístupová práva ke spisům a operacím.

Další důležitou částí této aplikace jsou ekonomické funkce. Lze evidovat zálohy zaplacené klienty. Účtování eviduje jednotlivé sazby za právní úkony (lze rozlišovat hodinové sazby, tarifní odměny) nebo náklady (soudní poplatky, cestovní náhrady).

Software nabízí další funkce jako například export účtování do MS Excel, nastavitelnost pro plátce a neplátce DPH či evidence odeslané a přijaté pošty. Tvůrcem aplikace je firma ATLAS consulting spol. s.r.o. [4].

Odměny a náklady		Zálohy		Faktury	
Položka	Částka	Položka	Částka	Položka	Částka
Hodinová sazba	12 000 Kč	Zálohy k dispozici	5 500 Kč	Položky bez faktury	350 Kč
Paušální odměna	0 Kč	Vyfakturované položky	4 500 Kč	Neproplacené faktury	0 Kč
Tarifní odměna	0 Kč	Nezaplacené položky	0 Kč	Zaplacené faktury	4 500 Kč
% z úspěchu	0 Kč	Celkem	10 000 Kč	Celkem	4 850 Kč
Skutečné náklady	25 850 Kč	Celkový součet vyúčtování práce advokátní kanceláře			
Soudní poplatky	2 200 Kč	<input checked="" type="checkbox"/> Zálohy	Všechny zálohy byly uhrazeny.		
Cestovní náhrady	0 Kč	<input checked="" type="checkbox"/> Náklady	Některé náklady dosud nebyly vyfakturovány. Celkem za 35 590 Kč.		
Promeškaný čas	0 Kč	<input checked="" type="checkbox"/> Faktury	Všechny faktury byly zaplacené.		
Paušální výdaje	0 Kč	<input checked="" type="checkbox"/> Přeplatky	Nějsou předpokládány žádné přeplatky.		
Tarifní paušál	0 Kč	<input type="checkbox"/> Finančně vyrovnáno <input type="checkbox"/> Spis byl finančně uzavřen dne 17. 3. 2011			
Celkem	40 050 Kč	<input type="button" value="Tisk"/> <input type="button" value="Export..."/>			

Obrázek č. 3.2 – Detail spisu v programu Advokátní spis. Zdroj [4]

3.5 Jurisdix AK

Jurisdix AK je specializovaný software pro vedení advokátní kanceláře. Je určen pouze pro advokáty registrované u České Advokátní Komory. Aplikace umožňuje evidovat dokumentaci ke spisům, klienty a veškeré informace týkající se daných kauz. Je možné využít plánování úloh, evidovat přijatou a odeslanou poštu.

Podporována je i práce v týmu, kdy každý uživatel může mít svůj unikátní přístup a své heslo. Administrátor poté může každému uživateli nastavit různá oprávnění. Pomocí programu lze evidovat i účetnictví, vystavovat faktury a zaznamenávat práci na uložených případech. Celá aplikace je

koncipována jako modulární (např. kauzy, administrativa, dlužníci klientů, adresáře, účetnictví, kniha jízd a další)

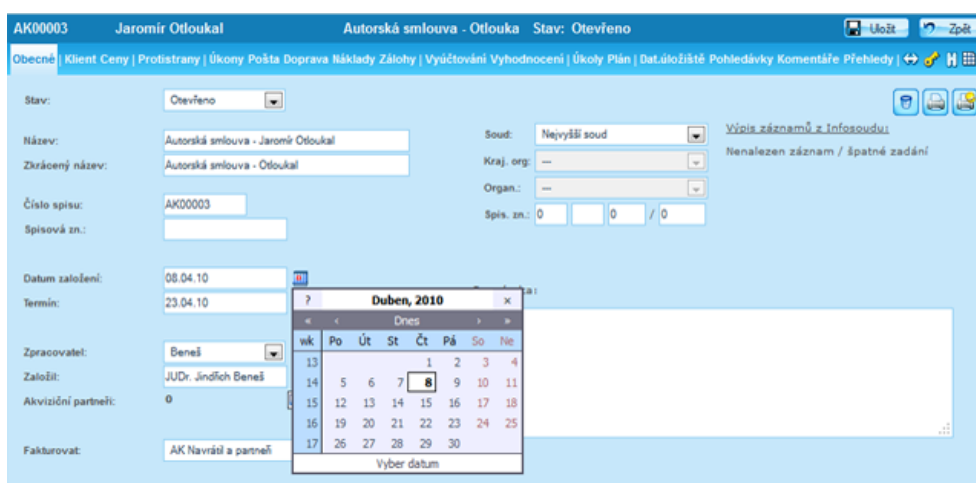
Autorem tohoto softwaru je Dalibor Smitka [5].

3.6 SynopsIS

SynopsIS je modulární informační systém, který není přímo určen pro advokátní kanceláře, ale obecně pro firmy s nutností vedení rozsáhlejší agendy (například účetní společnosti, správu pohledávek a další). Jeden z modulů je specializován pro advokátní kanceláře.

Tento modul vylepšuje práci s velkým množstvím dokumentů a případů, které každá střední a větší advokátní kancelář má. Umožňuje několik druhů oprávnění v závislosti na pozici pracovníka v dané firmě (koncipient, advokát, manažer, partner). Celý systém je napojen na insolvenční rejstřík, dále systém ARES (Administrativní Registr Ekonomických Subjektů), datové schránky a také systém Infosoud. Díky těmto napojením lze snadno získat velké množství souvisejících informací.

Výrobce aplikace je společnost KOMPL s.r.o. [6].

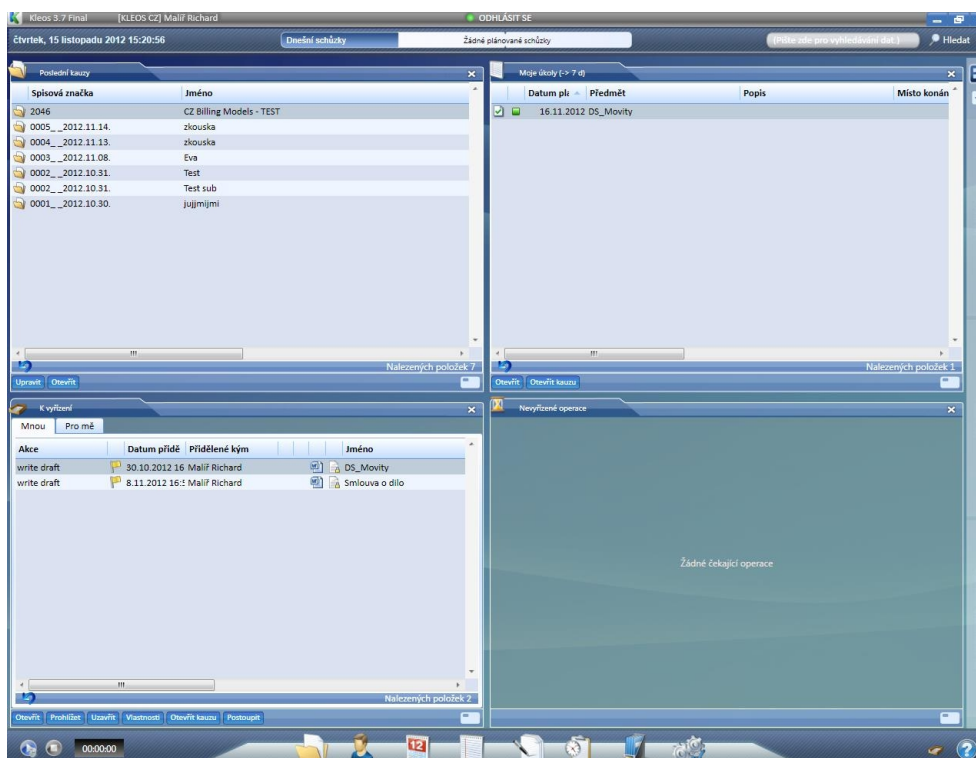


Obrázek č. 3.3 – Detail spisu v programu SynopsIS. Zdroj [6]

3.7 Kleos

Hlavní výhodou systému Kleos má být společné prostředí pro práci, které spojuje všechny zaměstnance advokátní kanceláře. Systém usnadňuje komunikaci s týmu, čímž by měl zvyšovat rychlost provádění týmových úkolů a celkově produktivitu zaměstnanců.

Systém se skládá ze 7 modulů: kauzy, aktivity, kontakty, úkony, kalendář, dokumenty a finance. Kleos umožňuje evidenci spisů, ke kterým lze přikládat dokumenty, komunikaci, jednotlivé úkony nebo třeba domluvené schůzky. Libovolné úkony, schůzky či jiné události lze plánovat dopředu. Systémový plánovač pak uživatele vždy včas informuje. Pomocí systému může advokátní kancelář svým klientům vystavit faktury za své služby. Funkcionalita tohoto produktu je dle popisu výrobce opravdu obsáhlá. Jako další funkce lze uvést například: desktopová aplikace „Mé úkony“, import dokumentů, PDF konverze, vytváření e-mailů z uložených šablon či automatické vytváření úkolů z kalendáře. Veškerá data jsou ukládána v datovém centru výrobce a jejich přenos je realizován pomocí šifrovaného protokolu HTTPS. Výrobce systému Kleos je Wolters Kluwer ČR, a. s. [7].



Obrzek . 3.4 – Pracovní plocha programu Kleos. Zdroj [7]

3.8 Porovnn

V tabulce 3.1 – Porovnn funkc existujcch řešen, je zobrazen přehled vech vye uvedench řešen. U každho řešen je uvedeno, zda-li nabízí tyto funkcionality: sprva spis, napojen na systmy eJustice, napojen na Insolvenn rejstřk, napojen na datov schrnky, sprva poty (elektronick, nebo paprov), sprva uetnictv (evidence kon, vystaven faktur), podpora pro prci v tmu. Z přehledu mžeme vidt, že nejmn ze stanovanch funkc nabízí aplikace Kardex P. O. (1 ze 7), nejvce funkc potom nabízí systmy SynopsIS a Advoktn spis (5 ze 7).

Funkce	InfoSoud	Kardex P. O.	Acta Safe	Advokátní spis	Jurisdix AK	SynopsIS	Kleos
Správa spisů	Ne	Ano	Ano	Ano	Ano	Ano	Ano
Napojení na eJustici	Ano	Ne	Ne	Ne	Ne	Ano	Ne
Napojení na IR	Ano	Ne	Ne	Ano	Ne	Ano	Ne
Napojení na DS	Ne	Ne	Ano	Ano	Ne	Ano	Ne
Správa pošty	Ne	Ne	Ano	Ano	Ano	Ne	Ano
Správa účetnictví	Ne	Ne	Ano	Ano	Ano	Ano	Ano
Podpora týmu	Ne	Ne	Ne	Ne	Ne	Ne	Ano

Tabulka č. 3.1 – Porovnání funkcí existujících řešení

4 Analýza požadavků

V kapitole 2 jsou popsány procesy, vyskytující se v běžné praxi advokátní kanceláře. Na základě těchto procesů je nutné vytvořit požadavky, které musí vytvořený systém splňovat, aby tyto procesy ulehčoval. Jako jednu z prvních věcí je nutné určit uživatelské role, které se budou v systému vyskytovat. Dále také akce, které budou těmto uživatelským rolím umožněny. Přehled rolí a jejich možných akcí v rámci systému je na konci kapitoly zobrazen pomocí diagramu případů užití.

4.1 Správa spisů

Hlavním cílem celého systému je uchování informací o vedených spisech. Každý spis by měl být propojen se zvláště evidovaným klientem. Díky tomu se zamezí nutnosti opakovaně zadávat informace o klientovi u každého spisu, který se k němu váže. Každý spis se může nacházet v jednom ze dvou stavů: otevřený, nebo uzavřený. Ke spisu by také mělo být možné přiložit informaci o protistraně. O té jsou evidovány stejné informace jako o klientovi.

System by měl nabízet možnost evidovat ke každému spisu jednotlivé úkony, které advokátní kancelář v rámci práce provedla.

4.2 Správa dokumentů

Ke každému uloženému spisu by měla být možnost uložit neomezené množství dokumentů. Slovo dokument v tomto případě nemusí přesně znamenat pouze dokument ve smyslu formátovaného textového souboru. Může to být libovolný soubor, tedy například obrázek (naskenovaný dokument), nebo celý archiv jiných souborů. V systému by měla být možnost verzování dokumentů. To znamená, že u každého dokumentu by mělo být možné vytvořit několik různých verzí tohoto dokumentu. To je vhodné zejména u dokumentů, které se v průběhu času mění, ale je stále nutné evidovat i jejich původní znění.

4.3 Správa pošty

Podobně jako je možnost vést databázi dokumentů, měl by systém poskytovat i možnost vést si evidenci pošty vztahující se k danému spisu. Pošta zde není chápána pouze ve smyslu elektronické pošty (e-mail), ale i jako papírová pošta. Uchovávána by měla být buď jako naskenovaný dokument (obrázek), nebo jako dokument (naskenovaný papírový dokument převedený na elektronický dokument pomocí OCR), obecně tedy jako soubor, ne pouze jako prostý text. U této funkce by měla být možnost rozlišovat poštu přijatou a odeslanou. Dále by mělo být dostupné i další členění (elektronická pošta, datové schránky, papírová atd.).

4.4 Činnosti a fakturování

System by měl zjednodušit vedení činností a v návaznosti na ně i fakturování všech aktivit vykonaných na jednotlivých případech. Vedoucí pracovníci by měli mít možnost vytvořit seznam možných činností, které mají kromě názvu uloženou také cenu za hodinu práce. Při práci na jednotlivých případech mohou poté pracovníci advokátní kanceláře využít položky z tohoto seznamu při zadávání činností, které prováděli na případu. Záznam o takové činnosti by měl vést také informace o její délce. Kromě samotných činností by mělo být umožněno zaznamenat k případu i další jednorázové náklady (které netrvají žádný čas), jako například kolky či soudní poplatky.

Na základě uložených činností i jednorázových nákladů by vedoucí pracovník měl mít možnost vystavit klientovi fakturu. K jednomu spisu mohou být vystaveny faktury opakovaně

(s novými aktivitami a náklady). Takto vystavené faktury by mělo být možné exportovat do PDF, aby je bylo možné zaslat klientovi.

4.5 Statistiky

System by měl nabízet základní statistický přehled hlavních položek. Tedy počty klientů, spisů, dokumentů atd. Statistiky by též měli pokrývat ekonomickou část. Ta by měla obsahovat přehledy finanční, které jsou získané z uložených činností a vystavených faktur. Protože tento statistický přehled může obsahovat citlivá finanční data, neměl by být přístupný všem zaměstnancům, ale pouze vedoucím pracovníkům.

4.6 Získávání informací

Velmi častou aktivitou v advokátních kancelářích je získávání informací ze státních informačních systémů. Nejčastěji se jedná o data o fyzických / právnických osobách a také informace o spisech, což je například přehled jednání. Právě tyto informace můžeme získat ze systémů eJustice, systému ARES či Insolvenčního rejstříku. Vytvářený systém by měl umožňovat automatické získávání informací z těchto zdrojů, nebo k nim alespoň usnadňovat přístup. Automaticky získávané informace v pozadí, by měly být v reálném čase zobrazovány uživateli pomocí malého informačního boxu. Více o konkrétních možnostech získávání informací z těchto systémů je uvedeno v kapitole 7 [8] [9] [10] [11].

4.7 Způsob realizace

Jedním z požadavků na aplikaci je možnost práce více uživatelů v systému současně. Zároveň by také bylo vhodné, aby aplikace a její data byla přístupná pro zaměstnance nejen v rámci advokátní kanceláře, ale i například doma, nebo na služební cestě. Je tedy nutné, aby aplikace měla data uložená centrálně, nejlépe přístupná přes internet, díky tomu bude možné získat přístup k datům odkudkoliv.

Abychom vyhověli výše uvedenému požadavku, jsou u samotné aplikace dvě možnosti řešení. První variantou je desktopová aplikace, která pracuje s daty uloženým na vzdáleném serveru. Tyto data stahuje a nahrává přes internet pomocí stanoveného protokolu. Jako druhá možnost se nabízí webová aplikace. Tato aplikace běží na hostitelském serveru, který je připojený na internet. Uživatel s takovou aplikací pracuje vzdáleně pouze pomocí webového prohlížeče.

Hlavní nevýhodou desktopové aplikace je nutnost mít ji nainstalovanou na všech počítačích, ze kterých bude využívána. To nemusí být vždy možné (například na pracovní cestě, u klienta). Dále také není možné ji využít na jiných platformách, než pro které byla primárně vytvořena (např. mobilní zařízení). Oba tyto jmenované problémy u webové aplikace nenastávají. K webové aplikaci je možné přistoupit z jakékoliv platformy a přístroje, který je připojen k internetu s dostupným webovým prohlížečem. Z těchto důvodů se webová aplikace jeví jako nejvhodnější možnost. Protože výrazným trendem současnosti je přístup na internet pomocí chytrých mobilních zařízení, je nutností správné zobrazení na těchto menších displejích. Výstup aplikace by tedy měl být proveden v responzivním designu.

4.8 Uživatelské role

Protože v reálné advokátní kanceláři pracuje více pracovníků na různých pozicích, kdy každá tato pozice má v rámci kanceláře jinou pravomoc, je třeba tuto skutečnost v aplikaci respektovat a umožnit více uživatelských rolí s různými stupni oprávnění. Pro potřeby menších a středních advokátních kanceláří jsem zvolil dvě základní uživatelské role: Pracovník a Manažer. Většina oprávnění pro tyto dvě role je společná. Role Manažer má ale navíc i další pravomoce, tak jak je to i v reálném prostředí.

Pro obě uživatelské role je nutné přihlášení do systému pomocí unikátního přihlašovacího jména a hesla. Systém samotný žádné veřejné rozhraní bez přihlášení nepotřebuje, proto jediná veřejná část systému je právě přihlašovací formulář. Prakticky tedy existuje ještě role Nepřihlášený uživatel, ale jeho jedinou aktivitou je možnost přihlásit se. Tato role není dále v této práci uváděna.

4.8.1 Pracovník

Uživatelé s pravomocemi Pracovník mohou vykonávat všechny aktivity spojené se správou klientů a protistran. Těmto uživatelům může uživatel s oprávněním Manažer přiřadit spisy (jeden spis i více Pracovníkům, nebo žádnému), ke kterým poté mají přístup. Mohou provádět úpravy se všemi položkami, které jsou vázány k danému spisu (dokumenty, pošta). Spisy, které Pracovník vytvoří jsou mu automaticky přiřazeny. Pracovník může do systému vkládat informace o své odvedené práci na jednotlivých případech či dalších nákladech, které bylo nutné vynaložit v rámci případu.

4.8.2 Manažer

Uživatel v roli Manažer může vykonávat všechny akce jako uživatel s rolí Pracovník. Narozdíl od Pracovníka má přístup ke všem spisům uloženým v systému i s náležejícími položkami. Administrátor navíc může mazat či přidávat nové zaměstnance v systému a nastavovat jejich oprávnění. Jednotlivým zaměstnancům může přiřazovat a odebírat existující spisy. Systém ukládá veškeré důležité akce provedené všemi uživateli, jejichž přehled je také dostupný pro tuto roli.

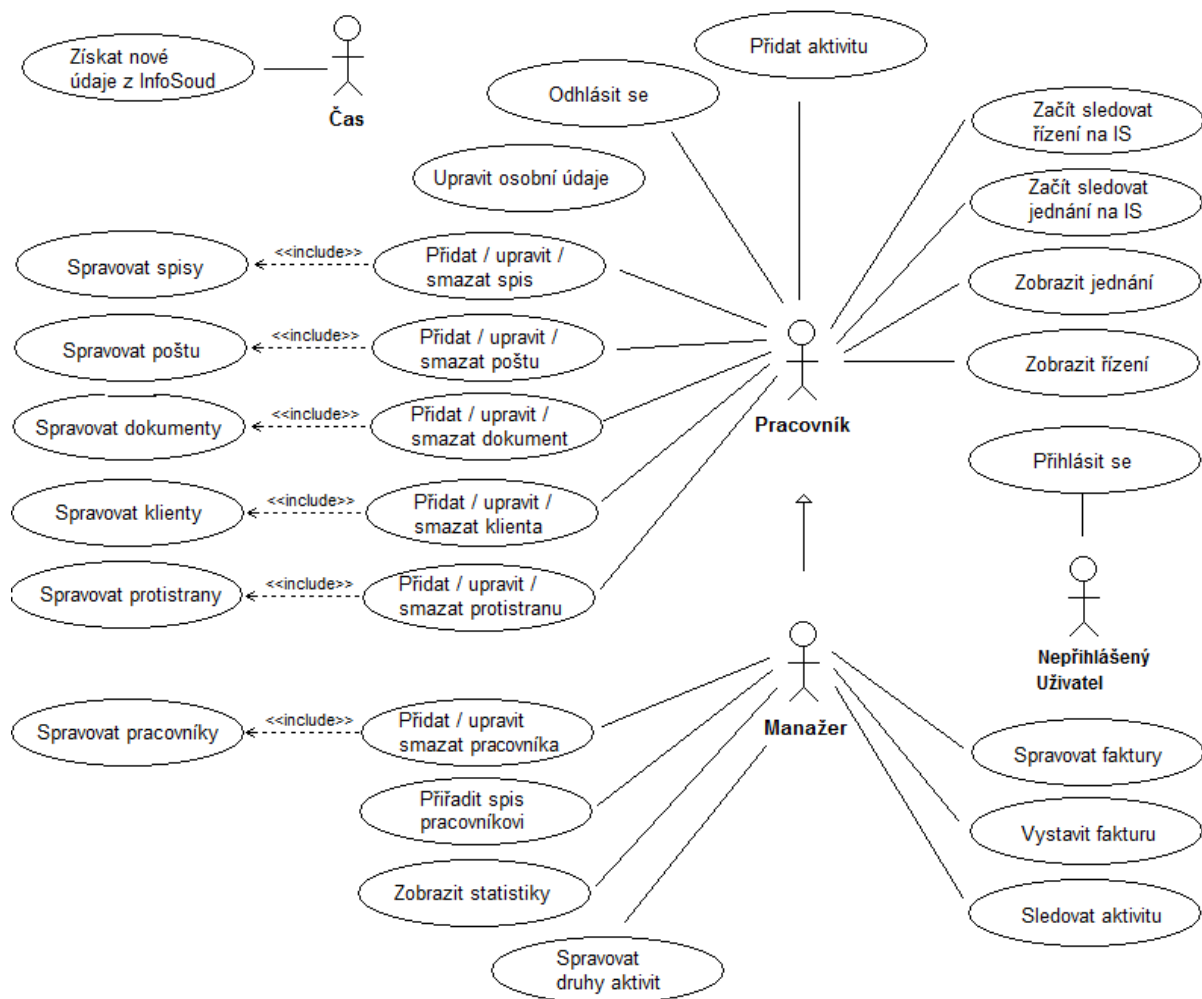
Manažer má dále také statistický přehled o fungování celé kanceláře (počty případů, klientů, fakturované částky, ...). Manažer může vystavovat jednotlivým klientům advokátní kanceláře faktury a po ukončení případu může celý spis uzavřít.

4.9 Diagram případů užití

Diagram případů užití slouží v UML pro modelování chování. Tento diagram zobrazuje vnější pohled na systém, který modeluje. Určuje tím hranice systému a také jeho hlavní aktéry. Takovým aktérem může být uživatel (v různých oprávněních), nebo třeba jiný systém. Ve speciálních případech může být aktérem například čas.

Hlavním účelem diagramu případů užití je zachycení aktérů a jejich komunikace se službami, které jim systém nabízí. Pro zobrazení se většinou využívá grafická podoba, případně je možné využít textový popis (pro detaily konkrétních případů užití). Forma tohoto diagramu je jednoduchá a přehledná, aby byla srozumitelná nejen vývojářům systému, ale i zákazníkům, se kterými je obvykle tvorba konzultována [17].

Přehled všech hlavních aktivit, které mohou obě uživatelské role v systému vykonávat, je zobrazen na obrázku č. 4.1 – Diagram případů užití



Obrázek č. 4.1 – Diagram případů užití

5 Návrh systému

5.1 ER diagram

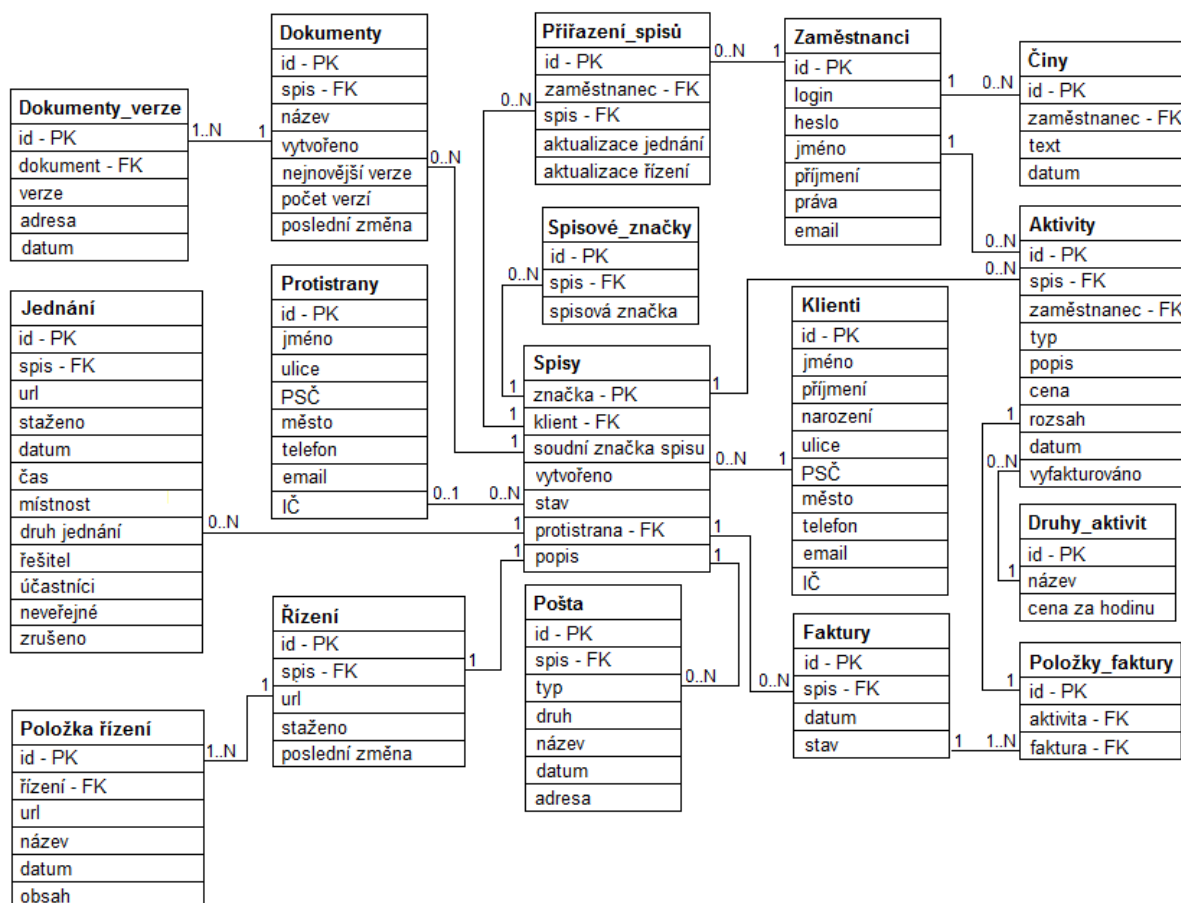
ER (Entity Relationship) Diagram je prostředek pro abstraktní a konceptuální znázornění ukládaných dat v modelovaném systému. Znázorňuje nejen uložená data, ale i jednotlivé vazby mezi nimi.

Důležitým pojmem v ER diagramu je entitní množina (v rámci relační databáze lze chápat jako tabulku). Tato entitní množina se skládá z jednotlivých atributů (v databázi reprezentováno jako sloupec tabulky). Každá položka této entitní množiny by měla mít jeden atribut jako svůj jednoznačný identifikátor v rámci množiny (primární klíč, který může být jednoduchý, nebo složený z více atributů).

Jednotlivé entitní množiny jsou spojeny vztahy různých typů. Tyto typy se dělí dle toho, s kolika prvky druhé entitní množiny může jedna položka první entitní množiny být svázána. Hodnoty mohou být například 1:N (jedna s mnoho), nebo N:M (mnoho s mnoho).

Přestože výsledná databáze, kterou aplikace využívá, nemusí nutně odpovídat ER diagramu (ve kterém se například nemusí uvádět spojovací tabulky), jde o hlavní prostředek využívaný pro modelování databáze [16].

ER diagram pro řešenou aplikaci je zobrazen na obrázku č. 5.1 – ER diagram



Obrázek č. 5.1 – ER diagram

5.2 Popis tabulek databáze

Spisy – jde o základní a nejdůležitější tabulku celého systému. Obsahuje hlavní údaje o uložených spisech. Jako primární klíč slouží interní značka spisu. Formát značky spisu jsem zvolil rok-pořadové_číslo_spisu, tedy například: 2014-2 pro druhý spis v kalendářním roce 2014. Jsou zde uloženy vazby na odpovídající řádky v tabulkách klientů a protistran. Tabulka dále obsahuje datum založení spisu, příznak zda-li je spis otevřený či uzavřený a jednoduchý popis.

Zaměstnanci – v této tabulce jsou uchovávána data o zaměstnancích advokátní kanceláře. Jsou zde uloženy přihlašovací údaje – unikátní přihlašovací jméno a zašifrované heslo. Dalšími údaji jsou e-mailová adresa, celé jméno zaměstnance a také jeho oprávnění.

Klienti – v této tabulce jsou uloženy údaje o klientech advokátní kanceláře. U klienta evidujeme jeho jméno, datum narození (a tedy věk), jeho poštovní adresu a kontaktní údaje (telefonní číslo a e-mail)

Protistrany – v této tabulce jsou zaznamenány údaje o případných protistranách klientů advokátní kanceláře. Kromě názvu protistrany se stejně jako u klienta ukládá její poštovní adresa a kontaktní údaje.

Přiřazení spisů – tato tabulka obsahuje data o přiřazení jednotlivých zaměstnanců ke spisům. Kromě samotného přiřazení jsou zde uložena data o tom, kdy naposledy si zaměstnanec prohlédl informace o řízeních a jednáních náležících k danému spisu. Tyto informace slouží k pozdějšímu označení nových položek, které byly staženy ze serveru eJustice.

Spisové značky – v této tabulce jsou uloženy jednotlivé soudní spisové značky (složené ze 4 částí), které byly přiřazeny vedeným spisům. Dále tabulka obsahuje informace o soudech, které spis projednávají. Jeden spis může mít navázaných více soudních spisových značek.

Dokumenty – zde jsou uloženy jednotlivé názvy a data vytvoření dokumentů přiložených ke spisům. Z důvodu možnosti uložení více verzí stejného dokumentu zde nemohou být přímo uloženy odkazy na samotné dokumenty. Je zde také uložena vazba, ke kterému spisu daný dokument náleží.

Dokumenty_verze - tato tabulka již obsahuje samotné odkazy na uložené dokumenty, společně s označením, o kterou verzi daného dokumentu se jedná, a také vazbu na tento dokument.

Pošta – tato tabulka eviduje poštu přikládanou k jednotlivým spisům. Podle příznaku je odlišena pošta přijatá a odeslaná a také druh pošty. Obsahuje odkaz na přiložený soubor.

Druhy aktivit – do této tabulky Manažeři ukládají výčet činností, které je možné provádět na jednotlivých případech. Kromě samotného názvu činnosti je evidována i cena za hodinu provádění činnosti.

Aktivity – tato tabulka eviduje jednotlivé aktivity na spisu (z účetního hlediska). Může zde být zaznamenána práce na daném případě. V takovém případě se eviduje cena a čas práce. Uložení může být také pouze jednorázový výdaj (například soudní poplatek), který bylo nutné vynaložit na práci na daném případě. Tabulka má vazbu na tabulku Druhy_aktivit. Cena je evidována v této i tabulce Druhy_aktivit, což se může jevit jako duplicitní uložení. Cena v tabulce Druhy_aktivit je však pouze orientační a slouží jako výchozí pro daný typ aktivity. U každého jednotlivého záznamu aktivity lze tuto cenu nastavit libovolně.

Faktury – v této tabulce jsou uloženy údaje o fakturách, které vystaví advokátní kancelář svým klientům. Je zde uložena vazba na spis ke kterému se faktura váže, na klienta pro kterého je vystavena, datum vystavení a stav, který uvádí, zda-li již byla faktura zaplacená.

Faktura_položky – tato tabulka uchovává informace o jednotlivých položkách vystavených faktur. Každá položka má uloženu vazbu na fakturu a na účtovanou položku z tabulky Aktivity.

Jednání – do této tabulky se ukládají data, která jsou stažena ze systému InfoJednání. Kromě vazby na daný spis tabulka obsahuje původní adresu na systému InfoJednání, datum stažení informací, datum a čas jednání, místo a druh jednání, řešitele a účastníky jednání a další doplňkové informace.

Řízení – tabulka řízení uchovává data stažená ze systému InfoSoud. Je zde vazba na spis, původní adresa v InfoSoudu, datum uložení a také získané datum poslední změny v řízení.

Položky_řízení – v této tabulce jsou data o jednotlivých událostech řízení. Tabulka obsahuje vazbu na řízení, původní adresu události na InfoSoudu, název a datum události a také její popisný text.

Činy – zde jsou uchovávány jednotlivé záznamy o provedených akcích zaměstnanců. Přehled těchto akcí může sloužit jako kontrola práce zaměstnanců pro manažera. Ukládaná akce může být například vytvoření spisu, úprava údajů o klientovi, nahrání dokumentu nebo zaevidování nové pošty.

5.3 Návrhové vzory

Návrhové vzory řeší obecné problémy, které často vznikají při vývoji softwaru. Nejde přímo o konkrétní počítačový zdrojový kód, ale obecně o popis vhodného řešení dané situace nebo problému. Návrhové vzory lze obecně rozdělit do tří hlavních typů: Vytvářející vzory (řeší vytváření objektů, většinou dynamicky za běhu programu), Strukturální vzory (řeší vzájemné uspořádání tříd v systému) a Vzory chování (řeší chování systému, spolupráci objektů). Celá zde navrhovaná aplikace bude tvořena dle návrhového vzoru MVC (model, view, controller), detailnější popis této architektury je uveden v další kapitole.

5.3.1 Repository

V aplikaci bude využíván návrhový vzor Repository. Hlavní myšlenkou tohoto vzoru je odstínit veškerou práci přímo s datovým zdrojem (v tomto případě databází), nebo komponentou model v části, která se stará o logiku aplikace. Repository poskytuje veškeré metody pro práci s daty, které může tato část potřebovat (vkládání, úpravy, mazání, dotazy). Výhodou takového odstínění je, že v případě změny datového zdroje (jiný typ databáze, změna na XML zdroj) není nutné přepisovat kód na různých místech aplikace. Jediná změna, kterou je nutné provést, je ve třídě reprezentující Repository a všechny ostatní části systému fungují dále bez nutnosti jakýchkoliv dalších změn.

6 Použité technologie

Protože podle požadavků na systém, je třeba aby k systému mohlo nezávisle na sobě přistupovat více uživatelů a různých počítačů, rozhodl jsem se systém řešit jako webovou aplikaci. Přístup pomocí internetu umožňuje používání aplikace z jakéhokoliv počítače na světě, který je k této síti připojen a toto řešení je také nezávislé na platformě.

6.1 Volba platformy

Další věcí, kterou je před započítím implementace nutné zvolit, je platforma, na které bude systém vyvíjen a na které bude později provozován. Pro vytvoření moderní webové aplikace máme na výběr z mnoha variant. V současné době je možností opravdu velké množství, proto níže uvádím pouze takové, které jsem reálně zvažoval.

První variantou je zvolit platformu LAMP. Jde o zkratku z hlavních technologií, které se zde využívají: operační systém Linux, webový server Apache, databázový systém MySQL a programovací jazyk PHP. Je velmi snadné naučit se základy těchto technologií a rychle začít vytvářet webové stránky. Často jsou takové systémy kritizovány za špatnou strukturu a nevhodné programátorské návyky, protože jazyk PHP nenutí programátory do nějaké pevně stanovené architektury aplikace. Proto v poslední letech velmi vzrostla obliba různých frameworků, které jednak práci zjednodušují a jednak zavádějí určitá vhodná pravidla, která celou architekturu aplikací zlepšují (populární jsou například český Nette, nebo Zend framework). Ačkoliv jde o oblíbenou a otevřenou platformu, nakonec jsem se pro toto řešení nerozhodl. Důvodem je to, že tuto platformu nepovažuji do budoucna za perspektivní oproti jiným řešením pro webové aplikace.

Další možností je řešit vše pomocí technologií z rodiny Java. Jako webový server nám zde může sloužit GlassFish, což je aplikační server od společnosti Sun Microsystems pro platformu Java EE. Samotná webová aplikace může být poté postavena například na architektuře MVC (více v kapitole 6.3). Tuto technologii považuji za perspektivní a efektivní pro tvorbu webových systémů, ale nezvolil jsem ji z důvodu menších zkušeností, které mám s jazykem Java.

Poslední možností, pro kterou jsem se nakonec rozhodl je ASP.NET (Active Server Pages). Konkrétně nejmodernější variantu, ASP.NET MVC. Jde o technologii od firmy Microsoft, která za svoji existenci prošla mnoha vývojovými fázemi. Na pozadí těchto aplikací běží operační systém Windows, jako webový server je využit IIS (Internet Information Services), také od firmy Microsoft. Jako databázový systém se nejčastěji využívá MSSQL od stejné firmy. Pomocí frameworků MVC a Entity lze velmi efektivně vytvářet a spravovat rozsáhlé a výkonné webové aplikace. Technologie se velmi rychle rozvíjí a má podporu silné technologické firmy. Do budoucna považuji tento přístup jako nejperspektivnější z výše uvedených a proto jsem se rozhodl právě pro ASP.NET MVC.

6.2 ASP.NET

ASP.NET je součástí .NET Frameworku a slouží pro vytváření webových aplikací. Tato technologie má za sebou několik stupňů vývoje.

Původní ASP byla první technologie od firmy Microsoft určená pro vývoj dynamických webových aplikací. Jde o dynamickou skriptovací platformu pro zpracování serverové části webových stránek. Systém fungující na ASP můžeme poznat podle přípony .asp v adrese. ASP nelze chápat jako programovací jazyk, pro programování aplikací v ASP můžeme využít více programovacích jazyků, nejčastěji je to VBScript a Jscript. ASP podporuje objektový přístup, ale pouze částečně. Existují zde určité systémové objekty, se kterými lze pracovat, nelze ale vytvářet či odvozovat své další nové třídy.

Prvním rozšířením, kterého se ASP dočkalo, je ASP.NET. Tato technologie již využívá .NET Framework. Díky tomu lze ASP.NET aplikace programovat v mnoha jazycích (například Visual Basic, C#) a programátor má také k dispozici velké množství knihoven. Na rozdíl od skriptovacího

ASP, je ASP.NET předkompilován do DLL slouborů, čímž je výrazně rychlejší než jeho předchůdce. Definované ovládací prvky lze využít jako šablony, čímž se redukuje duplicitní kód. ASP.NET také již umožňuje cachovat celou, nebo pouze části stránky, což je důležité pro zvýšení výkonu serveru a tím i celé aplikace.

Nástupcem ASP.NET je ASP.NET WebForms. V návaznosti na vývojové prostředí (MS Visual Studio) můžeme v této technologii lehce vytvářet webové stránky. Lze využívat předpřipravené komponenty a z nich skládat celý obsah stránky (podobně jako u desktopové aplikace ve WinForms). WebForms se pomocí HTML a JavaScriptu snaží zakrýt bezstavovost HTTP protokolu a snaží se zavést stavové prostředí pomocí dvou přístupů:

- ViewState: informace o uživateli jsou uchovávány mezi jednotlivými požadavky v zakódovaném tvaru v neviditelných formulářových polích. Výhodné je, že není třeba žádné další technologie, využívá se prosté HTML. Nevýhodou je zvýšený přenos mezi serverem a klientem – uživatelem a také pokud mezi stránkami nepřejdeme pomocí odeslání formuláře skrz metodu POST, ale například pomocí odkazu, přijdeme o všechna uložená data.
- Session state: v tomto přístupu jsou naopak všechna data uložena na session na serveru. Uživatel je identifikován pouze pomocí cookie nebo předávaného parametru v adrese.

ASP.NET WebForms je v současnosti stále oblíbenou a využívanou variantou pro řešení webových aplikací.

Nejnovějším přístupem, který ASP.NET zavádí, je ASP.NET MVC. Jde o framework, který umožňuje vytvářet webové aplikace s architektonickým vzorem MVC (model, view, controller). Hlavní výhodou oproti WebForms je nezávislost na JavaScriptu a výrazně snadnější možnost testování napsaného kódu. Právě tento přístup jako nejmodernější a nejrychleji se rozvíjející jsem se rozhodl využít pro tvorbu řešené aplikace.

6.3 MVC

MVC je framework, který začala firma Microsoft vyvíjet v roce 2007. Aktuálně se nachází ve verzi 4, nicméně dokončená verze 5 by měla být vydána v průběhu roku 2014. Tento framework značně usnadňuje tvorbu webových aplikací podle architektury Model / View / Controller.

Vzor MVC je softwarová architektura, která rozděluje aplikaci do tří částí. Datový model, koncové uživatelské rozhraní a logiku aplikace. Každá část má na starosti svoji práci a může komunikovat s jinou částí.

- Model: slouží k práci s daty. Stará se o zápis i čtení dat, nezávisle na tom z jakého zdroje. Může pracovat s datovými soubory, soubory ve formátu XML či nějakým typem databáze. Jde o určitou abstrakci, kdy ostatní části aplikace se nezajímají, jaký je použit datový zdroj, protože pracují pouze s modelem, který tento zdroj zastřeší.
- View: zajišťuje uživatelské rozhraní. Nestará se o žádnou aplikační logiku, ani práci s daty, pouze o to, jakým způsobem zobrazit data, která jsou do View předána. V ASP.NET MVC jsou jako View použity šablony (ASPX nebo novější varianta Razor). Rozhraní často odpovídá modelu (například výpis tabulky), ale není s ním nijak napevno spjato. Celkový výstup aplikace, který je zobrazen uživateli je většinou složen z více jednotlivých View.
- Controller: spojuje komponenty View a Model a zajišťuje aplikační logiku. Spolupracuje s komponentou Model, od které získává data, se kterými může dále pracovat. Poté, co jsou data zpracována na základě logiky aplikace, jsou předána do komponenty View k zobrazení klientovi. Na opačnou stranu Controller přijímá a zpracovává data odeslaná uživatelem, která může dále upravit a poslat Modelu k uložení.

V ASP.NET MVC při zpracování požadavku, který zašle klient na webový server, probíhá mnoho aktivit. Obecně lze průběh zpracování popsat takto:

- uživatel provede nějakou akci (stiskne tlačítko a odešle tak formulář, klikne na odkaz)

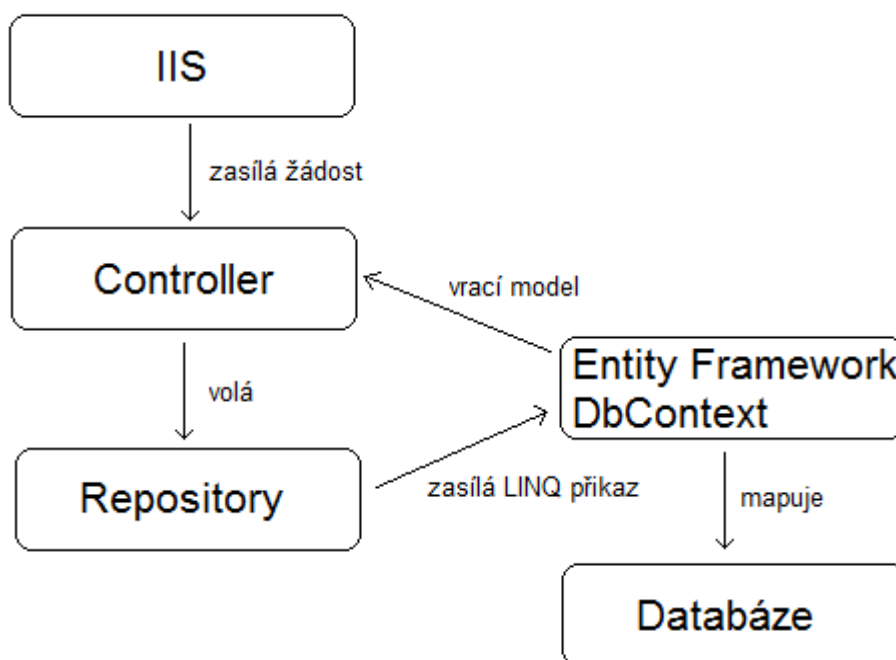
- vyhodnotí se routování (podle adresy, na kterou je zaslán požadavek), dle uložených pravidel se určí, který konkrétní Controller (třída) a která jeho metoda má být pro danou akci zavolány.
- Je zavolán vybraný Controller, který provádí další aktivitu. Může zpracovat zasláná data uživatelem. Může načítat nová data z modelu, nebo modelu zaslat nová či aktualizovaná data a sám aktualizovaný model použít dál. Jako finální akci Controller zavolá View, do kterého může a nemusí předat data pro zobrazení.
- vyhodnotí se volané View (složí se z jednotlivých podčástí, jsou doplněna vložená data) a výstup je poslán k zobrazení klientovi [12] [15].

6.4 Entity Framework

Entity Framework¹ je ORM (objektově-relační mapování) s otevřeným zdrojovým kódem pro .NET Framework. Pokud modelujeme data, většinou se snažíme věrně zachycovat realitu. Pro tyto účely vzniklo objektově orientované programování, entita reálného světa je reprezentována jako objekt. Většina databází (včetně MSSQL) je ale relační, tedy množina tabulek, jejich řádků a vazeb mezi nimi. Kvůli této rozdílné reprezentaci dat vznikly technologie ORM, které mapují relační databázi na objektový model. Odstiňují tak práci s databází a přímými SQL dotazy a nahrazují ji prací s objekty. Nejčastější využití je pro zápis, ukládání a úpravu dat. Pro určité složitější dotazy není možné se SQL dotazům vyhnout. Hlavním cílem ORM je synchronizace mezi datovým modelem aplikace a jeho reprezentací uloženou v relační databázi a to při zachování perzistence dat.

Pro implementaci aplikace jsem se rozhodl Entity Framework využít, protože výrazně zjednodušuje práci s databází a velmi dobře spolupracuje s ASP.NET MVC.

Entity Framework je aktuální ve verzi 6, je vyvíjen firmou Microsoft a je vydáván pod licenci Apache License v2.



Obrázek č. 6.1 – Provedení požadavku s využitím Repository a Entity Frameworku

¹ Domovské stránky Entity Framework - <http://msdn.microsoft.com/en-us/data/ef.aspx>

6.5 HTML5

HTML5 je nejnovější verze značkovacího jazyka pro webové dokumenty. Prošla dlouhým vývojem a schvalováním a konečná verze specifikace má být schválena koncem roku 2014.

Od verze 4 přináší velké množství novinek, které si vyžádal moderní přístup k webovým aplikacím. Hlavními novinkami které verze 5 přináší jsou:

- nové HTML značky, které definují sémantiku daného elementu. Dříve bylo nutné všechny sekce stránky strukturovat například pomocí značky `<div>`, jedinou přidanou informací mohl být název `id`, nebo `class`. Nyní je možné typické části stránky vložit do speciálních elementů, které již vyjadřují sémantiku dané části. Takto označit můžeme například navigaci značkou `<nav>`, patičku stránky značkou `<footer>`, nebo článek značkou `<article>`. Tyto nové značky zjednodušují práci vyhledávacím enginům (nebo obecně automatickému zpracování dokumentů), které mohou lépe porozumět významu dané sekce v HTML dokumentu.
- relační databáze s podporou transakcí a trvalé úložiště dat pomocí asociativního pole. Dříve bylo jedinou možností lokálního uložení dat cookies, které mají ale velmi omezené možnosti. HTML5 umožňuje ukládat lokálně i větší a komplikovanější strukturovaná data offline. Interaktivní webová aplikace tak může nadále fungovat, i pokud je uživatel offline. A až poté, co se opět připojí k internetu, odešle lokální data ke zpracování na server.
- možnost vkládat do stránky multimediální obsah. Do verze 4 bylo možné vkládat multimediální obsah (videa, zvuk) pouze pomocí externích pluginů nebo Flashe. HTML5 má již podporu pro nové značky `<video>`, `<audio>`, `<source>`, `<track>`, které vkládají multimediální obsah přímo do dokumentu.
- zjednodušení zápisu. Hlavička HTML dokumentu bývala dříve plná dlouhých definic. HTML5 výrazně zjednodušuje nutnost těchto počátečních definic. Dříve vypadala hlavička HTML dokumentu takto:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>title</title>
<meta name="Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
```

V HTML 5 lze udělat ekvivalentní zápis o mnoho jednodušeji:

```
<!DOCTYPE html>
<html>
<head>
<title>title</title>
<meta charset="UTF-8">
</head>
```

Využití HTML5 je nutností pro každou moderní dynamickou stránku, a protože je dostatečně podporováno všemi aktuálními prohlížeči, rozhodl jsem se ho zvolit pro svoji aplikaci.

6.6 Responzivní design

Tento pojem zavedl americký programátor jménem Ethan Marcotte². Nejde o žádnou novou samostatnou technologii, ale o princip, jakým je HTML dokument stylován pomocí kaskádových

² Webové stránky autora - <http://ethanmarcotte.com/>

stylů. Výsledkem responzivního designu by měl být HTML dokument, který je optimalizován pro nejrůznější druhy zařízení s různými velikostmi displeje (při naprosto totožném HTML a CSS kódu). Není tak nutné vytvářet více verzí webu, jak bylo v minulosti zvykem (základní a mobilní), pouze vhodně upravit šablonu stylů pro jedinou verzi. Díky nejnovější verzi CSS (verze 3) je možné rozpoznat určité vlastnosti zařízení, které je použito pro zobrazení a dle toho dokument přispůsobit. Responzivní design se skládá ze tří hlavních úrovní:

- Flexibilní struktura – této úrovni je dosaženo, když pro jednotlivé prvky dokumentu není využívána fixní šířka (např. v pixelech), ale je zadána pomocí procent. Takto definované prvky mají potom relativní velikost k ostatním prvkům i celému průzoru zařízení bez ohledu na jeho velikost.
- Flexibilní obrázky – tato technika zajišťuje, že šířce dokumentu se nebude přispůsobovat pouze struktura dokumentu, ale i vložené obrázky. Této vlastnosti se docíli velmi snadno, pouze pomocí jedné definice pro všechny vložené obrázky:

```
img{
    max-width: 100%;
    height: auto;
}
```

- Media Queries – tato úroveň responzivního designu je nejdůležitější. Umožňuje totiž měnit stylování dokumentu v závislosti na šířce průzoru prohlížeče na daném zařízení. Můžeme tedy vytvořit několik variant designu (kde CSS a HTML kód zůstává beze změn). Lze tak vytvořit například zvláštní variantu designu pro běžné zobrazení (šířka $\geq 1000\text{px}$), zobrazení na tabletu (šířka $\geq 480\text{px}$ a zároveň $< 1000\text{px}$) a pro zobrazení na mobilním telefonu (šířka $< 480\text{px}$). Zápis podobného rozčlenění (na dva typy zobrazení) může vypadat takto:

```
@media (min-width: 1000px) {
    #header_text {
        font-size 20px;
        color: #ff0000;
    }
}

@media (max-width: 999px) {
    #header_text {
        display: none;
    }
}
```

Takto nastýlovaný box, bude na zařízení se šířkou alespoň 1000 pixelů zobrazen s červeným textem velikosti 20 pixelů. Na zařízení s menší šířkou displeje nebude box zobrazen vůbec.

Protože jedním z požadavků na systém je možnost správného zobrazení na mobilních zařízeních, pokud nechceme vytvářet zvláštní verzi webu speciálně pro mobilní zařízení, je nutné využít v aplikaci možnosti responzivního designu.

6.7 Licence a cena technologií

Dle předchozích podčástí této kapitoly je zřejmé, že většina vybraných technologií je z rodiny produktů firmy Microsoft. Tato firma je známá tím, že většina jejích produktů nebývá otevřeným

softwarem či softwarem licencovaným zdarma. Dá se tedy předpokládat, že nasazení a další provoz produktu si vyžádá určité finanční investice do technologií. Obecně jsou dvě možnosti, jak k tomuto problému přistoupit.

6.7.1 Vlastní provoz

První možností je provozovat aplikace na svém serveru a řešit zajištění technologií svépomocí. Kromě zajištění fyzického serveru pro běh aplikace (tato část nepatří do oblasti řešení této práce) je nutné zajistit operační systém.

Nejvhodnější variantou operačního systému od firmy Microsoft je Windows Server (aktuální verze 2012). Tato verze systému navíc od běžných desktopových verzí Windows nabízí další pokročilé funkce pro provoz serveru, čímž se samozřejmě i výrazně zvyšuje cena licence tohoto produktu (cena začíná na 16 tisících Kč). Vzhledem k nenáročnosti řešené aplikace bude naprosto dostačující využít běžnou desktopovou variantu Windows 8, tu je možné pořídit již od 2500Kč³.

Další zmíněnou technologií je MVC framework. Ten je uvolněn pod Microsoft Public License (Ms-PL)⁴. Tato licence patří k typu otevřeného softwaru. Umožňuje distribuci kompilovaného kódu jak pro nekomerční tak i komerční účely pod licencí v souladu s Ms-PL. Pořizovací náklady na MVC framework jsou tedy nulové.

Další technologií je databázový server, v tomto případě SQL Server (aktuální verze 2012). Firma Microsoft nabízí různé edice tohoto produktu. Nejjednodušší verzí, avšak dostačující pro účely této aplikace, je edice SQL Server Express, který je poskytován zdarma⁵. Tato verze je vhodná pro malé klientské aplikace či menší webové servery. Dále využívaný Entity Framework je také uvolněn pod bezplatnou licencí.

Všechny ostatní technologie zmiňované v předchozích podkapitolách jsou také dostupné bezplatně. Proto jediným nákladem pro tento typ řešení je licence pro operační systém, což je pouze jednorázový náklad. Finančně se toto řešení zdá výhodné, problémem ale může být vlastní správa serveru. Dá se předpokládat, že menší a střední advokátní kanceláře nebudou zaměstnávat IT specialistu, ale zároveň zaměstnanci nebudou mít dostatečné znalosti a zkušenosti pro správu serveru.

6.7.2 Hosting

Druhou možností, jak řešit zajištění technologií potřebných pro provoz, je využít služby firem, které nabízí webhosting. Webhosting znamená pronájem prostoru a výkonu na cizím serveru. Tento server je připojen k internetu a umožňuje umístit webové stránky na internet i bez vlastního technického zázemí. Kromě samotného prostoru na disku poskytovatel hostingu nabízí další doplňující služby, jako například e-mailové schránky, databáze či zálohování dat.

Dostupné jsou i tzv. freehostingy, které poskytují webhosting zdarma, obvykle je potom na provozovaných stránkách vložena reklama poskytovatele. U freehostingů také nebývá obvyklá podpora pro řešení problémů, což je věc, která je samozřejmě nutná, pokud jde o provoz seriózní webové aplikace. Vhodnější variantou je tedy placený hosting, který garantuje dostupnost a uživatelskou podporu. V České Republice působí velké množství takových firem, menší část z nich poskytuje typ služeb vhodných pro řešenou aplikaci (podpora pro .NET). Cenově jsou nabídky těchto firem v rozmezí od několika desítek korun měsíčně až zhruba po tisíc korun za měsíc⁶. Roční provoz vhodného hostingu pro malou aplikaci může tedy ročně vyjít na jeden až dva tisíce korun v závislosti na vybrané firmě a tarifu. Ve srovnání s vlastním provozem jde zejména v dlouhodobém horizontu o dražší variantu, nicméně vzhledem k problémům zmíněným v předchozí podkapitole ji shledávám jako vhodnější řešení.

³ Dle srovnávače zboží Heureka - <http://www.heureka.cz>

⁴ Licence ASP MVC - <http://www.asp.net/mvc/open-source>

⁵ SQL Server 2012 Express - <http://msdn.microsoft.com/cs-cz/evalcenter/hh230763.aspx>

⁶ Dle srovnávače hostingů - <http://www.hostingy.cz/webhosting.html>

7 Napojení na veřejné databáze

Veřejné databáze jsou většinou zřizovány státem (jednotlivými ministerstvy) a nabízejí obrovské množství dat. Velká část z těchto dat je důležitých právě pro advokátní kanceláře. Díky přístupu k těmto datům můžeme jednoduše získat různé informace o fyzických osobách, právnických osobách, společnostech nebo třeba o soudních jednáních. Nejzajímavější data vzhledem ke kontextu řešené aplikace nabízejí systémy eJustice, systém Ares a Insolvenční rejstřík. V této kapitole jsou podrobněji rozepsány informace o každém z těchto systémů.

7.1 InfoSoud

V průběhu roku 2008 spustilo Ministerstvo spravedlnosti pod vedením ministra Jiřího Pospíšila nové projekty, které mají za úkol zjednodušit přístup k velkému množství informací, které vydávají soudy v České republice. Přístup ke zveřejňovaným datům je umožněn pomocí webových aplikací. Tyto systémy ale nejsou příliš uživatelsky přívětivé a neumožňují žádné pokročilé funkce, jako například přihlášení a automatizované sledování jednotlivých případů.

Jako první ze dvou projektů byl v dubnu 2008 spuštěn webový portál s názvem infoSoud na adrese infosoud.justice.cz. Cílem tohoto portálu je omezit nutnost navštěvovat fyzicky soudy kvůli získávání informací o konkrétních soudních řízeních. Tato služba umožňuje veřejnosti (za předpokladu že znají nutné identifikační údaje, což jsou v tomto případě jméno příslušného soudu a spisová značka) získat informace o průběhu soudního řízení, případně rovnou se soudy komunikovat. V databázi systému jsou uloženy informace o všech spisech, které byly započaty od 1.1.2006. Data do systému InfoSoud jsou přebírána automaticky z informačních systémů provozovaných jednotlivými soudy.

Po zadání potřebných identifikačních údajů získáme seznam všech událostí, které náleží do daného řízení. U každé události lze zobrazit její detail. O každé události tak můžeme získat informace například o datu, předmětu řízení, aktuálním stavu věci a další [8].

7.2 InfoJednání

Druhá část systému eJustice, která nese jméno InfoJednání byla spuštěna v říjnu roku 2008. Cílem tohoto systému je zprostředkovat uživatelům informace, který den a od kolika hodin probíhá konkrétní soudní jednání. Systém poskytuje informace o všech nařízených jednáních 30 dnů dopředu od chvíle položení dotazu. Tyto informace obsahují číslo jednací síně, čas a datum jednání, jméno řešitele (což je soudce, který řídí dané jednání), spisovou značku, předmět jednání, přehled jeho účastníků a také zda-li je toto jednání veřejné či ne. Nutné údaje pro získání informací jsou stejně jako v případě InfoSoudu spisová značka a jméno soudu. Data jsou opět získávána automaticky z jednotlivých provozních systémů [9].

7.3 ARES

Předchozí uvedené systémy sloužily k získávání dat o soudních procesech, nenalezneme v nich ale žádné detailnější údaje o zúčastněných stranách v těchto spisech. Právě k tomuto slouží další z veřejných databází, systém ARES (Administrativní Registr Ekonomických Subjektů). Opět jde o webovou aplikaci, tentokrát náležící pod Ministerstvo financí. V systému je možné dohledávat informace o všech ekonomických subjektech, které jsou registrované v České republice. ARES získává data z jednotlivých informačních systémů státní správy, například Obchodní rejstřík, Živnostenský rejstřík, Registr ekonomických subjektů a další. Systém umožňuje automatické získávání informací pomocí připravených XML služeb. Tento export je omezen počtem dotazů (1000

dotazů v době od 8:00 hod. do 18:00 hod, 5000 dotazů mimo tento hlavní čas⁷), nicméně vzhledem k řešené aplikaci (a předpokládanému počtu uživatelů) by se toto omezení nemělo projevit jako problém [10].

7.4 Insolvenční rejstřík

Dalším významným zdrojem informací je Insolvenční rejstřík. Jde o veřejný seznam, ve kterém jsou uvedeny subjekty, které procházejí insolvenčním řízením (řízení o projednání úpadku v případě fyzických osob). Jde o subjekty typu Fyzické osoby podnikatelé, Nepodnikatelské subjekty a Právnícké osoby. Tyto subjekty jsou zapsány do insolvenčního rejstříku, když je proti nim podán návrh na zahájení insolvenčního řízení (konkurs, oddlužení). Takové údaje je vhodné získávat zejména o svých klientech, aby se advokátní kancelář mohla vyhnout klientům, kteří nebudou schopni či ochotni zaplatit za vykonané služby. Systém poskytuje službu pro vkládání a získávání dat pomocí technologie SOAP⁸. Rozhodl jsem se ale pro získávání dat pomocí zaslání dotazu a parsování výsledku. Tuto možnost považuji za rychlejší, protože SOAP služby jsou určeny spíše pro vkládání a manipulaci s již existujícími daty. I pro Insolvenční rejstřík je stanoven limit dotazů, maximální počet je 3000 požadavků za den, nebo 50 požadavků za jednu minutu⁹. Při překročení těchto hodnot může dojít k omezení přístupu. Pro řešenou aplikaci by opět mělo jít o dostatečný limit, který nepředstavuje problém [11].

7.5 Způsob získávání dat

Systémy eJustice nenabízejí export dat, který by bylo možné využít pro automatické zpracování informací. Systémy zpřístupňují data pouze pomocí webového rozhraní. Jedinou možností jak tedy může aplikace automatizovaně získat data z těchto zdrojů, je vytvoření URL s požadovanými parametry (soud, spisová značka, IČ) a zavolání tohoto požadavku na cílový server. Webový server potom vrátí HTML výstup, který je nutné rozparsovat, abychom získali data, o která máme zájem. Až poté můžeme data uložit, nebo s nimi dále pracovat.

7 http://www.info.mfcr.cz/ares/ares_podminky.html cz

8 Specifikace SOAP - <http://www.w3.org/TR/soap/>

9 <https://isir.justice.cz/isir/common/stat.do?kodStranky=PROVOZPODMINKY>

8 Implementace systému

V této kapitole je popsáno technické řešení projektu, jehož návrh je uveden v předchozích kapitolách. V počátečních podkapitolách jsou rozepsány detailnější informace o tvorbě nového projektu v technologii ASP.NET MVC pomocí vývojového prostředí Visual Studio, které jsem pro tuto aplikaci využíval. V dalších podkapitolách jsou detailněji rozebrány konkrétní stěžejní a složitější funkce, které systém nabízí.

8.1 Vytvoření nového projektu

Při vytváření nového projektu ve vývojovém prostředí MS Visual Studio vybereme volbu ASP.NET MVC 4 Application, což je projekt v nejnovější vydané verzi frameworku pro vývoj webových aplikací. V dalším kroku zákládání nového projektu je třeba vybrat šablonu pro aplikaci. Na výběr máme několik možností:

Empty: zcela prázdný projekt, pouze s vytvořenou adresářovou strukturou, ale žádnou předpřipravenou funkcionalitou.

Basic: prázdný projekt s připravenou HTML šablonou pro zobrazení stránek, připraveným souborem pro definice CSS a přiloženými JavaScriptovými knihovnamy.

Internet Application: Projekt obsahující controller pro správu uživatelského účtu, formuláře pro autentifikaci a registraci nových uživatelských účtů. Vygenerovaná aplikace je již funkční a s její omezenou funkcionalitou ji lze využívat. Tato možnost je nejvhodnější při tvorbě nové aplikace, protože sama vygeneruje funkcionalitu, kterou by bylo stejně nutné vytvořit ručně. Právě tuto volbu šablony jsem při vytváření aplikace využil. Vygenerovaná a spuštěná demo aplikace je zobrazena na obrázku 8.1.

Intranet Application: Vývojové prostředí v tomto případě vygeneruje podobnou aplikaci jako při volbě Internet Application. Narozdíl od této možnosti je zde ale využívána Windows autentizace.

Mobile Application: U této možnosti je opět vytvořena demo aplikace. Co se týká funkcionality, je shodná s volbou Internet Application, ale vytvořené HTML šablony jsou speciálně uzpůsobeny právě pro zobrazení na mobilních telefonech.

Web API: při této volbě je vytvořen projekt, který slouží pro snadné poskytování webových služeb.

Home Page. Modify this template to jump-start your ASP.NET MVC application.

To learn more about ASP.NET MVC visit <http://asp.net/mvc>. The page features [videos, tutorials, and samples](#) to help you get the most from ASP.NET MVC. If you have any questions about ASP.NET MVC visit [our forums](#).

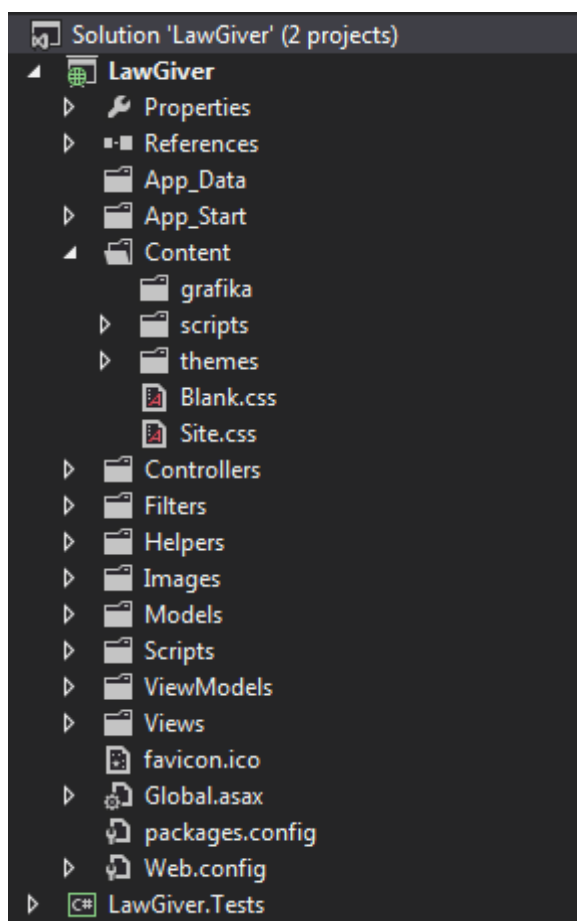
We suggest the following:

- 1 Getting Started**
ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and that gives you full control over markup for enjoyable, agile development. ASP.NET MVC includes many features that enable fast, TDD-friendly development for creating sophisticated applications that use the latest web standards. [Learn more...](#)
- 2 Add NuGet packages and jump-start your coding**
NuGet makes it easy to install and update free libraries and tools. [Learn more...](#)
- 3 Find Web Hosting**
You can easily find a web hosting company that offers the right mix of features and price for your applications. [Learn more...](#)

Obrázek č. 8.1 – Nově založený projekt ASP.NET MVC 4 Application

8.2 Struktura projektu

Po založení nového projektu vytvoří MS Visual Studio standardní šablonu s již předpřipraveným demo systémem, který již zvládá autentizaci uživatelů. Při tvorbě reálného projektu je nutné strukturu rozšířit o další položky. Finální verze stuktury projektu je zobrazena na obrázku 8.2.



Obrázek č. 8.2 – Struktura projektu v MS Visual Studio

V celkovém řešení je vidět dva projekty. První nese název LawGiver a jde o samotný řešený systém. Druhý projekt LawGiver.Tests slouží pro testy.

Projekt LawGiver obsahuje velké množství dalších podadresářů a souborů, mezi ty důležité patří:

Properties: obsahuje soubor `AssemblyInfo.cs`, který obsahuje základní informace o výsledné sestavené aplikaci. Je zde tedy uveden například název, popis, autor, společnost či ochranná známka.

References: zde jsou obsaženy knihovny, které jsou připojené k projektu. Nové knihovny je možné snadno připojit pomocí balíčkovacího systému NuGet.

App_Start: v tomto adresáři jsou uloženy konfigurační soubory pro celý projekt. `AuthConfig.cs` obsahuje třídu, jejíž jednotlivé metody slouží k předpřipravené autentizaci pomocí třetích stran. Všechny metody této třídy jsou zakomentované, stačí však pouze odkomentovat danou metodu a celá demo aplikaci již bude umožňovat daný způsob autentizace. Předpřipravené jsou možnosti pro autentizaci pomocí služeb LiveID (Microsoft), Twitteru, Facebooku a Google účtu. Zde řešená aplikace žádnou z těchto možností nevyužívá a spoléhá se pouze na přihlašování pomocí lokálního účtu.

V souboru `RouteConfig.cs` je uloženo nastavení pro mapování adres na controllery aplikace, více o samotném mapování je uvedeno v kapitole 8.2

Content: tento adresář obsahuje další nutné soubory pro finální aplikaci, které ale nejsou součástí samotného programového kódu. Jsou zde obsaženy například jednotlivé obrázky pro grafiku, šablony CSS či soubory JavaScriptu.

Controllers: v této složce jsou uloženy všechny použité controllery. Název souborů je vždy ve tvaru `JmenoController.cs`. Tento tvar je nutné zachovat pro správné namapování adres.

Filters: zde jsou umístěny třídy, které slouží pro filtrování přístupu ke controllerům. Pokud například chceme povolit přístup ke controlleru pouze přihlášeným uživatelům, vytvoříme zde daný filtr. Před každým controllerem nebo pouze jeho metodou, u které chceme omezit přístup, pak tento filtr uvedeme.

Helpers: do tohoto adresáře patří pomocné třídy. V řešené aplikaci jsou využity dvě. První je `EJusticeParser.cs` a slouží k získávání dat ze systému eJustice, druhá třída `Courts.cs` obsahuje kolekce značek a názvů českých soudů, které jsou využívány pro sestavování dotazů pro systémy EJustice.

Models: složka Models obsahuje jednotlivé třídy pro každou datovou entitu, která se v systému vyskytuje (spis, klient, dokument, faktura atd.). Pomocí Entity Frameworku je podle těchto tříd automaticky vytvořena databázová struktura.

ViewModels: v tomto adresáři jsou uloženy pomocné třídy, které jsou zasílány do Views. Tyto pomocné třídy se využívají v případě, že samotná třída z Models není dostatečná a je nutné zaslat více dat. Většinou je třída z ViewModels složena z více jednotlivých Models tříd (například položky faktury a faktura samotná).

Views: zde jsou umístěny Views. Ke každému existujícímu controlleru je zde jedna složka, ve které jsou jednotlivá View, které daný controller využívá. Kromě těchto jsou zde parciální Views, což je speciální případ View, který obsahuje pouze nějakou menší část, ze které se potom skládají další dohromady. Poslední speciálním případem Views jsou Layouty. To jsou šablony, které obsahují statické části stránek, které se nemění. Do těchto statických Layoutů se pouze do obsahové části vloží klasické View.

Web.config: tento XML soubor slouží jako konfigurace pro celý projekt, je zde například uloženo spojení do databáze, způsoby autentizace a další nastavení.

8.3 Adresní struktura

Struktura adres je definovaná v souboru `RouteConfig.cs` pomocí routovacích pravidel. Jednotlivá pravidla jsou uložena v tomto tvaru:

```
routes.MapRoute(
    name: "Default",
    url: "{controller}/{action}/{id}",
    defaults: new { controller = "Home", action = "Index", id =
        UrlParameter.Optional }
);
```

Položka `url` stanovuje obecný tvar adresy, v tomto výše uvedeném případě `{název controlleru}/{název akce}/{parametr}`. Tedy například adresa `domena/Client/Detail/1` bude namapována na controller `Client`, u něhož bude zavolána metoda `Detail` s parametrem `1`. Jako poslední část

se uvádí výchozí akce. Tedy controller a metoda, které mají být použity jako výchozí při zadání pouze domény bez dalších částí za lomítkem. Pokud nechceme využít této možnosti mapování, je možné před každou jednotlivou metodou v controllerech definovat adresu, při které má být zavolána.

8.4 Modely

Struktura databáze se pomocí Entity Frameworku vytváří velmi snadno. Jednotlivé modely objektu se popíší jako běžná třída a Entity Framework podle těchto tříd sám vytvoří strukturu databáze. Každá vytvořená třída znamená ve výsledné databázi tabulku, a každý její atribut je jeden sloupec tabulky. Tyto třídy neobsahují žádné metody. Pokud je typ atributu jiná, vlastní třída z adresáře Models, lze ji označit pomocí klíčového slova `virtual` a ve výsledné databázi slouží tento sloupec jako cizí klíč do další tabulky.

Tyto modelové třídy ale neslouží pouze ke generování struktury databáze, v MVC usnadní i mnoho dalších věcí. Pomocí atributů ze třídy `System.ComponentModel.DataAnnotations` můžeme více specifikovat chování a omezení těchto atributů (sloupců) v rámci aplikace.

Prvním druhem jsou atributy, které omezují hodnoty, jakých může atribut nabývat. Můžeme tak například definovat povinný atribut pomocí `[Required]`, primární klíč tabulky pomocí `[Key]`, určit minimální délku `[MinLength(20)]` či stavit přímo tvar pomocí regulárních výrazů `[RegularExpression]`. U každého takové omezení je možné přímo ve třídě uvést chybovou hlášku, která bude uživateli vrácena při nedodržení omezení. V samotném controlleru se již pouze zkontroluje, zda-li je objekt validní pomocí atributu `IsValid` (například při vytváření nové položky) a více se již není nutné o tato omezení starat. Šablony, které MS Visual Studio vygeneruje pro editaci těchto položek, umí dokonce ověřovat platnost těchto atributů již za běhu pomocí JavaScriptu.

Druhým typem jsou atributy, které upravují vzhled a chování atributů v uživatelském výstupu aplikace. Důležitý je atribut `[DisplayName]`, jehož hodnota bude poté uváděna u dané položky (například jako nadpis sloupce u seznamu, nebo před textovým polem při zadávání hodnoty). Tuto hodnotu je pak samozřejmě možné v šabloně ručně změnit. Dále máme k dispozici atribut `[DisplayFormat]`, který můžeme využít například u atributu typu `DateTime` (datum a čas) pro formátování vypsaného řetězce. Mezi ty nejvíce užitečné patří také atribut `[DataType]`, které určí například jaký druh editačního prvku se pro položku má využít (textarea, input text). Využívání obou typů atributů velmi usnadňuje tvorbu nejen logiky aplikace, ale s využitím MS Visual Studia i uživatelského rozhraní. Ukázka definice takové modelové třídy je uvedena níže:

```
public class Client {
    [Key]
    public int ClientId { get; set; }

    [DisplayName("Jméno")]
    [Required]
    public string Name { get; set; }

    [DisplayName("Příjmení")]
    [Required]
    public string Surname { get; set; }

    [DisplayName("Datum narození (d.m.yyyy)")]
    [DisplayFormat(DataFormatString= "{0:dd.MM.yyyy}",
ApplyFormatInEditMode = true)]
    public DateTime Birth { get; set; }

    [DisplayName("Ulice")]
    public string Street { get; set; }
```

```

        [DisplayName("PSČ")]
        public string PSC { get; set; }

        [DisplayName("Město")]
        public string City { get; set; }

        [DisplayName("IČ"), RegularExpression(@"\d{8}|\.{0}",
ErrorMessage = "IČ není ve správném formátu.")]
        public string Ic { get; set; }

        [DisplayName("Telefon")]
        public string Phone { get; set; }

        [DisplayName("E-mail")]
        [DataType(DataType.EmailAddress)]
        [EmailAddress(ErrorMessage="Zadaná emailová adresa není
platná")]
        [Required]
        public string Email { get; set; }
    }

```

8.5 Práce s databází

Práce s daty je v aplikaci běžnou činností. Obecně lze tuto práci rozdělit na čtyři typy: vkládání nových dat, aktualizace současných dat, mazání dat a získávání dat z databáze. Abychom v Controlleru mohli pracovat s databází, musíme vytvořit objekt typu `LawGiverRepository` (návrhový vzor `Repository`), jehož definice je v `Models/LawGiverRepository.cs`. Tento objekt již navenek nabízí samotné metody, které slouží pro konkrétní databázové operace. Uvnitř však s databází nepracuje přímo, ale pomocí `Entity Frameworku`. Obsahuje privátní atribut typu `LawGiverDataContext` z `Models/LawGiverDataContext.cs`. Tento objekt obsahuje jednotlivé atributy typu `DbSet<datový typ>`. Pokud do této třídy zapíšeme nový atribut tohoto typu, `Entity Framework` se již poté sám postará o vytvoření odpovídající tabulky v databázi. Jednotlivé sloupce tabulky pak odpovídají jednotlivým atributům použité třídy.

Metody ve třídě `LawGiverRepository` se poté nedotazují pomocí klasických SQL dotazů, ale využívají LINQ výrazy. LINQ výrazy můžeme využít na dotazování nad libovolným zdrojem, který implementuje rozhraní `IEnumerable<T>`. Obecně máme dvě možnosti jak LINQ výrazy zapsat. První variantou jsou lambda výrazy, druhá možnost je pomocí metod. Pomocí lambda výrazů je zápis podobný klasickému SQL dotazu.

V kódu níže je uvedena ukázka pro základní operace získávání, úpravy, vkládání a mazání dat pomocí metod.

```

//ziskani kolekce dat
public IQueryable<Client> GetAllClients() {
    return db.Clients.OrderBy(g => g.Surname);
}

//uprava udaju
public void UpdateFile(File file) {
    db.Entry(file).State = EntityState.Modified;
}

```

```
//ulozeni udaju
public void AddFile(File file) {
    db.Files.Add(file);
}

//uprava udaju
public void UpdateFile(File file) {
    db.Entry(file).State = EntityState.Modified;
}
```

Při získávání dat je možné využít metody `Include()` a zahrnout tak do výsledku i odpovídající (přes cizí klíč) data z dalších tabulek:

```
//ziskani kolekce dat z vice tabulek
public IQueryable<File> GetAllFiles() {
    return db.Files.Include(a => a.Client).Where(a => a.Active ==
true).OrderByDescending(g => g.Created);
}
```

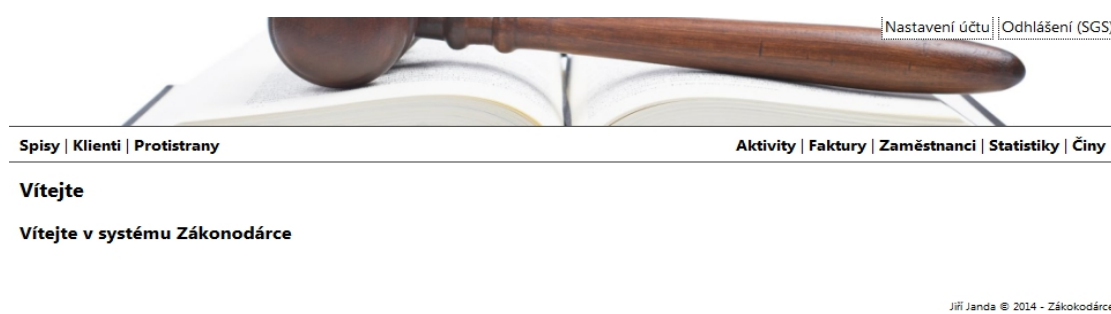
Po úpravě dat (všechny operace kromě získávání) je vždy nutné potvrdit modifikaci databáze pomocí metody `SaveChanges()` objektu `LawGiverDataContext`.

8.6 Rozložení stránky

Každá stránka aplikace je rozložena do tří hlavních oblastí. První oblastí je záhlaví stránky, které má určenou pevnou výšku a šířku (v základním zobrazení) a obsahuje odkazy pro práci s vlastním účtem a odhlášení. Záhlaví je zobrazeno na každé stránce ve stejné podobě. V případě sledování informací z eJustice jsou zde zobrazovány upozornění na nově stažená data.

Druhou částí, která je také zobrazena na všech podstránkách je navigace. Ta je rozdělena na dvě části. Vlevo je umístěna navigace, která je zobrazena všem uživatelům a umožňuje přístup k přiřazeným spisům, uživatelům a protistranám. V pravé části je navigace, která je přístupná pouze uživatelům s oprávněním Manažer. Z této navigace je možné se dostat k funkcím pro správu zaměstnanců, aktivit, faktur či přehledu statistik a činů.

Poslední částí stránky, která je umístěna pod navigací, je samotná obsahová část. Obsah tohoto prostoru se vždy liší dle zobrazené stránky. Náhled na úvodní stranu po přihlášení do systému je zobrazen na obrázku 8.3.



Obrázek č. 8.3 – Rozložení stránky

8.7 Responzivní zobrazení

Řešená aplikace je vytvořena s responzivním designem. Její vzhled a rozložení se tedy (částečně) mění v závislosti na aktuální šířce průzoru prohlížeče. Rozhodl jsem se využít dvě možné varianty zobrazení.

První variantou je zobrazení na běžném počítačovém monitoru. Jako běžný monitor je v tomto případě brán monitor s rozlišením alespoň 1024 pixelů do šířky, což je nejmenší běžně užívané rozlišení v současnosti¹⁰. Pro tuto variantu je využito rozložení stránky s pevnou šířkou 960 pixelů, což je šířka, která se vždy vejde do průzoru (maximalizovaného okna) bez nutnosti horizontálního skrolování a to i v případě, že jsou v prohlížeči zobrazeny vertikální posuvníky.

Druhou variantou je zobrazení určené pro tablety a chytré mobilní telefony. Toto rozložení se využije při šířce průzoru menší než 960 pixelů. Odlišností od první verze je šířka stránky, která nemá pevně definovanou velikost, ale je nastavena na 100%. Zaplněn je tedy vždy celý průzor prohlížeče, ať je jakékoliv šířky. Další odlišností je změna navigace, kde v první verzi jsou základní i manažerské menu v jedné rovině. V této mobilní verzi však není možné, aby se obě navigace vešly do užšího rozložení a jsou proto zobrazeny pod sebe. Tyto dvě možnosti zobrazení by měly pokrýt všechny typy běžně využívaných zařízení, na kterých se dá předpokládat použití této aplikace.

8.8 Autentizace a autorizace uživatelů

Již automaticky vytvořená demo aplikace, kterou vytvoří MS Visual Studio při založení projektu, umožňuje klasickou autentizaci pomocí webového formuláře. O přihlašování se stará `AccountController.cs`, který pracuje s lokálně uloženými loginy a hesly.

V aplikaci můžeme k informacím o přihlášeném uživateli přistupovat pomocí objektu `User`, nejdůležitější je vlastnost `User.Identity.Name`, což je unikátní identifikátor účtu v rámci celé aplikace. Tyto data jsou uložena v tabulce `UserProfile`, kterou ovšem nelze měnit. Není tak možné k uživateli navázat další dodatečné informace, které můžeme potřebovat uložit. V ASP.NET se tento problém řeší vytvořením další tabulky, ve které jsou uloženy všechny další informace vázané k účtu (e-mail, jméno atd). Tato tabulka obsahuje unikátní sloupec `UserName`, který je zároveň cizím klíčem do `UserProfile`, tabulky jsou k sobě tedy napojeny vztahem 1:1.

Na straně uživatele je přihlášení udržováno klasickým způsobem, tedy pomocí `sessions`, kdy má klient přidělen unikátní `session id` řetězec. Ten je nejčastěji uložen v cookies, ale je možné ho zasílat v každém požadavku jako GET parametr. V obou případech se tento identifikátor při každém požadavku odesílá na server a jednoznačně identifikuje uživatele.

Autorizace je v aplikaci řešená pomocí atributů (ze třídy `System.Web.Mvc`), které se uvádí před metodou či celou třídu controlleru. Pokud je před metodou uveden atribut `[Authorize]`, je možné metodu spustit pouze jako přihlášený uživatel. Pokud je požadavek veden od nepřihlášeného uživatele, je přesměrován na stránku, která je zadána v konfiguraci. Když atribut uvedeme před celou třídu, je platný pro všechny její metody. Pokud bychom chtěli pro některou z metod výjimku tak, aby mohla být využita i nepřihlášeným uživatelem, je to možné pomocí atributu `[AllowAnonymous]`.

V systému jsou rozlišovány dvě uživatelské role, běžný Pracovník a Manažer. Některé části a funkce systému jsou dostupné pouze roli Manažer. Omezení přístupu se řeší opět pomocí atributů. Nejdříve je nutné aktivovat ověřování rolí pomocí atributu `[InitializeSimpleMembership]`, a pak je již možné omezovat přístup pomocí `[Authorize(Roles="Manager")]`. Tento postup je opět možné využít jak pro celou třídu, tak i její jednotlivé metody.

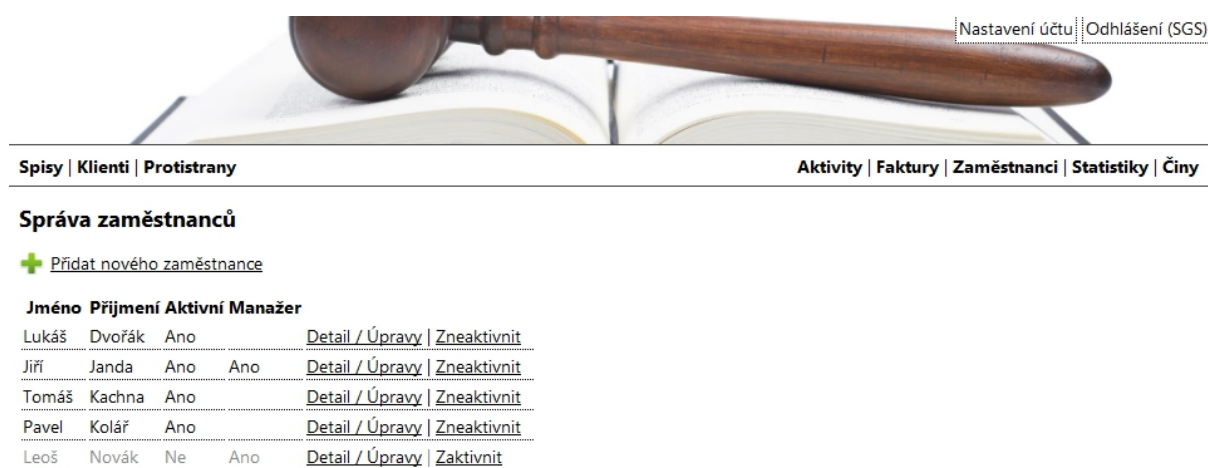
10 <http://www.toplist.cz/global.html> – globální statistika společnosti TOPlist

Možné je filtrovat přístup ke třídám či metodám i pro konkrétní uživatelské účty pomocí atributu [Authorize(Users = "Uživatelské jméno")], tento postup ale není nikde v této aplikaci využit.

8.9 Správa zaměstnanců

Využívat sekci správa zaměstnanců je umožněno všem uživatelům s oprávněním Manažer. V této sekci je možné vytvářet nové zaměstnanecké účty a upravovat údaje u účtů již existujících. V základním tabulkovém přehledu jsou uvedeny informace o všech uložených zaměstnancích, aktuálních i bývalých a to včetně jejich oprávnění.

Účet zaměstnance není možné zcela smazat, ale pouze zneaktivnit (zneaktivněný účet lze pak opět aktivovat). Uživatel s neaktivním účtem zůstává uložen v systému, ale není mu umožněno se přihlásit a tedy vykonávat jakékoliv akce. Důvodem k tomuto přístupu je konzistence uložených dat. Pokud bychom umožnili smazání celého zaměstnaneckého účtu, ke kterému jsou navázány další data z jiných tabulek (aktivity na spisech, činy atd.), vznikl by problém s nekonzistencí dat. Bylo by samozřejmě možné mazat i příslušná data v dalších tabulkách, ale v takovém případě by poté mizeli například údaje z vystavených faktur, což opět není vhodný výsledek. Z tohoto důvodu jsem řešil situaci možností pouze deaktivovat účet. Základní obrazovka pro správu zaměstnanců je zobrazena na obrázku 8.4.



Jiří Janda © 2014 - Zákodárce

Obrázek č. 8.4 – Správa zaměstnanců

8.10 Správa spisů

Sekce správa spisů je přístupná všem zaměstnancům, její obsah se ale liší dle oprávnění uživatele.

Běžný zaměstnanecký účet má možnost vytvářet libovolně nové spisy. Dále je zde umístěn hlavní přehled přiřazených spisů. Přiřadit spisy může uživateli manažer, a také jsou mu automaticky přiřazeny spisy jím samotným vytvořené. V tomto hlavním přehledu jsou umístěny všechny nejdůležitější informace a také akce s každým spisem. Uvedena zde je interní spisová značka, klient i protistrana. Při najetí na ikonu informací jsou zobrazeny i další uložené informace o spisu (datum založení, popis). V další části tabulky jsou uvedeny odkazy na všechny přiřazené položky ke spisu – pracovníky, dokumenty, poštu a zadávání aktivit. Obsahem dalších sloupců jsou akce

s celým spisem – editace údajů, jeho smazání a jeho uzavření (při skončení kauzy). Smazání zde, stejně jako v případě zaměstnanců není úplné, ale opět jde pouze o deaktivaci spisu. Důvod tohoto řešení je stejný jako v případě zaměstnanců. Poslední částí přehledu je sekce eJustice, v této části je možné spravovat přiložené soudní značky a díky tomu načítat data ze systémů eJustice. Na přehled těchto dat je přístup také z této části.

Účet s oprávněním Manažer má tuto sekci rozšířenou o filtrování přehledů spisů. V základním zobrazení má vypsány stejně jako zaměstnanec pouze přiřazené spisy ke svému účtu. Lze ale přepnout na výpis všech aktivních spisů uložených v systému a také na přehled všech již smazaných spisů. Náhled na sekci správa spisů je na obrázku 8.5.

Nastavení účtu | Odhlášení (SGS)

Spisy | Klienti | Protistrany Aktivity | Faktury | Zaměstnanci | Statistiky | Činy

Správa spisů Filtr: Přiřazené | Všechny | Smazané

+ Založit nový spis

Značka	Klient	Protistrana	Datum	Popis	Akce	eJustice
2020-16	Pavel Třetí			Založeno: 15. 02. 2014 Popis: Rozvod manželů, započeti 1.2.2014	Aktivita	Značky Jednání Řízení
2014-1	Petr Julíček	Argonet s.r.o.		Pracovníci Dokumenty Pošta	Aktivita	Značky Jednání Řízení
2014-2	Lucie Sedmihorská	Argonet s.r.o.		Pracovníci Dokumenty Pošta	Aktivita	Značky Jednání Řízení
2014-3	Petr Julíček			Pracovníci Dokumenty Pošta	Aktivita	Značky Jednání Řízení

Jiří Janda © 2014 - Zákodárce

Obrázek č. 8.5 – Správa spisů

8.11 Získávání dat

8.11.1 Systém ARES

Data ze systému ARES jsou získávána pomocí exportu dat, který tento systém nabízí. Lze využít několik druhů exportů, kde můžeme získávat informace z jednotlivých registrů a databází, které ARES sdružuje. Jako nejvhodnější mi ale přišla možnost využít základní výpis z více registrů „Basic“, který kombinuje ty nejdůležitější data. K adrese exportu je možné připojit několik parametrů, které upraví vrácená data (formát výstupu, jazyk, pouze aktivní subjekty). Jako unikátní identifikátor osoby se zde používá IČ. Výsledná adresa využitého exportu dat je:

`http://wwinfo.mfcr.cz/cgi-bin/ares/darv_bas.cgi?ico=cislo_ico&xml=0`

Z tohoto zdroje získáme základní údaje (název, datum založení, adresu, počet zaměstnanců), stav subjektu v registrech, předměty podnikání anebo obory činnosti v RŽP.

Stažená data ze zdroje jsou zpracovávána kontinuálním přístupem (SAX parser) k XML, kdy se postupně prochází celý dokument a když narazíme na hledaný element, uložíme jeho hodnotu.

8.11.2 Systémy eJustice

Systém eJustice (InfoSoud, InfoJednani) neposkytují žádný veřejný export dat. Jediná možnost získání informací je sestavení konkrétní URL a poté parsování získaných výsledků.

Pro sestavení URL kterou budeme volat musíme znát tyto údaje:

- Krajský soud, který případ řeší
- Okresní soud, který případ řeší
- Spisová značka (číslo senátu, druh věci, běžné číslo, ročník) případu

Jestliže jsou u spisu zadány uživatelem všechny tyto povinné údaje, je možné získat informace eJustice. Výsledná volaná adresa má pak tento tvar (uvedena je varianta pro InfoJednani, pro InfoSoud je obdobná):

```
http://infojednani.justice.cz/InfoSoud/public/searchJednani.do?
type=spzn&typSoudu=os&krajOrg=krajsy_soud&org=okresni_soud&cisloSena
tu=cislo_senatu&druhVec=druh_veci&bcVec=bezne_cislo&rocnik=rocnik&ag
endaNc=CIVIL&sin=&datum=&spamQuestion=23
```

Každý krajský a okresní soud má svoji zkratku, ty jsou uloženy jako kolekce v pomocné třídě `Helpers/Courts.cs`. Parametry `type`, `typSoudu` a `agendaNc` mají neměnnou hodnotu. Parametry `sin` a `datum` není nutné v tomto typu dotazu uvádět. Parametr `spamQuestion` slouží k ověření platnosti dotazu (jako ochrana proti automatickým scriptům), v době tvorby této aplikace byla jeho hodnota neměnná a to 23. Při případné změně této hodnoty by načítání z eJustice přestalo fungovat správně (nevracelo by žádné výsledky) a byla by nutná úprava tohoto parametru ve zdrojovém kódu aplikace.

K samotnému parsování získaného HTML kódu pak slouží třída `Helpers/EJusticeParser.cs`, která má metody na parsování zdrojového kódu z `InfoJednani` i `InfoSoudu`. K samotnému získání dat ze staženého HTML kódu je využívána knihovna `fizzler`¹¹. Tato knihovna umožňuje získávání dat z HTML dokumentu pomocí CSS selektorů, tak jak jsou známy z JavaScriptového frameworku `jQuery`¹². Níže je uvedena ukázka kódu pro získání všech elementů se třídou `content`.

```
var html = new HtmlDocument();
html.LoadHtml(html_string);
var document = html.DocumentNode;

var contents = document.QuerySelectorAll(".content");
```

8.12 Generování faktur

Fakturu je možné si v aplikaci zobrazit dvojnásobem. První možností je běžná HTML stránka, vytvořená klasicky pomocí `View` a předaných dat. Druhá možnost je zobrazení faktury jako vygenerovaného PDF. To je vhodnější varianta, pokud by uživatel takovou fakturu například zasilal klientovi pomocí e-mailu. Pro generování PDF je využita knihovna `iText`¹³ a šablony `RazorPDF`. V samotném `controlleru` probíhá zpracování dat obvyklým způsobem. Do `View` se předávají data typu objekt `PdfResult`. Ve `View` je poté místo klasického HTML využito `iTextXML`. Jde o značkovací jazyk, který poskytuje značky potřebné pro běžné formátování (odstavce, tabulky, obrázky atd.) dokumentu, na které je možné využívat CSS a tím měnit jejich vzhled. Ukázka `iTextXML` kódu je uvedena níže:

11 <https://code.google.com/p/fizzler> - domovské stránky projektu

12 <http://jquery.com> - domovské stránky projektu

13 <http://itextpdf.com> - domovské stránky projektu

```

<paragraph style="font-family:Helvetica;font-size:18;font-
weight:bold;font-style:italic;">
  <chunk red="0" green="0" blue="0">Faktura c.
  @Model.Invoice.InvoiceId</chunk>
</paragraph>

<table width="100%" cellpadding="1.0" cellspacing="1.0">
  <row>
    <cell borderwidth="0" left="false" right="false" top="false">
      <chunk style="font-weight:bold;">Dodavatel</chunk>
    </cell>

    <cell borderwidth="0" left="false" right="false" top="false">
      <chunk style="font-weight:bold;">Odběratel</chunk>
    </cell>
  </row>
</table>

```

Samotná faktura se skládá z několika částí. Každá faktura začíná svým evidenčním číslem, které slouží také jako variabilní symbol pro platbu účtem. Dále je uvedeno jméno a další údaje o dodavateli a odběrateli. Informace o odběrateli jsou načteny z uložených údajů o klientovi. Údaje o dodavateli, tedy advokátní kanceláři, která aplikaci využívá, jsou uloženy v textovém konfiguračním souboru umístěném v adresáři `App_Data/Invoice` v souboru `MyInfo.html`, který je možné libovolně editovat. Další částí faktury je způsob platby převodem z účtu na účet. Číslo účtu je možné opět libovolně nastavit v souboru `AccountNumber.html` ve stejném adresáři. Datum vystavení je shodné s datem vygenerování faktury a splatnost je nastavena na 1 měsíc od vystavení. Poslední částí faktury je výpis jednotlivých služeb, které jsou klientovi účtovány. U každé aktivity je uvedena její cena a poté celková cena k úhradě. Vygenerovaná faktura ve formátu PDF je zobrazena na obrázku 8.6.

Faktura c. 3

Dodavatel

Advokátní kancelář
Ulice 22 613 00 Brno
IC: 45615165

Odberatel

Petr Julíček
Hromova 22
215 00 Brno

Způsob platby

Prevodem
Číslo účtu: 15313213
Variabilní symbol: 3

Datum vystavení: 09. 03. 2014
Datum splatnosti: 09. 04. 2014

Fakturované služby:

Název	Cena
Kolek	66 Kč
Žádost u soudu	30 Kč
Celkem: 96 Kč	

Podpis dodavatele:

Obrázek č. 8.6 – Vygenerovaná faktura ve formátu PDF

8.13 Oznamování nových událostí

Data ze systému eJustice jsou ze serverů stahována automaticky na pozadí. Proto je nutné upozornit uživatele, pokud byly získány nové informace. Upozorňování jsem se rozhodl řešit pomocí technologie AJAX (Asynchronous JavaScript and XML), která umožňuje pomocí JavaScriptu na pozadí komunikovat se serverem, a to bez nutnosti opětovného načítání celého dokumentu.

Při každém načtení stránky je pomocí události `onload` spuštěna funkce, která aktivuje opakované zaslání požadavků na zjištění nových informací. Tyto požadavky na server se na pozadí odesílají každých 10 sekund. Serverová část funguje tak, že načte všechny přiřazené spisy k uživateli. U každého spisu zkontroluje pro všechny jeho uložené soudní spisové značky stažená data ze systému InfoSoud a InfoJednání. U každé stažené informace je uloženo datum stažení, taktéž má každý uživatel uloženo datum, kdy si naposledy zobrazil přehled informací z InfoSoudu a InfoJednání u každého svého přiřazeného spisu. Díky porovnání těchto dvou časových informací můžeme získat data, která byla nově stažena od posledního zobrazení uživatelem. Tyto informace jsou ve formátu XML opět zaslány uživateli. U uživatele jsou pomocí JavaScriptu tato data rozparsována, a pokud vrácená data nejsou prázdná, je uživateli zobrazeno upozornění v levé části stránky. Ukázka upozornění je zobrazena na obrázku 8.7. Vedlejší efektem tohoto řešení je také to, že uživatel není nikdy ze systému odhlášen. Při každém automaticky odeslaném požadavku na pozadí je prodloužena platnost jeho session.



Obrázek č. 8.7 – Upozornění na nové informace

Při rozkliknutí tohoto upozornění získá uživatel zjednodušený přehled o všech nových datech. Je zde vypsána značka daného spisu, datum jednání v případě informací z InfoJednání či název události v případě InfoSoudu. Rozkliknuté upozornění je uvedeno na obrázku 8.8.

Nové jednání u spisu 2020-16 (19.02.2014, 08:30)	
Nová událost u spisu 2020-16 (Nařízení jednání)	
Nová událost u spisu 2020-16 (Odeslání spisu)	
Nová událost u spisu 2020-16 (Podán opravný prostředek)	
Nová událost u spisu 2020-16 (Nařízení jednání)	
Nová událost u spisu 2020-16 (Zahájení řízení)	
2014-1 Petr Juříček	Argonet s.r.o.
	Protistrana Details Přidruže
	Pracovníci Dokumenty
	Pracovníci Dokumenty

Obrázek č. 8.8 – Otevřené upozornění na nové informace

Každé jednotlivé upozornění je aktivní odkaz, který vede na stránku s přehledem všech informací z InfoJednání nebo Infosoudu k danému spisu. Při zobrazení této stránky se aktualizuje uložené datum posledního zobrazení dat, a díky tomu na tyto informace již podruhé není upozorňováno. Nové, poprvé zobrazené informace jsou v přehledu barevně odlišeny od starších.

9 Testování

Jako každou aplikaci, bylo nutné i zde otestovat správnost fungování řešeného projektu. Testování, které proběhlo, lze obecně rozdělit na tři typy, každému je níže věnována samostatná podkapitola. V průběhu testování bylo objeveno množství chyb různé závažnosti. Všechny tyto nalezené chyby byly odstraněny.

9.1 Testování zobrazení

U webových aplikací vzniká problém různého zobrazení dle použitého prohlížeče, ve kterém je stránka zobrazena. Tento problém vzniká z důvodu různé interpretace HTML, CSS a JavaScriptu v každém z prohlížečů. V současné době není tento problém tak výrazný jako ještě před několika lety, kdy každý z hlavních prohlížečů využíval své vlastní technologie a webové standardy z velké části ignoroval. V této době bylo často nutné vytvářet samostatné CSS šablony a různé verze JavaScriptových funkcí pro každý prohlížeč zvlášť. V současnosti je již situace výrazně lepší a prohlížeče poměrně dobře dodržují standardy stanovené organizací W3C¹⁴. I nyní je ale nutné vždy otestovat správnost zobrazení v různých prohlížečích. Není možné otestovat všechny prohlížeče, z důvodu jejich velkého počtu a různých verzí. Aplikaci jsem otestoval ve třech v současnosti nejdůležitějších prohlížečích¹⁵, na prohlížeči Google Chrome (zobrazovací jádro WebKit) ve verzi 33, Mozilla Firefox (jádro Gecko) ve verzi 27 a Windows Internet Explorer (jádro Trident) ve verzi 9. Tyto tři programy dohromady pokrývají nad 90% používaných prohlížečů. Zbylá procenta jsou menší prohlížeče, které jsou ale z většiny postavené na jednom z výše uvedených vykreslovacích jader, dá se tedy předpokládat stejné zobrazení. Aplikace byla dále otestována na mobilním telefonu a operačním systémem Android.

Při testování bylo odstraněno několi chyb zobrazení a ve finální verzi je zobrazení aplikace ve všech testovaných prohlížečích funkční a stejné. V aplikaci jsou ale využívány některé novější technologie (např. CSS 3 pro responzivní zobrazení), které nemají podporu u starších verzí prohlížečů. V takovém případě se dají očekávat určité chybné zobrazení.

9.2 Testování funkcionality

Kromě správného zobrazení v prohlížečích, tedy ověření zda-li se zobrazují data správně, je také nutné ověřit zda-li se zobrazují správná data. Jako nejlepší způsob testování samotného fungování aplikace vidím otestování běžnými uživateli. Je tak možné odhalit nejen chyby, ale získat i další informace o použitelnosti a fungování aplikace. Aplikace byla otestována na dalších uživateli, kteří nahlásili několik chyb a poskytli zpětnou odezvu. Díky tomuto způsobu testování bylo nalezeno několik menších chyb. Šlo zejména o chybné výpisy textu, či nesprávné fungování JavaScriptu.

14 <http://www.w3.org/> - domovské stránky organizace W3C

15 <http://www.toplist.cz/global.html> – přehled zastoupení prohlížečů na Toplist.cz

9.3 Testování výpočtů a získávání dat

Pro ověření správnosti stahovaných a vypočítaných dat jsem využil možnost automatizovaného testování. Po prvotním napsání samotného testu je již samotné testování velmi efektivní. Po každé úpravě testovaného kódu můžeme vždy spustit stejný test a pouze ověřit jeho výsledek, dokud test neproběhne v pořádku.

Podpora těchto Unit testů, které ASP.NET MVC nabízí, nám umožňuje snadno testovat výsledky, které vrací jednotlivé metody controllerů. Testování tímto způsobem jsem využil k ověření správnosti stahovaných dat ze serverů eJustice a dále také k ověření vypočtených částek účtovaných na vystavených fakturách.

10 Závěr

Tato závěrečná kapitola práce obsahuje shrnutí dosažených výsledků. V první části jsou popsány všechny funkce, které se podařilo implementovat. V druhé části této kapitoly jsou uvedeny možnosti dalšího rozšíření řešené aplikace.

10.1 Dosažené výsledky

V úvodu této práce byly popsány základní procesy, které fungují v rámci běžné agendy v advokátních kancelářích. Na jejich základě byly popsány požadavky na software, který by usnadňoval a zjednodušoval práci zaměstnanců advokátní kanceláře. Zjednodušení se týká pohodlnější evidence klientů, spisů, příložených dokumentů a korespondence. Nejvýraznější usnadnění pak poskytuje spojení s veřejnými databázemi. Díky tomuto napojení není nutné vyhledávat ručně informace na několika různých portálech, ale vše je snadno dostupné v rámci jedné aplikace.

V práci byly dále popsány již existující softwarové alternativy vhodné pro tento účel. Na základě jejich analýzy a porovnání funkcionality, kterou nabízí, vyplynuly další vhodné funkce pro navrhovaný software. Díky získaným požadavkům je v další části této technické zprávy vytvořen návrh. Dále jsou vybrány vhodné technologie k implementaci aplikace, tak aby mohla splňovat všechny požadavky.

V druhé, praktické části této práce, byl implementován systém, který pokrývá všechny požadavky, které stanovila připravená specifikace systému. V průběhu tvorby systému se objevily některé nové možnosti rozšíření funkcionality, které byly implementovány a zpětně byl o tyto možnosti doplněn návrh i specifikace. Aplikace je vytvořena v technologii ASP.NET, podle obecně používaného návrhového vzoru MVC. K této technologii i přístupu implementace existuje velké množství dokumentace a studijních materiálů, což je výhodou pro případné pozdější úpravy a rozšíření jinými programátory.

Po vytvoření byla aplikace důkladně otestována na několika úrovních. Díky nalezení několika menších problémů a jejich následném odstranění je nyní aplikace funkční na zařízení s libovolným operačním systémem, které je připojeném k internetu (včetně chytrých mobilních telefonů a tabletů). Testování probíhalo na reálných datech a existujících osobách a případech uložených ve veřejných databázích.

Dle navržené a implementované funkcionality můžeme tuto aplikaci porovnat s jinými existujícími řešeními, které jsou popsány v kapitole 3. Z kritérií, které jsem použil pro srovnání těchto jiných řešení, splňuje zde řešená aplikace tyto funkce: správa spisů, napojení na eJustice, napojení na Insolvenční rejstřík, správa pošty a správa účetnictví. Jako hlavní konkurenční výhodou této aplikace vidím napojení na velké množství veřejných databází. Díky tomu je možné snadno získat značné množství informací. Jako další výhody, které nejsou zmíněny u žádného z dalších produktů, považuji zejména přehled veškerých vykonaných akcí v systému (výhodné pro zpětném dohledání například při problému se smazanými daty) a také přehledné statistiky.

10.2 Možnosti rozšíření

V porovnání s existujícími komerčními aplikacemi, řešícími obdobné téma, je řešená aplikace výčtem funkcí srovnatelná. Z jiných řešení lze ale určit směry, kterými by se mohla aplikace dále rozvíjet.

Možné je vylepšovat zejména další způsoby komunikace. Možnosti jsou například napojení na systém datových schránek, nebo spolupráci s e-mailovým účtem (uloženým na e-mailovém serveru, nebo lokálně s e-mailovým klientem).

Rozšířit je možné také oprávnění uživatelů. První možností jak tuto funkcionalitu pojmout je zavedení více uživatelských rolí (například koncipient, účetní). Oprávnění se dá také řešit způsobem vytvoření kompletního systému jednotlivých práv a každému uživateli přiřadit libovolná práva.

Další možností rozšíření aplikace je posílení manažerského účtu, například o vytváření a termínování úkolů, které je možné přiřazovat jednotlivým zaměstnancům.

Výraznějším rozšířením by mohlo být vytvoření podpůrné desktopové aplikace. Při spuštění aplikace by bylo nutné zadat uživatelské jméno a heslo pro přihlášení do systému. Poté by aplikace již běžela pouze na pozadí a zaslala dotazy na zde řešenou webovou aplikaci. Součástí rozšíření by bylo vytvoření služeb, které by poskytovaly informace pro desktopovou aplikaci. Šlo by zejména o oznamování nových událostí, které by se potom zobrazovaly uživateli. Ten by tak byl ihned informován o všech nových informacích i bez nutnosti mít zapnutý prohlížeč a být přihlášen ve webové aplikaci.

Literatura

- [1] HOLUBEC, P.: INFOSOUD - Vyhledávání informací o průběhu řízení [online]. [cit. 2014-05-8]. Dostupné z: <http://krpela.webnode.cz/products/produkt-1/>
- [2] Kardex Group: Software pro správu spisů [online]. [cit. 2014-05-08]. Dostupné z: <http://www.kardex-remstar.cz/cz/produkty/software-pro-spravu-spisu.html>
- [3] Acta Safe | Systém pro vedení kauzy a spisu [online]. [cit. 2014-05-08]. Dostupné z: <http://actasafe.cz/>
- [4] Atlas Consulting: Advokátní spis [online]. [cit. 2014-05-08]. Dostupné z: <http://www.atlascon.cz/software/advokatni-spis>
- [5] SMITKA, D. Smitka software [online]. [cit. 2014-05-08]. Dostupné z: <http://www.jurisdix.cz/jurisdix.htm>
- [6] Správa zakázek, spisů a pohledávek [online]. [cit. 2014-05-08]. Dostupné z: <http://www.synopsis.cz/>
- [7] KLEOS – evidence a fakturace kauz [online]. [cit. 2014-05-08]. Dostupné z: <http://www.kleos.cz/>
- [8] Ministerstvo spravedlnosti ČR: Informace o řízení [online]. Dostupné z: <http://infosoud.justice.cz/InfoSoud/public/search.jsp>, 2008 [cit. 2014-05-01].
- [9] Ministerstvo spravedlnosti ČR: Informace o jednání [online]. [cit. 2014-05-01]. Dostupné z: <http://infosoud.justice.cz/InfoSoud/public/searchJednani.jsp>
- [10] Ministerstvo financí ČR: *ARES* – Administrativní registr ekonomických subjektů [online]. [cit. 2014-5-1]. Dostupné z: <http://infosoud.justice.cz/InfoSoud/public/searchJednani.jsp>
- [11] Ministerstvo spravedlnosti ČR: Insolvenční rejstřík [online]. [cit. 2014-05-01]. Dostupné z: <https://isir.justice.cz/isir/common/index.do>
- [12] FREEMAN, Adam a Steven SANDERSON. *Pro ASP.NET MVC 3 framework*. 3rd. ed. New York: Apress, c2011, xxv, 824 s. ISBN 978-1-4302-3404-3.
- [13] SCHMULLER, Joseph. *Myslíme v jazyku UML*. Vyd. 1. Praha: Grada, 2001, 360 s. ISBN 80-247-0029-8.
- [14] SOMMERVILLE, Ian a Jakub GONER. *Softwarové inženýrství*. 1. vyd. Brno: Computer Press, 2013, 680 s. ISBN 9788025138267.
- [15] MVC : The Official Microsoft ASP.NET Site [online]. [cit. 2014-05-04]. Dostupné z: <http://www.asp.net/mvc>
- [16] Entity-relationship model [online]. [cit. 2014-05-05]. Dostupné z: http://cs.wikipedia.org/wiki/Entity-relationship_model
- [17] Diagram užití[online]. [cit. 2014-05-05]. Dostupné z: http://cs.wikipedia.org/wiki/Diagram_u%C5%BEit%C3%AD

Seznam příloh

Příloha A. Instalace aplikace

Příloha B. Ukázky uživatelského rozhraní

Příloha C. CD se zdrojovými soubory

Příloha A - Instalace aplikace

Veškeré zdrojové soubory projektu (adresář LawGiver) je nutné umístit do určeného adresáře, který je využíván serverem IIS. V souboru `Web.config`, který slouží pro konfiguraci, je potom nutné nastavit atribut `connectionString`, který slouží k připojení k databázi. Soubor s databází je umístěn v adresáři `App_Data`.

V tuto chvíli je aplikace funkční, ale je nutné vytvořit výchozí administrátorský účet. Ten lze vytvořit zavoláním adresy `domena/Account/Installation`. Po spuštění této adresy se v systému vytvoří (pokud již neexistuje) účet „admin“ s přístupovým heslem „pass“ a oprávněním Manažer. S pomocí těchto údajů je možné přihlásit se do systému pomocí formuláře na úvodní straně.

Pro správnou funkčnost stahování dat na pozadí je také nutné nastavit plánovač úloh na adresu `domena/InfoSoud/Download`.

Příloha B – Ukázky uživatelského rozhraní

Nová data z eJustice.cz (1) Nastavení účtu | Odhlášení (SGS)

Spisy | Klienti | Protistrany Aktivity | Faktury | Zaměstnanci | Statistiky | Činy

Správa klientů

+ Přidat nového klienta

Jméno	Příjmení	Adresa	Telefon	E-mail	Informace	Akce
Petr	Bednář	Úzká 10 61300 Brno		bednar@gmail.com		Upravit Smazat
Petr	Chvátíl			chvatil@seznam.cz		Upravit Smazat
Petr	Julíček	Hromova 22 215 00 Brno		SGSa@alpaka.cz	ARES Insolvenční r.	Upravit Smazat
Bára	Malá	U hluchých 54877 Praha		mala@seznam.cz		Upravit Smazat
Lucie	Sedmihorská			testi@alpaka.cz		Upravit Smazat
Pavel	Třetí			SGSa@alpaka.cz		Upravit Smazat

Jiří Janda © 2014 - Zákodárce

Obrázek č. B.1 – Přehled uložených klientů

Nová data z eJustice.cz (1) Nastavení účtu | Odhlášení (SGS)

Spisy | Klienti | Protistrany Aktivity | Faktury | Zaměstnanci | Statistiky | Činy

Správa spisů

+ Založit nový spis

Filtr: [Přirazené](#) | [Všechny otevřené](#) | [Uzavřené](#) | [Smazané](#)

Značka	Klient	Protistrana	Details	Přidružené položky	Akce	eJustice
2020-16	Pavel Třetí		i	Pracovníci Dokumenty Pošta Aktivita		Značky Jednání Řízení
2014-1	Petr Julíček	Argonet s.r.o.	i	Pracovníci Dokumenty Pošta Aktivita		Značky Jednání Řízení
2014-2	Lucie Sedmihorská	Argonet s.r.o.	i	Pracovníci Dokumenty Pošta Aktivita		Značky Jednání Řízení
2014-3	Petr Julíček		i	Pracovníci Dokumenty Pošta Aktivita		Značky Jednání Řízení
2014-19	Petr Bednář	Argonet s.r.o.	i	Pracovníci Dokumenty Pošta Aktivita		Značky Jednání Řízení
2014-48	Petr Chvátíl		i	Pracovníci Dokumenty Pošta Aktivita		Značky Jednání Řízení
2014-51	Petr Chvátíl	Česká pošta	i	Pracovníci Dokumenty Pošta Aktivita		Značky Jednání Řízení
2020-17	Bára Malá	Česká pošta	i	Pracovníci Dokumenty Pošta Aktivita		Značky Jednání Řízení

Jiří Janda © 2014 - Zákodárce

Obrázek č. B.2 – Přehled vedených spisů

Nastavení účtu | Odhlášení (SGS)

Spisy | Klienti | Protistrany Aktivity | Faktury | Zaměstnanci | Statistiky | Činy

Nastavení účtu

Změnit heslo

1. Současné heslo
2. Nové heslo
3. Ověření nového hesla

Změnit osobní údaje

Jméno

Příjmení

E-mail

Obrázek č. B.3 – Nastavení účtu

Nastavení účtu | Odhlášení (SGS)

Spisy | Klienti | Protistrany Aktivity | Faktury | Zaměstnanci | Statistiky | Činy

Přehled činů v systému

Zobrazit činy pouze od:


Datum a čas	Zaměstnanec	Text
14.5.2014 20:43:28	Jiří Janda	Vytvořil spis: 2020-17
14.5.2014 20:43:19	Jiří Janda	Vytvořil spis: 2014-51
14.5.2014 20:43:09	Jiří Janda	Přidal novou protistranu: Česká pošta
14.5.2014 20:42:49	Jiří Janda	Vytvořil spis: 2014-48
14.5.2014 20:42:38	Jiří Janda	Vytvořil spis: 2014-19
14.5.2014 20:40:52	Jiří Janda	Přidal nového klienta: Bára Malá
14.5.2014 20:40:25	Jiří Janda	Přidal nového klienta: Petr Bednář
14.5.2014 20:39:58	Jiří Janda	Přidal nového klienta: Petr Chvátíl
12.4.2014 10:10:21	Jiří Janda	Přidal nového zaměstnance: petr pavel
9.3.2014 10:22:59	Jiří Janda	Upravil údaje klienta: Petr Julicek
9.3.2014 10:22:47	Jiří Janda	Upravil údaje klienta: Petr Julicek
8.3.2014 13:11:04	Jiří Janda	Upravil údaje zaměstnance: Pavel Kolář
8.3.2014 13:06:39	Jiří Janda	Upravil údaje zaměstnance: Tomáš Kachna
8.3.2014 13:06:35	Jiří Janda	Upravil údaje zaměstnance: Tomáš Kachna
8.3.2014 13:06:29	Jiří Janda	Upravil údaje zaměstnance: Tomáš Kachna

Obrázek č. B.4 – Přehled aktivit v systému


Informace o událostech v řízení ke spisu 2020-16

Název	Datum	Zobrazit originál
Datum pravomocného ukončení věci ▼	18.04.2014	Zobrazit
Nañřízení jednání ▼	16.04.2014	Zobrazit
<p>Bylo nařizeno jednání ve věci Druh jednání: Hlavní líčení s dokazováním Jednací síň: č. 01 I. podlaží Začátek jednání: 16.04.2014 13:00 Datum zápisu výsledku: 17.04.2014 Výsledek: Vyhlášen rozsudek Jednání zrušeno: Ne</p>		
Vydání rozhodnutí ▼	16.04.2014	Zobrazit
Vyřízení věci ▼	16.04.2014	Zobrazit
Nañřízení jednání ▼	19.03.2014	Zobrazit
Vrácení spisu ▼	19.02.2014	Zobrazit
Nañřízení jednání ▼	19.02.2014	Zobrazit
Vyřízení opravného prostředku ▼	12.02.2014	Zobrazit

Obrázek č. B.5 – Výpis stažených informací z InfoSoudu

 Nová data z eJustice.cz (12)

Přehled faktur

 [Vytvořit novou fakturu](#)

Nezaplacené faktury

Č.	Spis	Datum vystavení	Částka	Zobrazit	Akce
3.	2014-1 -	09. 03. 2014	96Kč	Textová PDF	Změnit na zaplacenou
2.	2020-16 - Rozvod manželů, započetí 1.2.2014	08. 03. 2014	40Kč	Textová PDF	Změnit na zaplacenou

Zaplacené faktury

Č.	Spis	Datum vystavení	Částka	Zobrazit
1.	2020-16 - Rozvod manželů, započetí 1.2.2014	08. 03. 2014	16Kč	Textová PDF

Obrázek č. B.6 – Přehled faktur

Příloha C – CD se zdrojovými soubory

Obsah jednotlivých adresářů:

/docs – text diplomové práce v PDF a zdrojová verze ve formátu ODT

/src – zdrojové soubory projektu

/install – návod k instalaci