

UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

Podpora elektronického vzdělávání

Jan Plecháč

Diplomová práce  
2010

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2009/2010

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan MACHOŇ**  
Osobní číslo: **00000**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Webové aplikace – využití PHP a MySQL**  
Zadávající katedra: **Katedra informačních technologií**

### Zásady pro vypracování:

*(The following text is intentionally blurred for privacy and security reasons. It contains the detailed instructions and requirements for the thesis project.)*

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

\*Kathy Sierra, Bert Bates: **Head First Java**, O'Reilly Media 2005, počet stran 720, ISBN-10 1600330002

\*Kline Kevin, Kline Daniel, Hunt Brand: **SQL in a Nutshell**, O'Reilly Media 2008, počet stran 592, ISBN-10 0596518846

\*Ajit Sagar, Sue Spielman a kol.: **Professional Java Server Programming J2Ee 1.4 Edition**, Wrox Press 2003, počet stran 850, ISBN-10 1861008139

\*Vivek Chopra, Sing Li, Jeff Genender: **Professional Apache Tomcat 6**, WROX Press 2006, ISBN-10: 0471753610

\*Sun Microsystems: **The Java EE5 Tutorial** [online], 2007 [cit. 2009-10-08], dostupný z:

<http://java.sun.com/javaee/5/docs/tutorial/doc/docinfo.html>

Vedoucí bakalářské práce:

**Ing. Lukáš Čegan**

Katedra informačních technologií

Datum zadání bakalářské práce: **15. ledna 2010**

Termín odevzdání bakalářské práce: **14. května 2010**



prof. Ing. Simon Karasmanov, Dr.  
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2010

## **Prohlášení autora**

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 14. 05. 2012

Jan Plecháč

## **Poděkování**

Zde bych chtěl poděkovat především svému vedoucímu Ing. Janu Fikejzovi za jeho cenné rady a připomínky při tvorbě diplomové práce, že si na mne udělal čas, i když to pro něj nebylo snadné.

**Anotace**

Tato práce se zabývá vytvořením aplikace pro podporu elektronického vzdělávání a dále seznamuje čtenáře s touto aplikací.

**Klíčová slova**

E-learning, C#, LINQ, Entity Framework, ASP.NET, Sharepoint

**Title**

Support for electronic education

**Annotation**

This work deals with creating software for the support for electronic education and to acquaint the reader with this software.

**Keywords**

E-learning, C#, LINQ, Entity Framework, ASP.NET, Sharepoint

# Obsah

<b>1</b>	<b>Úvod</b> .....	<b>1</b>
<b>2</b>	<b>Základní pojmy</b> .....	<b>2</b>
2.1	E-learning .....	2
2.2	E-kurz .....	3
2.3	CMS.....	3
2.4	C3MS.....	3
2.5	LMS .....	4
2.6	LCMS .....	4
2.7	SCORM .....	4
2.8	AICC.....	5
2.9	IMS .....	5
2.10	LOM .....	5
<b>3</b>	<b>Historie e-learningu</b> .....	<b>6</b>
<b>4</b>	<b>Typy E-learningu</b> .....	<b>7</b>
4.1	Learner-led e-Learning .....	7
4.2	Instructor-led e-Learning .....	8
4.3	Facilitated e-Learning.....	8
4.4	Embedded e-Learning.....	9
4.5	E-Mentoring.....	9
4.6	M-Learning .....	9
<b>5</b>	<b>Srovnání E-learningu</b> .....	<b>10</b>
5.1	Výhody a nevýhody e-learningu.....	11
5.1.1	Výhody e-learningu .....	11
5.1.2	Nevýhody e-learningu .....	12
<b>6</b>	<b>Analýza</b> .....	<b>13</b>
6.1	Identifikace požadavků.....	13
6.1.1	Funkční požadavky.....	13
6.1.2	Nefunkční požadavky .....	14
6.2	Případ užití.....	15
<b>7</b>	<b>Použité technologie</b> .....	<b>18</b>
7.1	XHTML .....	18
7.2	CSS .....	18

7.3	JavaScript.....	19
7.4	ASP.NET .....	19
7.5	Telerik ASP.NET AJAX Controls.....	21
7.6	LINQ.....	21
7.7	Entity Framework .....	22
7.8	Sharepoint .....	23
7.9	Hermes Framework .....	25
<b>8</b>	<b>Uživatelská část.....</b>	<b>26</b>
8.1	Hlavní strana.....	26
8.2	Úprava detailu kurzu.....	27
8.3	Editace uživatelů v kurzu.....	27
8.4	Seznam testů v kurzu .....	28
8.5	Správa studijních materiálů .....	28
8.6	Správa testu.....	28
8.7	Editace otázky v testu .....	29
8.8	Detail kurzu .....	30
8.9	Detail testu .....	30
8.10	Průběh testu .....	31
8.11	Vyhodnocení testu .....	32
8.12	Statistiky testu.....	33
8.13	Dotazy na lektora.....	34
<b>9</b>	<b>Implementace .....</b>	<b>35</b>
9.1	Application layer .....	36
9.1.1	Popis struktury webové části .....	36
9.1.2	Návrh webové části .....	42
9.2	Business logic layer.....	51
9.2.1	Root .....	51
9.2.2	Entity BLL.....	51
9.2.3	Helpers.....	53
9.3	Data access layer .....	53
9.3.1	Interface .....	53
9.3.2	Data.....	53
9.3.3	Entity .....	54



9.3.4	Helpers.....	54
9.3.5	ElearningModel .....	54
<b>10</b>	<b>Závěr .....</b>	<b>58</b>
<b>11</b>	<b>Literatura .....</b>	<b>60</b>
<b>12</b>	<b>Seznam příloh .....</b>	<b>62</b>
	Příloha č.1: CD obsahující.....	62
	Příloha č.2: Srovnání e-learningových aplikací.....	62

## Seznam obrázků

Obrázek 1 - Learner-led e-Learning .....	7
Obrázek 2 - Instructor-led e-Learning .....	8
Obrázek 3 - Facilitated e-Learning .....	9
Obrázek 4 - případ užití .....	15
Obrázek 5 - příklad kaskádového stylu .....	19
Obrázek 6 - Funkční komponenty Microsoft .NET Framework .....	20
Obrázek 7 - Rozdělení LINQ technologie .....	22
Obrázek 8 - Princip práce Entity Framework .....	23
Obrázek 9 - Rozdělení Sharepoint technologie .....	24
Obrázek 10 - ukázka úvodní stránky aplikace .....	26
Obrázek č. 11 - ukázka stránky pro úpravu základních informací kurzu .....	27
Obrázek č. 12 - ukázka stránky pro správu uživatelů v kurzu .....	27
Obrázek č. 13 - ukázka stránky pro správu testů v kurzu .....	28
Obrázek č. 14 - ukázka stránky pro správu studijních materiálů v kurzu .....	28
Obrázek č. 15 - ukázka stránky se seznamem otázek v testu .....	29
Obrázek č. 16 - ukázka úpravy otázky v testu .....	29
Obrázek č. 17 - ukázka stránky detailu kurzu .....	30
Obrázek č. 18 - ukázka stránky detailu testu .....	31
Obrázek č. 19 - ukázka průběhu testu .....	31
Obrázek č. 20 - ukázka vyhodnocení testu lektorem .....	32
Obrázek č. 21 - ukázka statistik testů .....	33
Obrázek č. 22 - ukázka stránka pro dotazy na lektora .....	34
Obrázek č. 23 - Návrhový vzor aplikace .....	35
Obrázek č. 24 - Ukázka vlastní implementace tlačítka .....	41
Obrázek č. 25 - Ukázka vlastního grafického panelu .....	41
Obrázek 26 - Diagram tříd pro ukládání souborů .....	52
Obrázek 27 - Entitně-relační diagram .....	55

# 1 Úvod

Tato práce se zabývá vytvoření aplikace pro podporou elektronického vzdělávání, jinak nazývaného jako e-learning. Pod tímto pojmem je možné si představit čistou formu online vzdělání s maximálním využitím počítačových technologií. Díky těmto technologiím pak může student samostatně studovat vlastním tempem. Ukončení studia zde probíhá formou závěrečného testu.

V prvních kapitolách práce se nachází převážně teoretická část, kde bylo zapotřebí objasnit co to e-learning je a základní pojmy k němu se vztahující. S tím vším je spjat i jeho vývoj a to od tzv. vyučovacích automatů po dnešní online systémy. Během tohoto vývoje vnikly různé typy e-learning aplikací podle různých požadavků na styl výuky.

V další části je srovnání nejčastěji nasazovaných aplikací pro elektronické vzdělávání a popsány jejich výhody a nevýhody. Na základě srovnání bylo zvoleno, jakým způsobem a kde podpořit elektronické vzdělávání. Je zde tedy nastíněno, co a jak se bude vyvíjet. K vývoji bylo zapotřebí se rozhodnout jaké technologie použít. Tyto technologie by měly být co nejmodernější, aby zajišťovaly snadnou dostupnost, rychlost a přátelské uživatelské prostředí.

V poslední části je popsána uživatelská část a implementace e-learningu. V uživatelské části jsou popsány jednotlivé stránky aplikace. Popis obsahuje, co se na jaké stránce vyskytuje a k čemu je používána. Tyto stránky jsou poté popsány i z implementačního hlediska, jak byli vytvářeny a kde vznikli případné problémy. Na konci se se pak nachází zhodnocení aplikace v podobě závěru.

## 2 Základní pojmy

Zde naleznete vysvětlení základních pojmů, které k E-learningu patří. Tyto pojmy jsou základem k pochopení této práce a termínu E-learning.

### 2.1 E-learning

Slovo e-learning znamená v překladu elektronické učení, což v dnešní době nelze jednoznačně definovat. Je to díky tomu, že vše se vyvíjí a tedy i e-learning. Proto na internetu můžeme nalézt mnoho definic.

- E-learning je proces, který nám dovoluje popsat a řešit tvorbu výuky a její dostupnost tak, aby byla zajištěna automatická zpětná vazba. Zpětnou vazbou se rozumí testy, které se vyhodnotí hned po jeho dokončení.[1]
- E-learning je forma studia, kde získáváme znalosti pomocí elektronických zařízení.[1]
- E-learning je vzdělávací postup, při němž využíváme multimediální technologie, internet a jiná elektronická zařízení, pomocí kterých můžeme zajistit zlepšení výuky.[2]

Cíl e-learningu může být jeho rozšíření natolik, aby byl dostupný každému za přijatelnou cenu. Přičemž musí být zajištěna určitá kvalita tohoto vzdělávání.

E-learning a jiné pokročilé technologie nám dávají možnost, jak umožnit studium v různých formách také různě znevýhodněným skupinám lidí. To si můžeme představit jako počítačové výukové programy či různé audio knihy. Je však důležité si uvědomit, že ne pro všechny obory je vhodný, protože někdy je potřeba zachovat lidský faktor. Příkladem je třeba autoškola, kde teorii se pomocí e-learningu naučíme, ale řídit auto asi stěží.

## 2.2 E-kurz

K. Květoň (2005) zmínil, že jako u denního studia je základem kurz, tak i u elektronického vzdělávání najdeme elektronický kurz, neboli zkráceně e-kurz. Tento e-kurz nám vždy zastupuje jeden předmět a je upraven tak, aby byl zajištěn požadovaný cíl vzdělání daného e-kurzu. Všeobecně e-kurz charakterizují tři složky:

- **Vzdělávací obsah materiálů** – Daný vzdělávací obsah e-kurzu by měl být uspořádán tak, aby byla zajištěna automatická zpětná vazba pro samostudium.
- **Znovupoužitelné vzdělávací objekty** – Pod slovem objekty si můžeme představit různé vzdělávací materiály, ať jsou to odborné texty, obrázky v digitální formě, videonahrávky či internetové stránky. Tyto všechny vzdělávací objekty můžeme jednoduše znovu použít například u jiného e-kurzu. Znovu použitelnost po uložení do našeho centrálního úložiště nebo jinak elektronické knihovny je jednoduché, stačí pouze uvést odkaz či cestu k danému objektu.
- **Elektronická správa e-kurzů** – Pod tímto pojmem si můžeme představit organizaci a administraci daného e-kurzu.

## 2.3 CMS

Je to všeobecný software sloužící ke správě obsahu, jak napovídá celý název content management systém. Pod touto zkratkou si také můžeme představit redakční systém, běžící na internetových stránkách. Ten pak slouží k vytváření obsahu v podobě stránek a článků. Je navržen tak, aby ho bylo možné rozšířit pomocí modulů. Může se tedy vytvořit modul pro e-learning, který by se do CMS nainstaloval.

## 2.4 C3MS

Je to speciální typ CMS, který rozvíjí doménové specifické obsahy v rámci komunity. Tyto systémy využívají mechanismy určené pro spolupráci. Je možné si ho představit jako moduly pro vytváření posudků, kritik, vyhodnocování nebo modul pro zjištění kdo je online. C3MS můžeme tedy použít jako standartní CMS systém nebo pomocí něj vytvořit weblogy, které mezi sebou spolupracují[16].

## 2.5 LMS

Tato zkratka znamená Learning Management Systém, což z překladu znamená řídicí výukový systém. Tento systém je nejstarším představitelem e-learningu. Představit si to můžeme jako aplikaci, která nám dovoluje organizovat a spravovat výuku v rámci E-learningu[3].

Tyto aplikace by měly obsahovat nástroje pro:

- komunikaci studenta s lektorem a to pomocí diskuzí, chatu, emailové korespondence, atd.,
- administraci kurzů, testů a studijních materiálů,
- vyhodnocení studentů a vypočítání statistik.

## 2.6 LCMS

Zkratka LCMS (Learning Content Management System), zde oproti LMS se nám objevuje anglické slovo content, které znamená v překladu obsah. Z toho nám vyplývá, že řízení nebude zaměřené na výuku, ale na obsah. Tím je míněno samozřejmě obsah studijních materiálů. LCMS je tedy nástroj pro tvorbu, znovu použití, distribuci a úpravu studijních materiálů a měl by řešit věci, jako například:

- možnost více uživatelského přístupu a tvorby,
- upravovat obsah materiálu a zajistit možnost znovupoužití,
- sledování práce uživatelů se studijními materiály.

## 2.7 SCORM

O dvou systémech LMS a LCMS můžeme říct, že jsou na sobě nezávislé. Existuje však jedno propojení a to v podobě normy SCORM (Shareable Content Object Reference Model), což můžeme považovat za referenční model e-learningu. Tento model je souborem specifikací a standardů tak, aby studijní obsah vytvořený v LCMS bylo možno použít na jakémkoliv LMS. Z toho a i z názvu můžeme odvodit, že jde o standard zajišťující znovu použitelnost vytvořeného studijního obsahu. Díky standardu SCORM, a pomocí něj vytvořený studijní obsah, se dá použít na jakémkoliv LMS systému[3].

## 2.8 AICC

Zkratka v sobě skrývá slova Aviation Industry CBT Committee, což je organizace, která vytvořila jeden z prvních standardů. Tento standard v sobě skrývá specifikace o tom, jak by měla vypadat výměna studijních materiálů mezi kurzy či jinými aplikacemi. Další specifikace jsou například způsob uchovávání dat a výsledků kurzů[15].

## 2.9 IMS

Je sdružení přibližně 150 organizací, které mají společný cíl, navrhovat standardy pro výměnu dat. Tato výměna by neměla být závislá na daném přenášeném oboru dat. Přenášeni by mělo probíhat v podobě XML souborů. Vytvářené standardy touto organizací odráží potřeby jak výrobců LMS aplikací, poskytovatelů obsahu, tak i uživatelů těchto aplikací[15].

## 2.10 LOM

Tento standard určuje ve studijních materiálech strukturu metadat, dále definuje vlastnosti k jeho úplnému popisu. Celkem je to 79 metadatových vlastností rozdělených do 9 kategorií[15]:

- Obecné údaje
- Životní cyklus
- Meta-metadata
- Technické informace
- Studijní informace
- Licenční podmínky
- Vztahy
- Anotace
- Klasifikace

### 3 Historie e-learningu

H. Stříteská (2007) popsala vývoj e-learningu v České republice, který vypadal následovně:

V druhé polovině šedesátých let dvacátého století se objevily tzv. vyučovací automaty. Byly to experimentální stroje sloužící k učení. Podle některých názorů byl u nás vyvinut jeden z nejlepších vyučovacích automatů jménem Unitutor. Jednalo se o elektronické zařízení na bázi tranzistorů s obrazovkou, kde vykládaná látka se zobrazovala na jednotlivé stránky. Na konci každé stránky se nacházela kontrolní otázka, u které bylo několik možností pro odpověď. Podle zvolené odpovědi bylo možné program větvit a přecházet na další stránky. U tohoto automatu byla zajištěna automatická zpětná vazba, v podobě zobrazení špatných či správných odpovědí na kontrolní otázky.

V druhé polovině osmdesátých let dvacátého století vcházejí na trh osobní počítače. Tyto počítače obsahovaly šestnáctibitovou sběrnici a měly za následek, že se počítače začínají objevovat v domácnostech. Díky velkému zájmu, se začaly vyvíjet kancelářské aplikace ve velkém množství. Takovýto pokrok vedl k tomu, že se počítač začal používat jako učící a zkoušející stroj. Také vznikaly názory, že by počítače mohly nahradit učitele a díky velkému rozšíření počítačů se mohly tyto teorie začít ověřovat.

Na začátku devadesátých let dvacátého století se začaly vyvíjet aplikace, u kterých by bylo možno dlouhodobě sledovat výukový proces. Bylo potřeba, aby takováto aplikace v sobě zahrnovala výklad učiva, jeho procvičování a závěrečné testy. Důležité bylo, aby ve výkladu učiva bylo možné využívat například grafiku, animace a zvuk. Zde se stal učitel pouze jakousi obsluhou a kontrolou, protože testy se automaticky vyhodnotily a zobrazily výsledek, zda student prošel či ne.

Na přelomu dvacátého a jednadvacátého století vznikaly na univerzitách tzv. multimediální zdroje, kam se přesouvaly zdroje jako sylaby, obsahy přednášek a knihovní zdroje. Tento nárůst popularity elektronického vzdělávání vedl ke vzniku virtuálních univerzit. Pomocí těchto virtuálních univerzit mohl student získat certifikát z daného kurzu, aniž by musel někam osobně docházet. Vše probíhalo v rámci internetu online. Představa byla taková, že by zaměstnaní dospělí mohli pracovat a zároveň vystudovat vysokou školu přes internet.



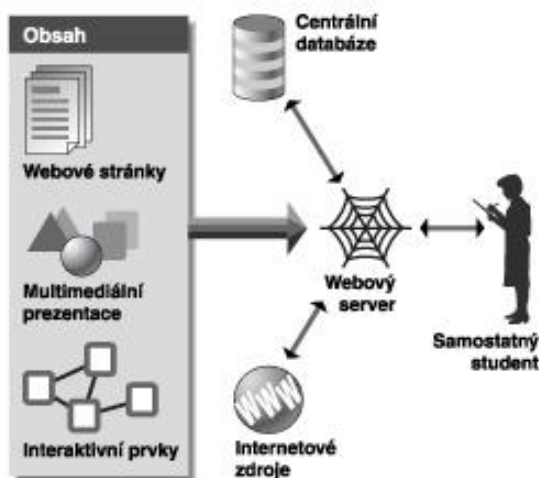
## 4 Typy E-learningu

E-learning můžeme rozdělit na nesčetné množství. Zde je zmíněno pár rozdělení podle P. Šiky (2005).

### 4.1 Learner-led e-Learning

Jak již překlad napovídá, tak jde e-learning, který vede student. Jde o druh samostudia, který je určen pro samostatné studenty. Tito studenti mezi sebou nemohou nijak komunikovat, proto všechny informace musejí čerpat z materiálu, které jsou přiřazeny k danému kurzu. Tyto materiály se pohybují od interaktivních aplikací, multimediálních či standartních prezentací, až po celé webové stránky. Z tohoto samostudia vyplívá, že se zde nenachází žádný lektor, může se zde však nacházet osoba, která sleduje jednotlivé kroky studenta. Tyto kroky se nacházejí v centrální databázi a slouží pro zlepšení výuky. Způsob práce studenta je znázorněn na obrázku č. 1.

#### LEARNER-LED E-LEARNING

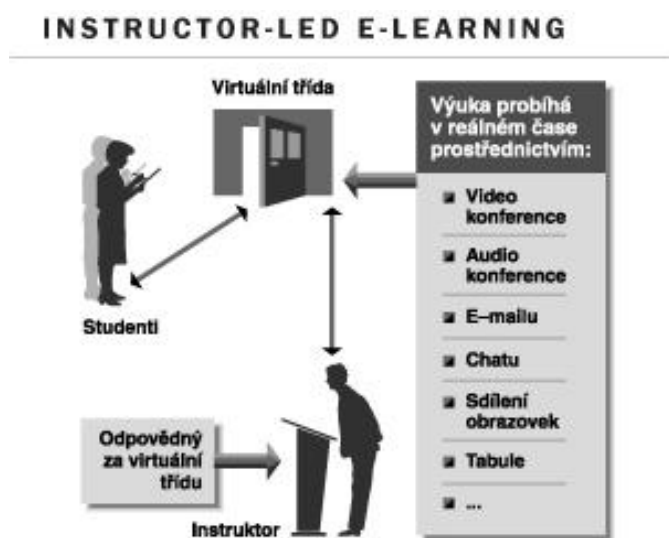


Obrázek 1 - Learner-led e-Learning<sup>1</sup>

<sup>1</sup> Zdroj: Dostupný na WWW [http://ihned.cz/2-22676725-000000\\_d-92](http://ihned.cz/2-22676725-000000_d-92)

## 4.2 Instructor-led e-Learning

V tomto typu e-learningu fungují virtuální třídy, které jsou vedené instruktorem. Instruktor zde může prezentovat pomocí video konference. Jako pomůcky má se studenty k dispozici technologie, které pracují v reálném čase, jako sdílení pracovních ploch, chat, dotazování pomocí elektronické korespondence nebo klasický telefon. Těchto technologií stále přibývá. Tato výuka se velmi podobá normální formě studia, protože student může svého instruktora vidět v reálném čase a položit mu případné dotazy. Tento typ studia je zobrazen na obrázku č. 2.



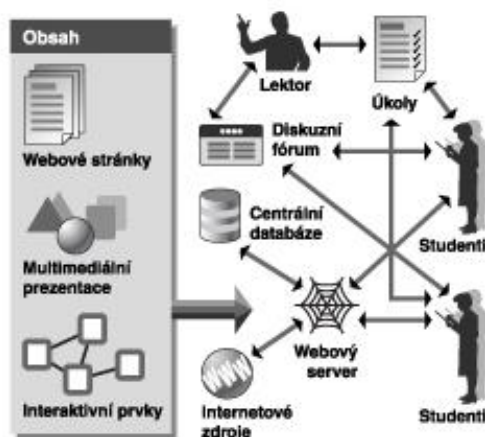
Obrázek 2 - Instructor-led e-Learning<sup>2</sup>

## 4.3 Facilitated e-Learning

U této formy e-learningu je kladen důraz na úkoly, které zadává lektor. Ten se liší od instruktora tím, že neprovádí žádnou výuku, ale zadává studentům různé úkoly pro splnění. Studenti tedy získávají informace pro splnění úkolů v rámci samostudia, diskuzích s jinými studenty nebo lektorem. Tento typ studia ocení především studenti, kteří si rádi zvolí svoje tempo studia a nechtějí být omezováni předem daným rozvrhem. Příklad takovéto spolupráce studentů a lektora je zobrazen na obrázku č. 3.

<sup>2</sup> Zdroj: Dostupný na WWW [http://ihned.cz/2-22676725-000000\\_d-92](http://ihned.cz/2-22676725-000000_d-92)

## FACILITATED E-LEARNING



Obrázek 3 - Facilitated e-Learning<sup>3</sup>

### 4.4 Embedded e-Learning

Tato výuka nastává jen v případě potřeby. To znamená, pokud student nemůže vyřešit problém. Pokud tedy chceme nastavit router na internet a nevíme jak, tak si můžeme vyhledat nápovědu nebo dokonce spustit průvodce instalací, který nám pomocí animací může ukázat dané řešení.

### 4.5 E-Mentoring

Základem e-mentoringu je mentor, což je učitel, který má ke studentům mnohem blíž. Tento mentor se snaží studentovi předat všechny znalosti z dané problematiky, dokonce i ty, které nejsou napsané v žádných materiálech, které má student k dispozici. Jde zde tedy o předávání zkušeností. K tomu aby mentor mohl vše uskutečnit, využívá nejmodernější technologie jako je chat, video konference apod. Takto získané znalosti a zkušenosti se mnohem lépe uplatňují v reálném světě.

### 4.6 M-Learning

Jako nejmladší formou elektronického vzdělání je m-learning, který se občas objevuje pod názvem m-training. Tato forma výuky probíhá prostřednictvím mobilních telefonů a kapesních počítačů (PDA), které se velmi rychle vyvíjejí a technické možnosti se značně přibližují k osobním počítačům. Díky velkému rozšíření těchto zařízení se doufá, že ke vzdělání přivede velké množství lidí.

<sup>3</sup> Zdroj: Dostupný na WWW [http://ihned.cz/2-22676725-000000\\_d-92](http://ihned.cz/2-22676725-000000_d-92)

## 5 Srovnání E-learningu

V příloze č. 2[6] je vidět srovnání aplikací pro e-learning, které bylo vyhotoveno nezávislou institucí. Ta na základě zkoumání vytvořila základní kritéria pro aplikace určené pro e-learning. Tyto kritéria jsou ve srovnání zastoupeny třemi barvami, podle toho jak jednotlivé aplikace splňují tyto kritéria, jako jsou:

- Komunikace
- Didaktika
- Generování obsahu
- Použitelnost
- Administrace
- Cena

Na základě tohoto srovnání byl potřeba nalézt způsob jak rozšířit působnost e-learningu. Byla zvolena možnost vytvoření e-learning aplikace jako samostatné stránky, tak i jako modul pro Microsoft Office Sharepoint. Ten slouží k vytvoření intranetových a internetových stránek a je popsán v kapitole 7.8. Pro tuto technologii existuje velmi málo aplikací v podobě e-learningu. Některá mají velmi složitá nastavení a jiná zas nejsou uživatelsky přívětivá. Opakem jsou pouze aplikace placené. Zdarma existuje modul Sharepoint learning Kit. Ten je nástupcem aplikace ClassServer, která je vidět v tabulce srovnání. Tento modul je však uživatelsky nepřívětivý a to hlavně pro administrátora a lektora.

## 5.1 Výhody a nevýhody e-learningu

E-learning v sobě skrývá některé výhody a nevýhody, které naznačil F. Mazal (2006) ve své přednášce. To je rozvinuto v následujících dvou kapitolách.

### 5.1.1 Výhody e-learningu

Zde jsou popsány výhody e-learningu, a to vzhledem jak studujícím, tak i k provozovatelům.

- **Nezávislost** – Pokud by v normálním kurzu lektor onemocněl a nebyla by náhrada, nemohl by takovýto kurz po dobu nemoci lektora dále probíhat. U e-kurzu, kde jsou materiály dostupné online, může e-kurz dále probíhat a být řádně zakončen v podobě testu s automatickým vyhodnocením.
- **Možnost studia kdekoliv a kdykoliv** – Studující není svazován rozvrhem, jako v denním studiu, ale může si sám zvolit, kdy bude studovat, protože daný e-kurz a jeho studijní obsah je přístupný 24 hodin denně a tedy kdykoliv. Pokud je e-learning provozován pomocí internetu, může se studující na něj připojit odkudkoli pomocí počítače, internetu a webového prohlížeče.
- **Studium ve vlastním tempu** – Zde si studující může zvolit svoje vlastní tempo studia, ať to je rychlost probírání určité látky nebo kdy a jak dlouho bude studovat. To je výhodné, pokud danou látku studující zná, a tak mu to stačí v rychlosti projít nebo pokud o dané problematice nic netuší a potřebuje na ni víc času než ostatní.
- **Omezení nákladů** – Toto tvrzení má dvě odlišné části a to z pohledu studujícího a provozovatele e-learningu. Z pohledu studujícího je to jednoznačné plus, a to díky omezení nákladů na dopravu. Z druhého pohledu provozovatele e-learningu jsou počáteční náklady vysoké, to se však mění s postupným přibýváním studujících. Projevuje se to ušetřením nákladů za lektory, učebny, studijní pomůcky, apod.
- **Snazší administrace** – Pokud nastala změna, například v zákonu, nebo se daná problematika týkající se vytvořeného e-kurzu změnila, stačí nám jednoduše upravit elektronické materiály a studium může probíhat bez jakéhokoli problému dál. Přičemž u normálního kurzu by se musely znovu koupit či nechat vytisknout nové studijní materiály.
- **Získání vyšší úrovně počítačové gramotnosti u studujících** – Studující při práci s elektronickými studijními materiály, a vlastně s celým e-learningem, si zlepšují svoji schopnost práce s počítačem.

- **Okamžitý přístup ke statistikám** – Lektor má v průběhu kurzu možnost zobrazit úspěšnost testů, ať přípravných nebo zakončujících. Druh zobrazovacích statistik se liší od určité implementace e-learningu.

### 5.1.2 Nevýhody e-learningu

Stejně, jako se výhody dotýkaly studenta a provozovatele, tak je to u nevýhod e-learningu popsané v této kapitole.

- **Vysoké vstupní náklady** – Provozovatel má na začátku velké náklady a to na zakoupení daného softwaru pro e-learning a pronajmutí či zakoupení počítačových serverů, kde tento software poběží. Tím to však nekončí, protože je potřeba zakoupit či vytvořit studijní obsah pro dané e-kurzy.
- **Nutná vybavenost počítačem a připojení k internetu** – Studující a jeho lektor potřebují počítač na práci s daným e-learning softwarem a také internetové připojení, pokud e-kurz probíhá online.
- **Chybí fyzický kontakt s lektorem** – Studující nemá žádný osobní kontakt s lektorem, ten je nahrazen elektronickou korespondencí jako je například e-mail či elektronická korespondence v rámci e-learning aplikace.
- **Zaškolení v e-learning aplikaci** – Potřeba zaškolit lektory a studující. Může probíhat osobně, multimediální nahrávkou, či textovým manuálem.

## 6 Analýza

Tato kapitola se zabývá analýzou potřebnou pro vytvoření e-learning aplikace. Jsou zde popsány požadavky a případ užití.

### 6.1 Identifikace požadavků

V kapitole se nacházejí požadavky na aplikaci. Ty se dělí na dvě základní skupiny a to funkční a nefunkční. Funkční se zaměřují přímo na aplikaci a nefunkční požadavky pak jsou ty, které nám aplikaci nějakým způsobem omezují nebo také doplňují, příkladem může být dostupnost, přívětivé uživatelské prostředí apod.

#### 6.1.1 Funkční požadavky

Zde jsou popsány funkční požadavky kladené na e-learning aplikaci.

##### **Přihlášení**

Uživatel se před použitím aplikace musí vždy přihlásit a na základě mu přidělené role dostane oprávnění k určitým stránkám.

##### **Správa kurzů**

V aplikaci bude zajištěno vytváření, mazání a editace kurzu. Editace musí obsahovat možnost zvolení od kdy, do kdy bude kurz publikován a aktivní.

##### **Přiřazení role uživateli**

V aplikaci bude zajištěno přiřazení rolí studenta a lektora jednotlivým uživatelům, podle kterých bude moci provádět operace pro něj určené.

##### **Správa studijních dokumentů v kurzu**

Aplikace bude obsahovat prostředí pro ukládání studijních dokumentů, jejich editaci a mazání.

##### **Správa, vyhotovení, hodnocení testu**

U testu v aplikaci bude zajištěno jeho vytvoření a editace. To zahrnuje i vytváření otázek v testu. Dále vyhotovení testu studentem a v neposlední řadě jeho vyhodnocení lektorem nebo automaticky aplikací.

##### **Nastavení testu**

Test bude možné nastavit, aby při vykonávání testu student se mohl pohybovat libovolně mezi otázkami, náhodné zobrazení otázek, či nutnost odpovědět na každou otázku.

### **Více typů otázek**

V aplikaci se bude nacházet více typů otázek. Otázky mohou být typu, kdy student na ně odpoví vlastními slovy v podobě textu. Nebo otázky s nabídnutými odpověďmi a to jednou nebo vícero správně.

### **Komunikace mezi lektorem a studentem**

V aplikaci bude možnost komunikace mezi lektorem a studentem v podobě textového dotazu studenta na lektora.

### **Dva typy testů**

Testy v aplikaci budou dvojího typu. Hlavním typem bude závěrečný test kurzu a další bude přípravný test sloužící pro procvičování.

### **Grafický report**

Lektor v aplikaci bude mít přístup ke statistikám jednotlivých testů, které budou vyobrazeny v přehledných grafech.

## **6.1.2 Nefunkční požadavky**

Kapitola obsahuje souhrn nefunkčních požadavků, kladených na aplikaci.

### **Dostupnost 24x7 prostřednictvím internetu**

Aplikace bude dostupná na internetu 24 hodin denně, 7 dní v týdnu. Výjimky mohou nastat v případě údržby a to v době nejmenší vytíženosti serveru.

### **Možnost rozšiřitelnosti**

Aplikace bude připravena vzrůstající se počet uživatelů a studijních oblastí.

### **Snadné vytvoření nové jazykové mutace**

Jazykové mutace v aplikaci bude možné jednoduše rozšířit.

### **Bezpečnost**

Aplikace bude zabezpečená proti neoprávněným zásahům z internetu.

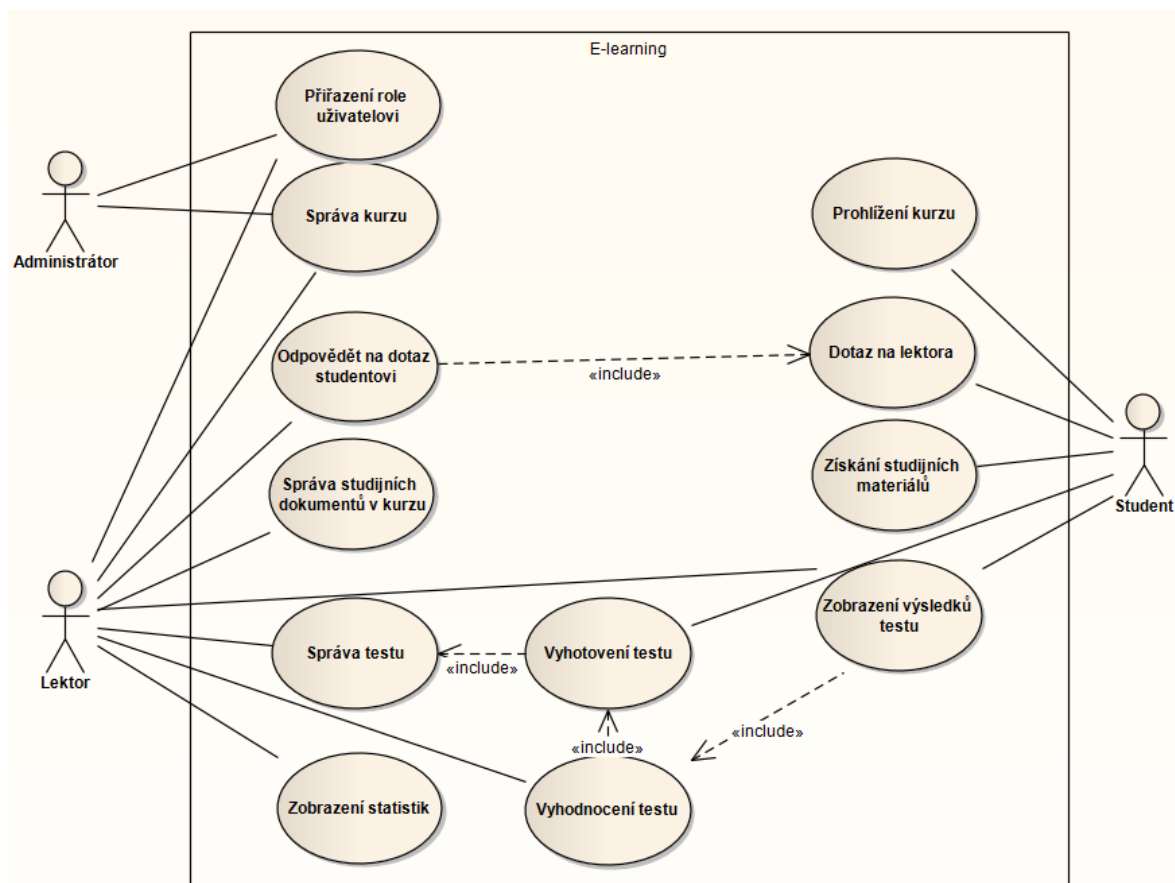
### **Výkon a spolehlivost**

Při velkém počtu dat musí aplikace pracovat rychle a spolehlivě i při velkém vytížení.



## 6.2 Příklad užití

Na obrázku č. 4 je vidět případ užití e-learning aplikace. Ten byl vyhotoven podle požadavku na aplikaci, které jsou popsány v kapitole výše. V tomto případě užití se nacházejí tři základní role administrátor, lektor a student. Každý uživatel, nezávisle na jeho roli se musí přihlásit. Pro přehlednost se v diagramu činnost přihlášení nenachází, protože jsou na ní závislé všechny ostatní činnosti.



Obrázek 4 - případ užití

### Přířazení role uživateli

Administrátor po vytvoření kurzu do něj přiřadí uživatele pomocí jejich přihlašovacích jmen. Těmto uživatelům se přiřadí role lektora nebo studenta. Tuto operaci může provádět i lektor v tomto kurzu.

### Správa kurzu

Správa kurzu v sobě obsahuje jeho vytvoření a smazání, což může provádět pouze administrátor. Dále se zde nachází přiřazení uživatele do kurzu, správa studijních dokumentů v kurzu a samozřejmě úprava detailních informací o kurzu. Tyto všechny věci už mají lektor a administrátor společné.

### **Správa studijních dokumentů v kurzu**

Lektor a administrátor mohou do kurzu hromadně nahrávat studijní materiály, editovat jejich popis a mohou je také smazat.

### **Správa testů**

Ve správě testů se nachází editace detailu testu název, popis apod. Také se zde nachází správa otázek, jejich vytvoření, editace a smazání. Bude zde možnost vytvoření 3 typů otázek. První bude textová, druhá s jednou možnou odpovědí a třetí bude otázka, kde bude více možných odpovědí.

### **Prohlížení kurzu**

Student si může otevřít kurz, do kterého je přiřazen. V tomto kurzu uvidí všechny dostupné testy, studijní materiály a informace o samotném kurzu.

### **Dotaz na lektora**

Student bude mít možnost se dotázat ohledně informací týkající se daného kurzu, případně i testu.

### **Odpověď na dotaz studentovi**

Pokud student pošle dotaz na kurz, zobrazí se tento dotaz všem lektorům v dotazovaném kurzu. Libovolný lektor, pak může studentovi odepsat.

### **Získání studijních materiálů**

Student bude mít přístup ke všem studijním materiálům v kurzu, kde je přiřazen. Tyto materiály si poté může uložit k sobě do počítače.

### **Vyhotovení testu**

Po zveřejnění testu v kurzu, může student tento test vykonat. Tento test může být přípravný v rámci výuky nebo závěrečný k ukončení kurzu.

### **Vyhodnocení testu**

Po vyhotovení testu studentem, se pak tento test vyhodnotí sám aplikací e-learning, nebo ho vyhodnotí lektor. To záleží, jestli test obsahuje otázku typu text nebo je test tak nastaven.

### **Zobrazení výsledků**

Po vyhodnocení testu si student bude moci zobrazit výsledky testu spolu s jeho hodnocením.

**Zobrazení statistik**

Lektor bude mít možnost si v daném testu zobrazit graficky statistiky. Statistiky budou zobrazovat úspěšnost jednotlivých studentů v testu a procentuální úspěšnost studentů v jednotlivých otázkách.

## 7 Použité technologie

### 7.1 XHTML

Jak už nám název napovídá, bude mít označení XHTML něco společné se značkovacím jazykem HTML (Hyper Text Markup Language). Písmeno X zde zastupuje také značkovací jazyk, ale v obecné formě XML (Extensible Markup Language). Tyto dva jazyky jsou všeobecně známé, kde HTML byl vytvořen na psaní hypertextových dokumentů vkládaných na internet. Zatímco jazyk XML byl spíše vyvinut pro přenos dat. Hlavním rozdílem mezi těmito jazyky je, že HTML zobrazí stránku i v případě, že nastala chyba. Ale XML pokud nastane jakákoli chyba, třeba neuzavření závorky, tak zobrazí pouze chybu a text nikoli[12].

Asi jako hlavní důvod spojení jazyka XML a HTML byl, že díky jazyku XML, který slouží pro přenos dat a nebere ohled jak, jsou data naformátována. Jazyk XHTML má pak mnohem lepší kompatibilitu s jinými datovými formáty.

### 7.2 CSS

Zkratka v sobě skrývá slova Cascading Style Sheets, která v překladu znamenají tabulka kaskádových stylů. Je to tedy jazyk, který slouží k efektivnímu formátování dokumentů, k úpravě stylu webové stránky. Největší využití má určitě v jazycích HTML a XHTML, kde pomocí tohoto jazyka můžeme nastavit styl celé stránky, ale i jednotlivým prvkům.

Velkou výhodou je, že nijak neovlivňuje obsah dokumentů. Styly můžeme mít vnořené na stránce, to se však nedoporučuje, jak kvůli velikosti dat, které zabírají, tak hlavně pro nepřehlednost. Proto se ukládají do samostatných souborů s příponou `.css`, kde se nastaví styly pro jednotlivé značky.

Při psaní jednoho kaskádového stylu, píšeme vlastně pravidlo, které se skládá ze sektoru a deklarace. V sektoru se nachází výpis značek, pro které bude platit deklarace. Deklarace je pak určena pro nastavení výčtu vlastností, kterým přiřazujeme hodnotu námi požadovanou. Jako příklad může být nastavení stylu tagu `h1` na obrázku č.5. Tag `h1` slouží pro zobrazení nadpisu prvního řádu a na obrázku mu nastavujeme modré písmo fontem `serif`, které je zarovnané na střed.



Obrázek 5 - příklad kaskádového stylu<sup>4</sup>

### 7.3 JavaScript

Název JavaScript nám napovídá, že tento programovací jazyk je nějak spojený s jazykem Java. Ale není tomu tak, je to pouze marketingový trik. Tento jazyk byl vyvinut firmou NetScape a slouží nám k vytváření skriptů s objektově orientovaným modelem. Tyto skripty mají největší využití ve stránkách HTML či XHTML, kde můžeme provádět operace rovnou na straně klienta, což je velká výhoda[13].

Výhoda spočívá v tom, že událost vytvořená na straně klienta se nemusí posílat na server a může být obsloužena rovnou na straně klienta. Toto nám ušetří data a čas, při kterém by se čekalo na odpověď od serveru. Příkladem tomu je validace dat ve formuláři, kde dokud nejsou data validní, tak nám je stránka nepovolí odeslat na server. Tímto se dá ošetřit i bezpečnost dané stránky. JavaScript se často používá k vytvoření různých vizuálních efektů zobrazujících se na stránce.

### 7.4 ASP.NET

ASP.NET je technologie, která je vyvinuta nad .NET Framework technologii od společnosti Microsoft. ASP.NET WebForm technologie nám umožňuje přechod od programování těžkých klientů pro Windows k programování webových aplikací se zachováním objektu, ovládacích prvků tak jako u formulářových aplikací. To znamená, že nám zůstávají například tlačítka a k nim dané metody k zachycení události jako je kliknutí. Tyto ovládací prvky pak při vykreslování generují (X)HTML kód pro zobrazení ve webovém prohlížeči.

<sup>4</sup> Zdroj: Dostupný na WWW <http://www.kosek.cz/vyuka/4iz228/prednasky/pic/style-css.png>

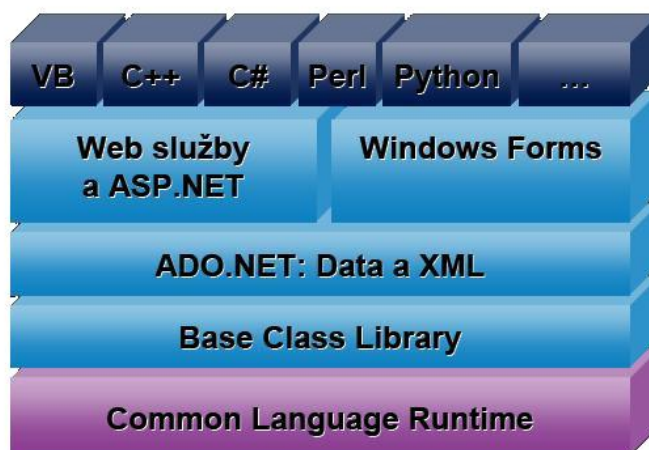
Jak zmíněno výše, ASP.NET je událostmi řízené programování a pro svoji práci potřebuje zachovat svůj stav. Je to kvůli uchování kontextu mezi jednotlivými požadavky.

Zde se může namítat jak tedy ASP.NET může pracovat na protokolu HTTP, který je bez stavový. Je to díky (X)HTML, Javascriptu a hlavně dvou technik[5]:

- **ViewState** – Ten zachovává stavy mezi dvěma postbacky (opakované zaslání formuláře na server) v skrytých formulářových polích. Ty jsou samozřejmě zakódované.
- **Sessionstate** – Všechny informace jsou na straně serveru, nejčastěji v URL, či cookie.

.Net Framework poskytuje ASP.NET systémové služby a dále k sobě obsahuje:

- Společný modul pro běh programů Common Language Runtime (CLR), kde CLR řídí vykonávání .NET kódu, včetně řízení paměti a životnosti objektů v paměti. Kromě těchto řídicích služeb CLR umožňuje vývojářům ladit aplikace, ošetřovat chyby a využívat dědičnost mezi jednotlivými vývojovými jazyky.
- Knihovny základních tříd NET, ze kterých mohou vývojáři odvozovat své vlastní třídy. Patří sem třídy poskytující základní datové typy nebo přístupy k datům, ale také třídy nabízející takové služby jako je kreslení, podpora síťových funkcí a mnoho dalších.[14]



Obrázek 6 - Funkční komponenty Microsoft .NET Framework<sup>5</sup>

Jako příklad, které komponenty obsahuje .NET Framework je vidět na obrázku č. 6.

U této kapitoly je čerpáno z literatur [5] a [14].

<sup>5</sup>Zdroj: Dostupný na WWW <http://owebu.blogger.cz/Programovani/ASP-NET-2-0-2-dil-Microsoft-NET-Framework-2-0->

## 7.5 Telerik ASP.NET AJAX Controls

Je to knihovna ovládacích prvků, která rozšiřuje základní prvky nacházející se v ASP.NET. Každý tento ovládací prvek od firmy Telerik v sobě skrývá mnohá vylepšení jak na straně serveru tak i klienta a obsahuje velmi obsáhlou dokumentaci s již připravenými scénáři pro snadnější pochopení a použití v praxi. Dále v sobě skrývají předem připravené skiny vzhledu. Je to velmi mocný nástroj, který při správném použití zjednodušuje vývojáři práci.

## 7.6 LINQ

U této kapitoly je čerpáno z internetového článku v literatuře [7] a ze své vlastní zkušenosti.

LINQ je zkratka Language INtegrated Query, což v češtině znamená integrovaný dotazovací jazyk. Tato technologie vznikla spolu s .NET Framework 3.5 a byl to velký zvrat v dotazování k datům na platformě .NET Framework.

Velkou výhodou je zde odhalování chyb již v době kompilace, to znamená, pokud uděláme chybu v dotazu, kompilátor nám zahlásí chybu a tím případně napoví, jak jí opravit. Díky tomu je naprostá většina chyb odhalena ihned.

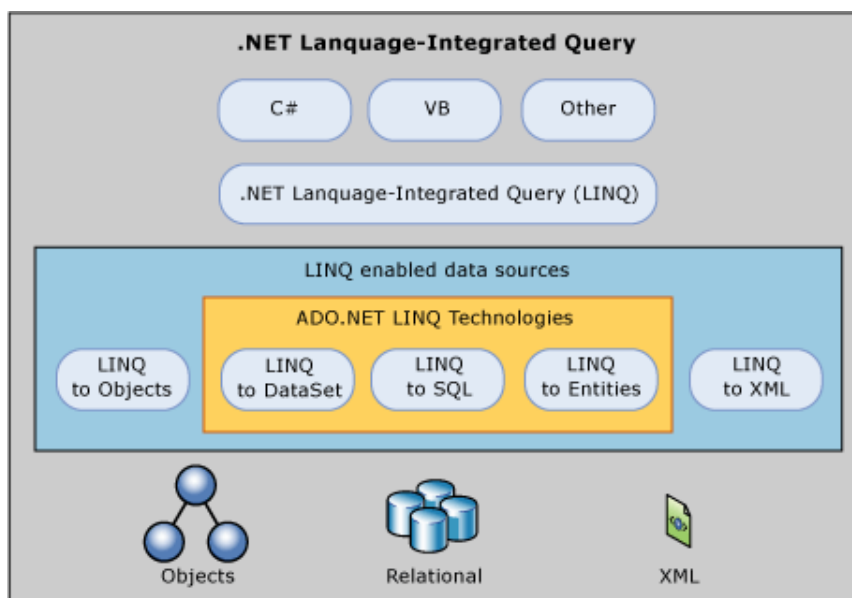
Technologie LINQ byla navržena tak, aby ji bylo možné jednoduše rozšířit. To si můžeme představit jako interface, pro který postupem času můžeme vytvářet jeho jednotlivé implementace. Toto je od .NET Framework 3.5 zajištěno pomocí Expression Trees a pomocí něj můžeme pracovat s kódem jako s daty. Díky tomu daná implementace LINQ může přeložit náš dotaz z kódu do formy, která je typická pro danou reprezentaci dat. Z tohoto můžeme říct, že nám sjednocuje způsob práci s daty různých typů nad .NET Framework 3.5.

V dotazovacích metodách technologie LINQ, kdy se nám má vrátit kolekce výsledků, je vrácena v generickém rozhraní `IEnumerable<T>`. Toto rozhraní je vybaveno operacemi vyhledání, seskupení nebo jen omezení množiny podle určité podmínky. Tyto podmínky jsou zde zastoupeny pomocí delegátů. Toto je velká výhoda, protože zde můžeme použít lambda výraz, který nám zpřehlední a zkrátí náš zápis v kódu.

Příklady implementací LINQ, které jsou implementovány ve verzi .NET Framework 3.5:

- **LINQ to Objects** – implementace LINQ pro standardní kolekce nacházející se v paměti,
- **LINQ to SQL** - implementace LINQ pro Microsoft SQL Server 2000 a vyšší,
- **LINQ to XML** - implementace LINQ pro práci s XML daty,
- **LINQ to DataSet** - implementace LINQ pro práci s ADO .NET datasety,
- **LINQ to Entities** – implementace LINQ pro práci s Entity Framework.

Názorný příklad implementací je uvedeno na obrázku č. 7, kde je vidět rozdělení na dotazování do relační databázi (Relational), objektů v paměti (Objects) a XML formátu.



Obrázek 7 - Rozdělení LINQ technologie<sup>6</sup>

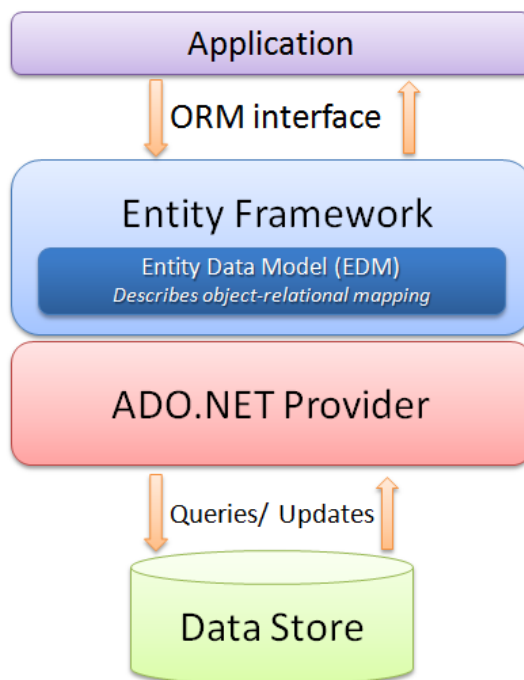
## 7.7 Entity Framework

Je to nástroj pro O/R mapování neboli objektově relační mapování. Je to programovací technika, která zajišťuje konverzi dat mezi relační databází a objektově orientovaným programovacím jazykem[9].

Při používání entit pak pracujeme s virtuální objektovou databází, do které se dotazujeme nejčastěji pomocí technologie LINQ. Pomocí LINQ vytvoříme dotaz na daný kontext a vrátí se nám výsledek, se kterým můžete dál pracovat (modifikovat data, mazat, přidávat přes kontext nová data atd.)[9].

<sup>6</sup> Zdroj: Dostupný na WWW <http://msdn.microsoft.com/en-us/library/bb399365.aspx>





**Obrázek 8 - Princip práce Entity Framework<sup>7</sup>**

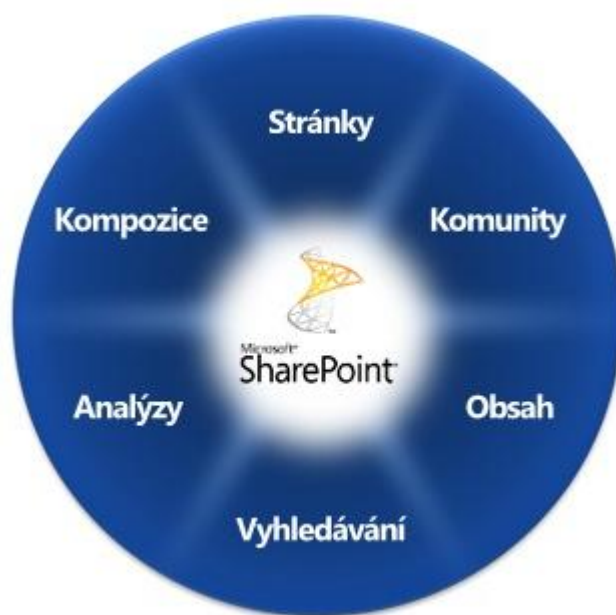
Entity Framework ve spojení s LINQ jsou velmi užitečné nástroje pro napsání datové vrstvy. Tímto se kód stává kratším a přehlednějším. Příklad, jak Entity Framework funguje od aplikace (Application) přes O/R mapování (ORM interface) v aplikaci až po dotazování (Queries) a aktualizace dat (Updates) v datovém uložišti (Data Store), přes ADO.NET Provider je zobrazeno na obrázku č. 8.

## 7.8 Sharepoint

Sharepoint je to platforma, která existuje od roku 2001 a slouží k tvorbě intranetových a internetových stránek. Dále má za úkol[8]:

- usnadnit práci mezi lidmi a pracovními týmy,
- zajistit správu dokumentů a webového obsahu,
- umožnit přístup k informacím ,
- zajistit sdílení znalostí,
- umožnit vývoj dalších aplikací.

<sup>7</sup> Zdroj: Dostupný na WWW <http://msdn.microsoft.com/en-us/data/aa937709>



Obrázek 9 - Rozdělení Sharepoint technologie<sup>8</sup>

Jak je vidět na obrázku č. 9, tak technologie Sharepoint je rozdělená do šesti základních sekcí[8]:

- **Stránky** - Je to prezentační prostředí, pomocí kterého jsou informace zpřístupňovány uživatelům. Informace mohou být různého typu od textu, grafů či video prezentace.
- **Komunity** - slouží pro týmovou spolupráci a umožňují uživatelům spolupracovat na dokumentech, úkolech, kontaktech, událostech a dalších informacích.
- **Obsah** – Může být v podobě textu, grafů, videa. Je to tedy obsah, který potřebujeme uživatelům předávat. U toho obsahu je zajištěna archivace a verzování dokumentů.
- **Vyhledávání** – Je to silná stránka Sharepointu, která nám umožňuje vyhledávání osob a informací, a to nejenom v oblasti Sharepoint, ale i v souborech na discích, databázi, podnikových aplikacích či nastavených webových stránkách. V rámci Sharepointu je podporováno fultextové vyhledávání v dokumentech.
- **Analýzy** - Uživatelé mohou využít i speciální nástroje Business Intelligence pro prezentaci dat, které jsou důležité pro rozhodování.
- **Kompozice** – Slouží pro další rozšíření vytvářením webových aplikací, nástrojů. To lze například pomocí Sharepoint designér, který je k dispozici zdarma.

<sup>8</sup> Zdroj: Dostupný na WWW <http://www.microsoft.com/cze/sharepoint/2010/>

## 7.9 Hermes Framework

Framework zajišťuje snadnou integraci webové aplikace do prostředí Microsoft Office Sharepoint Server 2007 / 2010.

Framework byl navržen s co největší snahou o budoucí rozšiřitelnost a spravovatelnost a zároveň se snahou co nejméně omezovat vývojáře při vývoji aplikací tak, aby každá společnost nebo vývojář mohl používat nástroje, postupy a pravidla, na které je zvyklý.

Cílem frameworku je zajistit snadnou integraci aplikací do prostředí Microsoft Office Sharepoint Server 2007 / 2010 a zajistit jednotný způsob ověřování, logování a auditu.

Stručný přehled důvodů a výhod Hermes frameworku:

- Automaticky zpřístupňuje další komponenty pro ověřování práv, logování, ...,
- Implicitně ověřuje uživatele při přístupu ke stránce,
- Hermes striktně nedefinuje strukturu, ale pokud jsou dodržena obecně stanovená pravidla použití, je vývoj rychlejší, nároky na údržbu nižší a aplikace mají obecně vyšší výkon,
- Vývoj aplikace bez nutnosti prostředí MOSS/WSS,
- Snadná správa a analýza případných problémů,
- Podpora jednotného a přesně definovaného způsobu nasazení těchto aplikací,
- Podpora jednotného logování.

## 8 Uživatelská část

V aplikaci se nacházejí tři základní role, a to administrátor, lektor a student. Administrátor zde slouží hlavně jako správce kurzů, které může vytvářet a mazat pouze on a kde také přiřadí příslušným uživatelům role lektora a studenta. V případě administrace testu v kurzu by si musel přiřadit roli lektora. Lektor má v aplikaci za úkol spravovat testy, komunikovat se studentem a vlastně řídit výuku. Nakonec se zde nachází role studenta, která může čerpat studijní materiály z kurzu, komunikovat s lektorem a vykonávat testy.

V dalších kapitolách se nachází ukázky stránek z aplikace a popis co se na jednotlivých stránkách nachází.

### 8.1 Hlavní strana

Tato stránka se nám zobrazí při načtení stránky, kde jsem přihlášen jako administrátor. Díky tomu jsou zobrazena u každého kurzu ikony tužky pro úpravu a křížky pro smazání kurzu. Ukázku můžete vidět na obrázku č. 10. Jako u jediné, zde zobrazím hlavičku a patičku stránky, kvůli velikosti obrázků. Zde je vidět seznam kurzů, kde na každém řádku mohou být maximálně dva.



Obrázek 10 - ukázka úvodní stránky aplikace

## 8.2 Úprava detailu kurzu

Tato stránka slouží k úpravě základních informací o kurzu, kde v levé části je vidět aktivní menu pro úpravu detailu neboli základních informací. Toto je vidět na obrázku č. 11, kde se ještě nachází položky uživatel, testy a studijní materiály.

Hlavní stránka > Informační a komunikační technologie

### Detail kurzu

Název:

Úvodní obrázek: 

Popis:

Aktivní od:    Aktivní do:

Publikován od:    Publikován do:

Obrázek č. 11 - ukázka stránky pro úpravu základních informací kurzu

## 8.3 Editace uživatelů v kurzu

Další položkou menu jsou uživatelé, kde v pravé části jsou dvě tabulky. Jedna je pro přidání studenta a druhá pro přidání lektora do kurzu. Nad oběma seznamy se nachází textové pole, kam se napíše přihlašovací jméno uživatele pro přidání. To vše je vidět na obrázku č. 12.

Hlavní stránka > Informační a komunikační technologie

Přidat studenta:   | Login | Jmeno |  | Login | Jmeno |  |
| --- | --- | --- | --- | --- | --- |
| xperiensis\Student1 | Jiří Hrdlička | ✖ | xperiensis\ExamAdmin | Jan Pavelec | ✖ |
| xperiensis\Student2 | Jan Molnár | ✖ | xperiensis\Lektor1 | Karel Petřík | ✖ |
|  |  |  | xperiensis\Lektor2 | Pavel Vojtěch | ✖ |

Obrázek č. 12 - ukázka stránky pro správu uživatelů v kurzu

## 8.4 Seznam testů v kurzu

Při kliknutí na položku v menu testy, se nám v pravé části načte tabulka s testy vytvořená pro daný kurz. V tabulce je uveden název, typ a od kdy do kdy je daný test platný, neboli přístupný studentovi ke spuštění. Dále je zde sloupec operace, kde jsou ikony zastupující operace, co můžeme s testem provádět. Jak je vidět na obrázku č. 13.

Test	Typ testu	Platný od	Platný do	Operace
Test ICT	Zkušební	0:00 - 17.04.2012	0:00 - 01.05.2012	

Obrázek č. 13 - ukázka stránky pro správu testů v kurzu

## 8.5 Správa studijních materiálů

Poslední položkou menu jsou studijní materiály, zde v pravé části můžeme vidět hromadný upload neboli hromadné nahrávání studijních materiálů. Pod tím nahráváním se nachází tabulka s již nahranými materiály. Ty můžeme vidět na obrázku č. 14, kde je ikona znázorňující typ materiálu, odkaz na stažení, případný popis nebo kdy byl naposled upraven.

Typ	Název	Popis	Změněno
	IT_ICT_2012.pptx		16:07 - 17.04.2012
	Znackovaci_jazyky_2012.pptx		20:02 - 18.04.2012

Obrázek č. 14 - ukázka stránky pro správu studijních materiálů v kurzu

## 8.6 Správa testu

Tato stránka se velmi podobná úpravě detailu kurzu, s kterou také mají stejnou první položku v menu detail a podobný obsah, jen vztažený k vlastnostem daného testu. Proto je položka detail na stránce pro správu testů přeskočena. Na obrázku č. 15 je vidět tabulka se všemi otázkami v testu seřazené podle přiděleného pořadí. Každá položka, neboli otázka, obsahuje název, popis, počet bodů a odpovědi k ní vztažené. Dále je zde vidět ikona zelené fajfky u sloupce, která znázorňuje, jestli může být vyhodnocena automaticky. Pokud by tomu tak nebylo, byl by zde křížek. Zbývá nám zde sloupec pořadí,

kde se nacházejí červené šipky pro změnu pořadí otázky. Toto pořadí lze změnit i přetáhnutím položky, jak je vidět na obrázku. Podle pořadí jsou pak otázky načteny v testu.

Název	Popis	Bodů	Automaticky	Odpovědí	Pořadí
jaký je správný zápis syntace JavaScriptu pro výpis textu "hello world"?	vyberte jednu odpověď	20	✓	3	▲ ▼ ✕
označte pravdivé tvrzení	vyberte jednu odpověď	20	✓	3	▲ ▼ ✕
grafické formáty jsou:	vyberte 2 odpovědi	20	✓	4	▲ ▼ ✕
označte pravdivé tvrzení	vyberte alespoň jednu odpověď	20	✓	4	▲ ▼ ✕
jaký je správný zápis syntace JavaScriptu pro výpis textu "hello world"?	vyberte jednu odpověď	20	✓	3	▲ ▼ ✕

Obrázek č. 15 - ukázka stránky se seznamem otázek v testu

## 8.7 Editace otázky v testu

Na obrázku č. 16 je vidět stránka pro úpravu otázek vztahujících se k testu. Stránka je rozdělena na dvě části. V levé části vidíme informace jako je název, popis, bodové ohodnocení a typ odpovědi. Pokud se zvolíme typ `RadioButton` nebo `CheckBox`, tak se v pravé části zobrazí tabulka pro odpovědi. V případě, že zvolíme typ `Text`, tak se tabulka schová a v levé části se zobrazí ještě 2 pole. Tyto pole slouží k omezení velikosti psané odpovědi a znázorňují minimální a maximální počet znaků, kolik odpověď může obsahovat. Pokud odpověď nemá být nijak omezoována, stačí tyto pole nevyplnit.

Hlavní stránka > Informační a komunikační technologie > Test ICT > jaký je správný zápis syntace JavaScriptu pro výpis textu "hello world"?

**Detail otázky**

Název: jaký je správný zápis syntace JavaScriptu pro výpis textu

Popis: vyberte jednu odpověď

Bodů: 20 Typ odpovědi: RadioButton Uložit

Odpověď	Popis	Procent z bodů	Správná

Body: 0% Správná:

Vložit Zrušit

document.write("hello world")		100%	✓	✕
white.document="hello world"		0%	✕	✕
new text hello world.display		0%	✕	✕

Obrázek č. 16 - ukázka úpravy otázky v testu

## 8.8 Detail kurzu

Tato stránka se zobrazí, pokud kliknete na hlavní stránce na název některého kurzu, jak je vidět na obrázku č. 17. Vlevo nahoře je vidět panel, kde se nachází název kurzu, napravo vedle něj se nachází panel s informacemi o uživateli a odkaz na odhlášení z aplikace. Pod panelem s informacemi o uživateli je panel, kde je odkaz, pomocí kterého je možné se dostat na stránku s dotazy na lektora. Pokud máme od lektora odpověď, zobrazí se odkaz tučně a v závorce za ním je počet nepřečtených odpovědí.

Dále se na stránce nachází panel s informacemi o kurzu, pod kterým se nachází seznam testů a seznam studijních materiálů. Jak je vidět, u testu není tlačítko přidat a v operacích je jen ikona pro detail. Důvod je, že je přihlášen uživatel s rolí studenta.

Hlavní stránka > Informační a komunikační technologie

**Informační a komunikační technologie**

**Informace o uživateli**  
Uživatel: Jiří Hrdlička  
Role: Student  
Odhlásit

Dotaz na lektora (1)

**Popis kurzu:** Cílem je rozvinout znalosti studentů o nových informačních a komunikačních technologiích.  
**Platný od:** 0:00 - 17.04.2012      **Platný do:** 0:00 - 29.05.2012  
**Počet studijních dokumentů:** 2      **Počet Testů:** 1

**Testy:**

Test	Typ testu	Platný od	Platný do	Operace
Test ICT	Zkušební	0:00 - 17.04.2012	0:00 - 01.05.2012	

**Studijní materiály:**

Typ	Název	Popis
	IT_ICT_2012.pptx	
	Znackovaci_jazyky_20...	

Obrázek č. 17 - ukázka stránky detailu kurzu

## 8.9 Detail testu

Tato stránka je velmi podobná předešlé stránce detailu kurzu a je vidět na obrázku č. 18. Hlavním rozdílem je, že v popisu testu se nachází velké tlačítko v podobě zeleného trojúhelníku pro spuštění testu. Toto tlačítko je nahrazeno oranžovou šipkou zpět, pokud student má spuštěný test, který neukončil a testu neuběhla jeho doba trvání. Dále pod popisem testu je tabulka s již vykonanými testy. Nad touto tabulkou je filtr v podobě přepínače, kde si můžeme nechat zobrazit možnosti, jako jsou všechny testy, nejnovějších 10, vyhodnocené, nebo nevyhodnocené testy. U každého vyhodnoceného testu je ikona lupy pro zobrazení detailu výsledku.



Hlavní stránka > Informační a komunikační technologie > Test ICT

## Test ICT

**Informace o uživateli**

**Uživatel:** Jiří Hrdlička  
**Role:** Student

[Odhlásit](#)

[Dotaz na lektora \(1\)](#)

**Popis testu:**

**Instrukce:**

**Parametry testu:**

- U otázek v testu může být 1 a více správných odpovědí.
- Během testu nemůžete měnit odpovědi, již odeslaných otázek.
- V testu je nutné odpovědět na všechny otázky.

**Způsob hodnocení:** Test se po dokončení automaticky vyhodnotí a zobrazí výsledky.

**Typ testu:** Zkušební

**Délka testu:** 5 min.

**Platný od:** 0:00 - 17.04.2012

**Platný do:** 0:00 - 01.05.2012

**Maximální počet spuštění:** Neomezeno

**Celkový počet spuštění:** 5

**Počet otázek:** 4

[Spustit test](#)

**Filtr**

Všechny
  Nejnovější
  Vyhodnocené
  Nevyhodnocené

Student	Typ testu	Dokončil	Počet otázek	Stav	Operace
Jiří Hrdlička	Zkušební	20:23 - 18.04.2012	4	✓	
Jiří Hrdlička	Zkušební	19:55 - 18.04.2012	4	✓	
Jiří Hrdlička	Zkušební	18:22 - 17.04.2012	4	✓	
Jiří Hrdlička	Zkušební	18:14 - 17.04.2012	4	✗	
Jiří Hrdlička	Zkušební	18:12 - 17.04.2012	4	✗	

Obrázek č. 18 - ukázka stránky detailu testu

## 8.10 Průběh testu

Na obrázku č. 19 je vidět příklad, jak vypadá vykonávání testu. Ve vrchním panelu se nacházejí informace o názvu kurzu, ve kterém se nachází test a i samozřejmě název tohoto testu. Dále je zde vidět začátek a předpokládaný konec testu, s tím je spojený i zbývající čas, který je na pravé části pod odkazy pro nápovědu a předčasnému ukončení testu. Ve spodním panelu se pak nacházejí otázky, kde v titulku panelu je samotná otázka a popis se pak nachází nad samotnými odpověďmi. Ve spodní části se objevují tlačítka další, předchozí a k odeslání výsledků. To záleží na nastavení testu, u kterého můžeme zvolit, jestli student se může vrátit k již zodpovězené otázce, či jestli musí zodpovědět všechny otázky, aniž by nějakou přeskočil.

**Kurz:** Informační a komunikační technologie      **Test:** Test ICT      Nápověda

**Začátek:** 14:42 - 22.04.2012      **Konec:** 14:47 - 22.04.2012      Nedokončit test

Zbývající čas: 00:03:25

**1. označte pravdivé tvrzení**

**Popis:** vyberte jednu odpověď

značka <meta> může obsahovat klíčová slova  
 značka <meta> se obvykle používá k připojení CSS souboru  
 značkou <meta> se dělí dokument HTML na 3 základní části

[Další](#)

Obrázek č. 19 - ukázka průběhu testu

## 8.11 Vyhodnocení testu

Na vrchní části stránky se nacházejí informace v podobě tří bloků o názvu kurzu, informacích o uživateli a odkazu na dotazy. Pod těmito bloky se nalevo nachází výpis otázek s označenými odpověďmi. Napravo od tohoto výpisu otázek jsou souhrnné informace o testu, jestli student prošel, kolik měl bodů apod.

Stránka slouží jak pro lektora, tak i pro studenta. Oběma se zde zobrazí výpis otázek s odpověďmi. S tím rozdílem, že pokud je v testu otázka typu text nebo je u testu nastaveno vyhodnotit lektorem, tak lektorovi se u otázek zobrazí ikona tužky. Jak je vidět na obrázku č. 20 u první otázky, tak po kliknutí na ikonu tužky se u otázky zobrazí textové pole pro případný komentář odpovědi. Dál se místo ikony tužky zobrazí ikona pro uložení a zrušení editace otázky bez uložení. Samozřejmě je zde také číselné pole pro ohodnocení odpovědi.

Hlavní stránka > Informační a komunikační technologie > Test ICT > Hodnocení

### Test ICT

**Informace o uživateli**  
Uživatel: Jan Pavelec  
Role: Lector  
Odhlásit

**Dotazy od studentů (1)**

**1. označte pravdivé tvrzení (vyberte jednu odpověď)** 20 / 20

A. značka <meta> může obsahovat klíčová slova  
B. značka <meta> se obvykle používá k připojení CSS souboru  
C. značkou <meta> se dělí dokument HTML na 3 základní části

**2. Napište správný zápis nadpisu 2. řádu "toto je nadpis 2. řádu"**

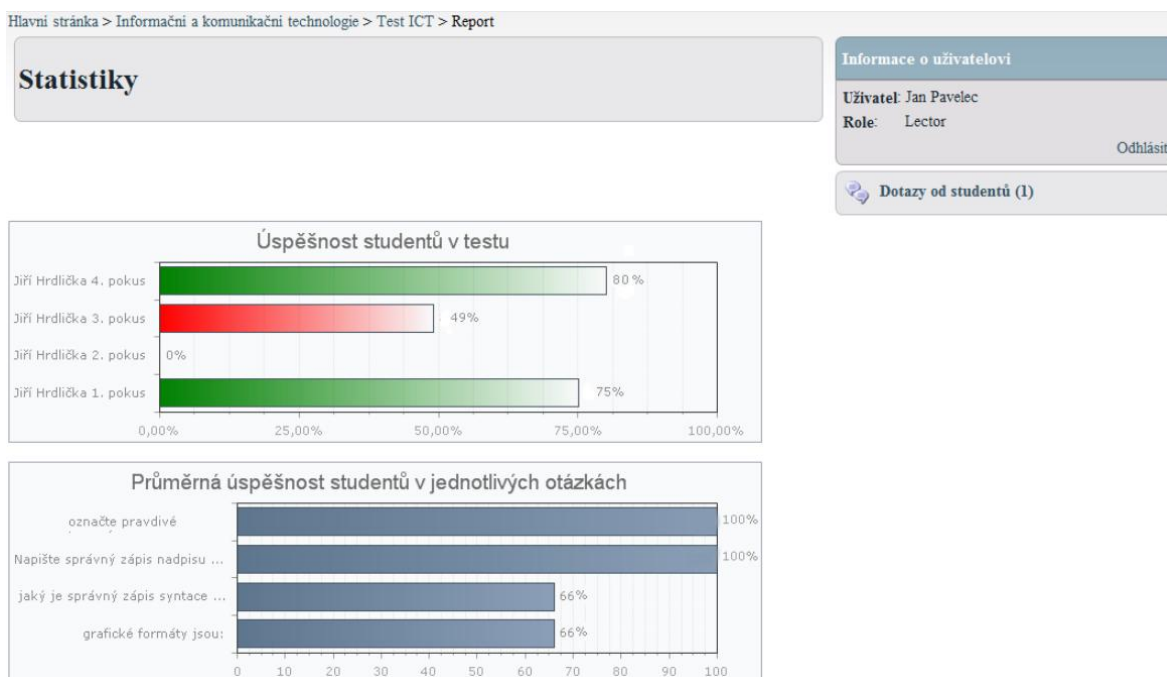
**Odpověď:** <h2>toto je nadpis 2. řádu</h2>  
**Komentář:**

**Student:** Jiří Hrdlička (xperiensis Student1)  
**Popis testu:**  
**Typ testu:** Zkušební  
**Délka testu:** 5 min.  
**Začátek testu:** 21:11 - 22.04.2012  
**Ukončen:** 21:12 - 22.04.2012  
**Počet otázek:** 5/5  
**Minimum bodů:** 57  
**Maximum bodů:** 95  
**Vyhodnocen:** Ne  
**Výsledek:**  
**Získaný počet bodů:** 60  
**Komentář:**

Obrázek č. 20 - ukázka vyhodnocení testu lektorem

## 8.12 Statistiky testu

Na této stránce se nacházejí statistiky dvojího druhu. První je úspěšnost studentů v testu s ohledem i o kolikátý pokus se jedná. V případě úspěšného vykonání se sloupec zbarví do zelené barvy. V opačném případě je sloupec grafu červený. Ve druhém grafu je průměrná úspěšnost studentů v jednotlivých otázkách. To vše je vidět na obrázku č. 21.



Obrázek č. 21 - ukázka statistik testů

## 8.13 Dotazy na lektora

Na některých výše popsaných stránkách byl vidět dotaz na lektora nebo případně dotazy od studentů, když byl přihlášen lektor. Na obrázku č. 22 je vidět, jak vypadá stránka pro dotazy na lektora, na které je zrovna přihlášen lektor. V levé části se nachází tabulka s dotazy od studentů, kde je uvedeno studentovo jméno, přihlašovací jméno a kdy student dotaz zaslal. Dále v nejpravější části tabulky se nachází ikona, jestli již lektor odpověděl na dotaz. Pokud odpověděl, zobrazí se zelená fajfka. V tabulce se může nacházet ještě jedna ikona a to v podobě žlutého trojúhelníku s vykřičníkem. Ten se nachází vedle jména studenta a vyjadřuje, jestli se v daném dotazu objevila odpověď, která přihlášeným uživatelem nebyla shlédnuta.

V pravé části se nachází dotaz od studenta podle aktivního řádku v tabulce dotazů. Dotaz v sobě obsahuje kurz a případně i test, kterého se týká. Pod tímto dotazem je textové pole, kam student píše dotazy a lektor na ně odpovídá. Může tedy zde vzniknout i delší konverzace.

Hlavní stránka > Informační a komunikační technologie > Dotazy a odpovědi

### Dotazy studentů

[+ Přidat](#) [Refresh](#)

Student	Přihlašovací jméno	Zasláno	Odpovězeno
Jiří Hrdlička	xperiensis\Student1	9:00 - 15.04.2012	✘
⚠ Jan Molnár	xperiensis\Student2	9:00 - 14.04.2012	✘

**Student:** Jiří Hrdlička      **Zasláno:** 9:00 - 15.04.2012  
**Kurz:** Školení SQL      **Test:** Test z DML příkazu1  
**Dotaz:** Dobrý den, chtěl bych se zeptat jestli, se bude v testu vyskytovat otázky z materiálu IT ICT\_2012.pptx

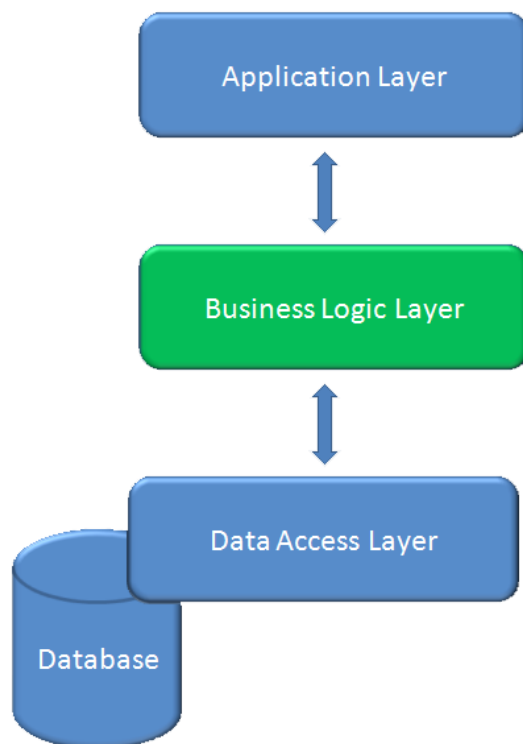
**Odpověď:**

[Odeslat](#)

Obrázek č. 22 - ukázka stránka pro dotazy na lektora

## 9 Implementace

Aplikaci je navržena podle návrhového vzoru MVC (model, view, controller), která má za úkol, aby oddělil prezentační, někdy nazývanou jako aplikační vrstvu (view), od datové vrstvy (model). Provede se to vložím další vrstvy a to controller, kde se provádí business logika aplikace.



**Obrázek č. 23 - Návrhový vzor aplikace<sup>9</sup>**

Na obrázku č. 23 je vidět návrhový vzor aplikace, kde application layer představuje aplikační vrstvu, která slouží pro zobrazení uživateli. Veškerá interakce aplikace s uživateli probíhá přes tuto vrstvu. S další vrstvou je spojená business logika, na obrázku znázorněná jako business logic layer a zastupuje v MVC modelu controller. V této vrstvě se data, která jdou z nebo do databáze upravují podle potřeby. Nakonec datová část, neboli model, který je na obrázku zastoupen data access layer. Tato vrstva slouží pouze k získávání a ukládání dat například do databáze. Zde se nenachází žádná logika.

<sup>9</sup> Zdroj: Dostupný na WWW [http://lepoque.blogspot.com/2010/05/layered-architecture-for-aspnet\\_03.html](http://lepoque.blogspot.com/2010/05/layered-architecture-for-aspnet_03.html)

## 9.1 Application layer

Aplikace je postavena na Hermes Frameworku, který je zmíněn v kapitole 7.7, slouží pro integraci do Microsoft Office Sharepoint. V aplikaci je tato vrstva implementována pomocí samostatného projektu s názvem *Elearning* a jeho struktura je popsána v následující podkapitole:

### 9.1.1 Popis struktury webové části

#### Root:

**.config** – Soubory s touto příponou slouží pro konfiguraci v podobě XML. Výhodou těchto souborů je, že můžeme změnit nastavení aplikace tzv. za běhu. Aplikace obsahuje základní konfigurační soubor *Web.config*, ve kterém se nachází například způsob autentifikace, uložené tag prefixi a odkaz na další 2 konfigurační soubory. Prvním souborem je *hermes\_1.3.4.0.config*, kde se nachází nastavení Hermes Frameworku. V tomto souboru jsou uloženy cesty pro načtení kaskádových stylů, nebo JavaScriptové knihovny. Pomocí druhého souboru *ConnectionString.config* můžeme jednoduše nastavit aplikaci připojení na konkrétní databázi.

**.aspx** – V kořenové složce se nacházejí dva tyto soubory *Default.aspx* a *DefaultFull.aspx*. Do těchto souborů se načítají veškeré stránky odvozené od `BaseHermesUserControl`. Přičemž jak pojmenování naznačuje, může být zobrazen na celou stránku pomocí *DefaultFull.aspx* nebo v určitém bloku *Default.aspx*. To se především využije, pokud aplikaci nasadíme do prostředí Microsoft Sharepoint.

**Key.snk** – Slouží k zabezpečení aplikace, kdy po podepsání aplikace dostane sestavení silný název a ten ztěžuje útočnickům například vložit do aplikace nebezpečný kód.

**RadEditorTools.xml** – Tento soubor využívá třída `RadEditor`, která slouží podobně jako `TextArea`, ale máme k dispozici nástroje pro úpravu textu, jako je tučné písmo, kurzíva, apod. Tyto nástroje jsou specifikovány právě v tomto souboru.

**ElearningCommon** – Je to třída kde se nacházejí konstanty jmen proměnných, které posíláme parametrem. Názvy všech user controlů<sup>10</sup>, používající se při navigaci na jinou stránku. Dále se zde nachází názvy ikon a k nim je uložena cesta kde se nachází na disku. Tato třída slouží pro usnadnění a k lepší přehlednosti kódu. Příkladem může být změna adresářové struktury, kdy v aplikaci nám stačí změnit pouze cesty v této třídě.

#### **\_masterPages:**

Tato složka obsahuje soubory pro určení vzhledu hlavní stránky, která se využívá jen při nasazení do Microsoft Office Sharepoint. Zde nebyly prováděny žádné změny.

#### **App\_Data:**

Zde se nacházejí dva soubory *portalSecurityDB.xml* a *SPActiveDirectory.xml*, které jsou potom po nasazení nahrazeny buď tabulkou v databázi, nebo nastavením doménových účtů v Microsoft Office Sharepoint.

V souboru *portalSecurityDB.xml* se nacházejí jednak informace o aplikaci, ale i popis všech základních user controlů, kde se u nich uchovává informace, ke které aplikaci patří, cesta k němu a i příznak, jestli má být načten jako první pro spuštění aplikace. Dále se zde nachází seznam skupin a přiřazení těchto skupin k jednotlivým user controlům.

Druhý soubor *SPActiveDirectory.xml* obsahuje seznam povolených uživatelů, seznam skupin a přiřazení uživatelů do těchto skupin.

#### **Handlers:**

V aplikační vrstvě je pouze jeden handler, který slouží ke stáhnutí studijních materiálů. Tento handler je implementací interfacu `IHermesAppHandler`, díky kterému můžeme pracovat s HTTP protokolem a máme přístup k `ApplicationContext` dané aplikace. To znamená, že můžeme přistupovat ke konfiguraci aplikace, logování, atd. Každý handler odvozený od zmíněného interface má hlavní metodu `Process`.

Do uvedené metody `Hermes Framework` předá jednak `HttpContext`, stejně jako v případě běžného HTTP handleru, zároveň ale předá i kolekci naplněnou parametry. Aby byl `Hermes Framework` schopný nalézt implementaci `DownloadHandler`, je potřeba jej zaregistrovat pod konkrétním jménem v unity kontejneru. Např. takto:

---

<sup>10</sup> Je to označení pro stránku nebo část stránky v ASP.NET.

```

<configSections>
  <section name="hermes.unity"
type="Microsoft.Practices.Unity.Configuration.UnityConfigurationSection
, Microsoft.Practices.Unity.Configuration, Version=2.0.414.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
</configSections>

<hermes.unity>
  <alias alias="IHermesAppHandler"
type="CSCR.Hermes.Core.IHermesAppHandler, CSCR.Hermes.Core" />

  <container name="appContainer">
    <register name="DownloadHandler" type="IHermesAppHandler"
mapTo="Elearning.Handlers.DownloadHandler, Elearning, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=90520991ceaec1b9"/>
  </container>
  <container name="hermes.Container">
    <types />
  </container>
</hermes.unity>

```

### Helpers:

Zde se nachází třída statická `EvaluateHelper`, která slouží k vyhodnocení testu. Tato třída má pouze jednu metodu `Evaluate`, která má jako vstupní parametr ID instance testu některého studenta.

Metoda prochází všechny instance otázek, to jsou otázky zobrazené studentovi při testu. Při procházení se automaticky vyhodnocují zaškrťovací otázky typu `CheckBox` a `RadioButton` a to tak, že se podívá, jestli odpověď, kterou uživatel zaškrtnul, je správná. Jestliže jsou dvě nebo více odpovědí správně u typu otázky `CheckBox`, tak uživatel dostane body za zaškrtnuté správné odpovědi, i když nezaškrtnl úplně všechny správné. Nesmí však zaškrtnout ani jednu špatnou odpověď. V tom případě dostává z otázky nula bodů.

Pokud se v testu nenachází otázka typu `Text` a test nemá nastaven příznak vyhodnotit lektorem, tak se test vyhodnotí a toto hodnocení uzavře. V opačném případě se výsledek testu neuzavře a výsledku otázky se nastaví příznak `NeedCorection`. Uzavřít ho musí teprve lektor, který rozhodne, zda odpověď je správně a přidělí body.



**Images:**

Jak z názvu vyplývá, zde se nacházejí veškeré obrázky aplikační vrstvy. V této složce se nacházejí další tři složky:

- **icons** – složka obsahující ikony aplikace,
- **iconsFileType** – složka obsahující ikony jednotlivých typů studijních materiálů,
- **page** – složka obsahující všechny obrázky, které jsou potřeba k správnému načtení (X)HTML stránky.

**Js:**

Tato složka je určena pro soubory jazyka JavaScript a jiné skriptovací jazyky. Zde se nachází pouze soubor *jquery.min.js*, což je malá skriptová knihovna pro lepší interakci JavaScriptu s (X)HTML stránkou.

**Styles:**

Složka určená pro všechny soubory s kaskádovými styly na správné zobrazení stránky. Je zde soubor *Style.css*, který je společný pro všechny stránky v aplikaci. Dále jsou zde dva soubory *EditorContentAreaStylesGray.css* a *EditorContentAreaStyle.css*, ty slouží k úpravě komponenty `RadEditor` tak, aby vypadal jako obyčejný `TextBox`. Je to kvůli tomu, že tato komponenta umí automaticky escapovat případné nežádoucí znaky a je možné u ní zobrazit nástroj pro úpravu textu.

**UIL:**

Je to zkratka anglických slov user interface layer a uchová zde objekty, které se týkají aplikační vrstvy:

`AjaxManagerProxy` – Tato třída je odvozená od `RadAjaxManagerProxy`. V této třídě je upravena metoda `Dispose`, která se volá při dealokaci objektu. V metodě je přidáno odstranění nastavení ajaxu. Úprava byla provedena, aby všechny ajax prvky fungovaly správně. Protože pokud by na stránce zůstalo uložené ajax nastavení, mohlo by to nepříjemně ovlivnit chování jiné stránky.

`Tab` – Třída, která se používá při vytváření menu položek. Je implementována tak, aby mohla být serializována neboli uložena do `Viewstatu` stránky. Obsahuje text zastupující položku menu, název user controlu, který po kliknutí bude načten. Dále parametry pro další stránku nebo případnou ikonu zobrazenou u textu.

## Usecases:

Tato složka ne nejdůležitější z celé aplikační vrstvy, protože obsahuje všechny user controly pro zobrazení ve webovém prohlížeči. Základem je soubor s příponou `.ascx` ve které se nachází (X)HTML kód spolu s ovládacími prvky ASP.NET. K tomuto souboru je vždy přidělen soubor se stejným jménem s příponou `.cs`, kde se zachytávají všechny události vzniklé na `.ascx` stránce. Složku Usecases dělíme na:

**Root** – V kořenu složky Usecases se nachází třída `HernesUserControl`, která dědí od `BaseHermesUserControl`. Pro zjednodušení je tato třída v dalších kapitolách označována jako stránka, protože ji taky představuje. V této třídě jsou implementované vlastnosti, které se uloží do `ViewStatu`, pokud pro ně existuje hodnota v parametru předaného z předešlé stránky. Jsou zde uloženy například ID kurzu, testu, atd. Další třídou zde je `BaseUserControl`, která dědí od základní třídy pro zobrazení `UserControl` a kde implementují pouze jednu vlastnost s názvem `HermesControl` k získání `HernesUserControl` neboli základního user controlu, kde se nachází, abychom se dostali z tohoto user controlu k vlastnosti `ApplicationContext`.

**CustomField** – Zde se nacházejí ovládací prvky, které byly vytvořeny v rámci aplikace. Je zde například graficky upravené tlačítko nebo vlastní panel s grafickým pozadím.

Nový ovládací prvek tlačítko je implementováno ve třídě `ButtonWithImage`. Při zobrazení tohoto tlačítka je důležité mu předat obrázek a text, který bude zobrazen. To však nestačí k tomu, aby tlačítko mohlo být použito v různých user controlech. Muselo se tedy zajistit, aby klik na tlačítko v daném user controlu mohl být zachycen a obslužen. Z tohoto důvodu byl ve třídě tlačítka vytvořen delegát `OnClickCallback`, který je volán v obsluhující metodě kliku na obrázek a text. Při použití tlačítka v user controlu pak musíme zaregistrovat do vlastnosti tohoto delegáta metodu, která se provede při kliku na tlačítko.

Na obrázku č. 24 jsou vidět dvě tlačítka, uložit a zrušit, kde tlačítko uložit je ve stavu hover. To znamená, že se nad ním nachází kurzor myši.



Obrázek č. 24 - Ukázka vlastní implementace tlačítka

Jako v ASP. NET máme třídu `panel`, tak bylo potřeba vytvořit vlastní panel, kterému by se mohl nastavit vlastní vzhled a mohly se do něj vkládat další ovládací prvky. Proto byl vytvořen ovládací prvek `BlockContent`, který dědí z třídy `WebControl`. Hlavním důvodem vytvoření je, aby se neopakoval pořad stejný (X)HTML kód, který pro zobrazení vlastního panelu není krátký. Je to z důvodu, že tento panel má zaoblené rohy a má proměnnou výšku a šířku jak je vidět na obrázku č. 25. Dále je na obrázku vidět u jednoho bloku jinak zbarvené záhlaví. To se zobrazí v případě, že tomuto panelu se nastaví vlastnost `Title`.



Obrázek č. 25 - Ukázka vlastního grafického panelu

**MainUC** – V této složce se nacházejí třídy typu `HernesUserControl`, které v aplikaci představují jednotlivé stránky. Přejít z jedné stránky je implementováno funkcí `Navigate`, která má jako parametr název stránky a parametry, které chceme předat na další stránku. Dále na této stránce máme přístup k `ApplicationContext`.

**UserControl** – Tato složka vznikla, aby se v aplikaci neopakoval stejný kód. Nacházejí se zde samostatné user controly neboli části stránky, které je možné použít na více stránkách.

### 9.1.2 Návrh webové části

Nejprve je vysvětleno způsob načítání stránek. O to se zde stará Hermes Framework, který je popsán v kapitole 7.7. Základní user control neboli stránka, jak je uvedeno v kapitole 9.1.1 ve složce *root*, musí být třída, která je odvozená od třídy *HermesUserControl*. Tyto stránky mají prefix UC a máme v nich přístup k funkci *Navigace*, do které se vkládá název stránky a případně kolekce parametru, které budou předány načítané stránce. Stačí nám uvést název, protože cesta ke stránce je uložena v souboru *portalSecurityDB.xml*, ten je popsán kapitole 9.1.1 ve složce *App\_Data*.

Zmíněná základní stránka má v souboru s příponou *.ascx*, která je popsána v kapitole 9.1.1 ve složce *Usecases* uložen (X)HTML kód pro zobrazení celé stránky. To znamená, že obsahuje hlavičku, obsah a patičku stránky. Do tohoto kódu pak můžeme načíst jiné user controly, které jsou uloženy ve složce *UserControl* a popsány také v kapitole 9.1.1. V této složce se nachází hlavička a patička stránky jako samostatné user controly. Ty se načítají na každé základní stránce.

V dalších podkapitolách je popsána implementace stránek spolu s ukázkami jejich zobrazení. Zde popsané stránky jsou vysvětlené v kapitole 8, se kterou mají i stejné názvy podkapitol. V závorce se pak nachází název třídy, která stránku implementuje.

#### Hlavní strana (UCLessonPeview)

Na hlavní straně se nachází seznam kurzů vytvořen pomocí ovládacího prvku *Repeater*. *Repeater*, jak z názvu můžeme odvodit, se jedná se o jakýsi opakovač. Je to velmi mocný nástroj, pomocí kterého můžeme vytvořit kdejaký seznam a jiné struktury. Stačí vytvořit šablonu v tagu *ItemTemplate* v (X)HTML kódu jak má vypadat jedna položka. Poté v serverové části stránky se přiřadí tomuto ovládacímu prvku do vlastnosti *DataSource* položky seznamu, což v mém případě je seznam kurzů. Nakonec je potřeba implementovat metodu *OnItemDataBound*. Tato metoda se volá před načtením stránky a musí se v ní implementovat kód, který najde ovládací prvek v šabloně a přiřadí mu hodnotu námi zvolenou. Příkladem může být nadpis kurzu, kde se získá instance ovládacího prvku *LinkButton* a nastaví se mu text a URL adresa, kam má odkazovat.

### **Úprava detailu kurzu (UCManagmentLesson):**

Pro připomenutí je zde, že všechny parametry pro načtení user controlů na stránku se získávají z vlastnosti `HernesControl`, popsané v kapitole 9.1.1 ve složce *Usecases*. Rozdělení této stránky je pomocí user controlu `LeftMenuAndContent`. U user controlu `LeftMenuAndContent` již podle názvu je možno usoudit, že obsah stránky bude vertikálně rozdělen. Kde na levé části se nachází menu a na pravé části potřebný obsah.

Rozdělení této stránky je pomocí user controlu `LeftMenuAndContent`. Menu je zde vytvořeno pomocí ovládacího prvku `Repeater`, který byl popsán v minulé kapitole. Položky menu se získávají v metodě `GetMenuItems`, která podle toho v jaké základní stránce se user control `LeftMenuAndContent` nachází, vrátí příslušnou kolekci třídy `Tab`, popsané v kapitole 9.1.1 ve složce *UIL*. Po kliknutí na takovouto položku menu ve webovém prohlížeči, se nastaví vlastnosti `ActiveLeftMenu` název požadovaného user controlu pro načtení do pravé části pro obsah. Název se získá z argumentu položky menu, která je zastoupena ovládacím prvkem `LinkButton`. Tomu se nastaví také atribut `class` na `active` a tím změní svůj vzhled na aktivní.

Do pravé části, kde se nachází obsah, se pak načítají user controly podle vlastnosti `ActiveLeftMenu`. To se provádí v metodě `LoadContent`, kde se název user controlu doplní o cestu a je načten a přidán do panelu pro zobrazení. Nad tímto panelem se nachází další user control `BreadCrumbs`, co v češtině znamená drobký chleba. Tento user control používám na většině stránkách. Díky němu se uživatel dozví, kde se nachází a může také přejít o libovolnou úroveň výš. Pracuje samostatně. V jeho kódu je metoda, která zjistí, kde se nachází a podle toho se vygenerují odkazy. Tyto odkazy se taky vytvářejí pomocí ovládacího prvku `Repeater` a kolekci třídy `Tab`.

### **Editace uživatelů v kurzu (LessonUsers)**

Nacházejí se zde dvě tabulky, které jsou v user controlu `LessonUsers` a jsou implementovány pomocí ovládacího prvku `RadGrid` a v aplikaci je využíván velmi často. K tomuto ovládacímu prvku byl vytvořen a přiřazen kaskádový styl pro lepší grafické zobrazení. Data jsou pak získávána a ukládána pomocí třídy `UserBLL`. Způsob práce je nastíněn v kapitole 9.2.

Před vložením se kontroluje pomocí metody `GetUserInfoFromLogin` ve třídě `SecurityProvider`, jestli dané přihlašovací jméno je povoleno. Povolení je chápáno tak, jestli se nachází v souboru `SPActiveDirectory.xml`, ten je popsán v kapitole 9.1.1 ve složce `App_Data`. Dále se kontroluje, aby uživatel nemohl být v obou rolích. To by mělo za následek nepředvídatelné chování aplikace.

### **Seznam testů v kurzu (ExamList)**

Tabulka představující seznam testů je zde načtena z user controlu `ExamList`, kde se při vkládání jednotlivých testů generují jednotlivé ikony operací. To se provádí podle metody `IsLectorInLesson` ve třídě `LessonBLL`, které dáme jako parametr ID kurzu získané z parametru stránky a přihlašovací jméno uživatele. Pokud se uživatel nachází v kurzu jako lektor má přístupné všechny operace, které jsou vidět na obrázku č. 15. Pokud však ne, tak je mu povolena pouze operace zobrazení testu.

### **Správa studijních materiálů (LessonDocuments)**

K hromadnému nahrávání byl použit ovládací prvek `RadAsyncUpload`. Ten nám dovoluje zobrazit materiály, které jsou již na serveru nahrány pomocí zeleného puntíku. Pomocí žlutého puntíku nám zobrazuje ty, které právě nahrává nebo budou nahrávány. U tohoto ovládacího prvku lze nastavit typy souboru, které jsou zakázány.

Při stisku tlačítka nahrát se u každého souboru zjistí, jakou má příponu a podle této přípony se mu přiřadí příslušná ikona. Soubor se pak pomocí třídy `DocumentBLL` popsané v kapitole 9.2.2 nahraje do knihovny v Microsoft Office Sharepointu a informace o něm do databáze.

V tabulce se studijními materiály je vidět u první položky editační formulář pro přidání popisu. Ten se zobrazí, pokud se klikne na šedou tužku nacházející se nalevo od ikony typu položky. K tomu bylo za potřebí v ovládacím prvku `RadGrid` přidat `GridEditCommandColumn` pro zobrazení ikony tužky a implementovat vlastnost `EditItemTemplate` u položky popis. V případě kliknutí na šedou fajfku v této editaci pak odchytním v metodě `rgridDocuments_OnItemCommand` argument kliknutí `Update` a provedu zápis do databáze. Stejně tak zpracovávám kliknutím na červený křížek pro smazání. Šedý křížek slouží k zavření editace položky beze změn.

### **Správa testu (UCManagmentExam)**

Jak bylo zmíněno v kapitole 8.6, tak user control `ExamDetail` byl přeskočen a bude zde popsán user control `QuestionList`. Zde byla stěžejní část této stránky nebo spíše user controlu `QuestionList` zajistit, aby šlo u otázky změnit pořadí pouhým přetáhnutím. Toto je zajištěno nejjednodušším nastavením ovládacího prvku `RadGrid`. Dále byl zde použit ovládací prvek `AjaxManagerProxy`, který je popsán v kapitole 9.1.1 ve složce `UIL`. U tohoto prvku se zaregistruje ovládací prvek, který vyvolává znovu načtení stránky a k němu ovládací prvky, které se mají znovu načíst. Byla tedy zaregistrována tabulka, aby při její manipulaci se načetla pouze tato tabulka. To velice urychluje práci uživatele, protože nemusí čekat, až se načte celá stránka s kaskádovými styly a `JavaScripty`.

Nejprve bylo důležité nastavit na straně uživatele neboli klientském nastavení povolení `AllowRowsDragDrop` a `AllowRowSelect` a zakázání `EnableDragToSelectRows`. K tomu, aby server mohl na akci přetažení otázky zareagovat, byla vytvořena metoda `rgridQuestion_OnRowDrop`. Tato metoda při zachycení této události zkontroluje, jestli přetahujeme položku v rámci naší tabulky. V případě, že je toto splněno, musí se získat pořadí těchto položek. Na základě získaných pořadí se ve třídě `QuestionBBL` metodou `ReOrderQuestion` upraví pořadí otázek podle toho za nebo před jakou otázkou jsme položku přesunuli a uloží do databáze.

### **Editace otázky v testu (UCQuestionDetail)**

V této části bylo důležité zajistit, aby po zvolení typu odpovědi se okamžitě zobrazila správné ovládací prvky a ostatní skrýt. Proto byla u vyjíždějícího menu typu `RadComboBox`, kde se zobrazuje typ odpovědi, napsána metoda `rcmbTypesOfAnswer_OnSelectedIndexChanged`. V této metodě se podle právě zvolené položky v menu se provede zobrazení a schování ovládacích prvků. Toto menu je zaregistrované ve třídě `AjaxManagerProxy` a díky tomu se nemusí načítat celá stránka.

Dalším složitějším úkolem na této stránce bylo vytvořit formulář pro tabulku odpovědí k vkládání nových nebo úpravu existujících odpovědí. Tento formulář je vidět v pravé části stránky na obrázku č. 18 a je implementován jako samostatný user control `AnswDetail`. K tomuto user controlu byla předána cesta do vlastnosti `UserControlName` a nastaveno `EditFormType` na typ `WebUserControl`. Toto se

nastavuje v tabulce typu `RadGrid` v sekci `EditFormSettings`. Při tomto nastavení se mi formulář začal zobrazovat, ale nebylo možné na stránce odchytnout události vzniknuté na formuláři jako je například stisknutí tlačítka vložit. To bylo vyřešeno tak, že nejprve se vytvořily delegáty pro vložení a aktualizaci, které se volaly při vzniku události pro ně příslušící. Poté, v základní stránce, se při vytváření tohoto formuláře v metodě `OnItemCreated` získala instance formuláře a zaregistrovala u zmíněných delegátů metody jím příslušející. Poté na stránce vše fungovalo tak, jak má.

### **Detail kurzu (UCLessonSummary)**

Při vytváření této stránky byl brán ohled na to, že tři bloky, které se nacházejí nad popisem kurzu, se budou opakovat. Proto se pro ně vytvořil samostatný user control jménem `HeadingBlocks`, ten je popsán v kapitole 9.1.1 ve složce `CustomField`. Dolní části jsem pak načtl user control `ExamList` a vedle něj vytvořil tabulku se studijními materiály ke stáhnutí.

### **Detail testu (UCExamSummary)**

Při vytváření stránky se bral ohled na to, že je společná pro studenta i lektora. Proto, když se přihlásí lektor, tak v seznamu uvidí všechny vykonané testy a místo tlačítka pro spuštění je odkaz pro zobrazení statistik v podobě grafů.

Tabulka s vykonanými testy je implementována v usercontrolu `ExamsIterationList` spolu s filtrem nad ní. Při změně tohoto filtru se vyvolá znovunačtení tabulky, která se naplní podle zvoleného filtru. Tato tabulka je také určena pro obě role, proto student může vidět buď ikonu lupy pro detail, nebo v případě, že vykonával test, který s nějakých důvodů neukončil, zobrazí se mu zde ikona pro návrat do testu. To je umožněno tím, že každá implementace testu v sobě skrývá jak začátek testu, předpokládané ukončení a skutečné ukončení testu studentem. V případě přihlášeného lektora je zobrazena u instance testu buď ikona kalkulačky pro vyhodnocení testu. Po vyhodnocení pak u té implementace testu uvidí také lupu pro zobrazení detailu.

### **Průběh testu (UCExamOnGoing)**

V této části bylo asi nejdůležitější implementovat odečítání času na klientské straně. To je provedeno pomocí JavaScriptu, kde po načtení stránky se zavolá funkce `Down`, která si načte čas z ovládacího prvku `hfTimeLeft` typu `HiddenFieldu`, kde ve vlastnosti `Value` se nachází čas, který se bude sestupně odpočítávat. Tento čas se rozdělí do proměnných pro hodiny, minuty a vteřiny, které se popořadě snižují. Ke snížení se volá



funkce `DownRepeat`, která odečte jednu vteřinu a pomocí funkce `setTimeout` se znovu zavolá za jednu vteřinu. V případě, že všechny časové proměnné jsou nulové, vyskočí dialogové okno s informací, že vypršel čas. Poté se stránka přesměruje na vyhodnocení testu.

V dolním panelu je zaregistrován v ovládacím prvku `AjaxManagerProxy`, aby při změně otázky nedocházelo k načítání celé stránky. Do tohoto panelu jsem pak dynamicky načítal jednotlivé otázky podle pořadí. S tím byla spojena další věc, že lektor mohl zadat, aby se otázky zobrazovaly náhodně. K tomuto je vytvořena vlastnost `ListOrderQuestions` ukládající se do `ViewStatu` stránky, která byla typu `Dictionary`. Tato vlastnost měla jako klíč pořadí otázky a jako klíč její ID. V případě tedy, že otázky měly být v náhodném pořadí, stačilo zavolat vložit ID všech otázek do listu a ten zamíchat pomocí algoritmu `AsRandom` popsaného v kapitole 9.2.3. Tento list se pak vložil v cyklu od 1 do potu otázek do vlastnosti `ListOrderQuestions`. Když už bylo pořadí otázek určené, musela se vytvořit vlastnost, kde se bude uchovávat aktuální pořadí načtené otázky, a která se zvyšuje a snižuje podle stisknutí tlačítek další a předchozí. Tato vlastnost se hlavně používala jako klíč k vybrání ID otázky z vlastnosti `ListOrderQuestions`, která byla načtena na stránku.

Další část pro implementaci byla spojená s nastavením testu, kde lektor může zvolit, aby student musel odpovědět na všechny otázky. Muselo být tedy zajištěno, aby student nemohl přejít na další otázku, pokud neodpoví na aktuálně zobrazenou. To bylo zajištěno pomocí validátorů na klientské straně, které jsem generoval dynamicky spolu s otázkou.

U otázek typu `RadioButton` s jednou odpovědí správně stačilo vytvořit ovládací prvek typu `RequiredFieldValidator`, který nedovolí opustit stránku, pokud není zvolena žádná odpověď. Otázka typu `CheckBox`, neboli více možných odpovědí správně, byla složitější, kde se muselo vytvořit `CustomValidator`, kterému se nastavila vlastnost `ClientValidationFunction` na implementovanou metodu v klientské části `checkChceckboxList`. V této metodě se procházejí všechny odpovědi, a pokud nebyla ani jedna odpověď správně, nastavil jsem argument `IsValid` na `False`, který zabránil opuštění stránky.

Poslední otázka typu `Text`, se implementovala asi nejhůře. Muselo se totiž zajistit, že pokud lektor nevyplnil minimální a maximální délku odpovědi tak, aby se vytvořil pouze `RequiredFieldValidator`. Tento typ validátoru kontroloval jen, jestli student napsal nějaký znak. V opačném případě, kdy jsou vyplněny otázky omezení minimální a maximální délky, muselo se kontrolovat počet znaků. To se nakonec zajistilo pomocí ovládacího prvku `RegularExpressionValidator`, kterému, jak je poznat podle názvu, se muselo nastavit regulární výraz. Podle tohoto regulárního výrazu se kontrolovala textová odpověď.

### **Vyhodnocení testu (`UCExamEvaluate`)**

Při vytváření této stránky se pro výpis zvolil ovládací prvek `Repeater`, kterému se předaly otázky, které mají referenci na případné vyhodnocení otázky a reference na správné a studentem zvolené odpovědi. Poté, před načtením stránky, se zavolala metoda `repQuestions_OnItemDataBound` tohoto ovládacího prvku. V této metodě se pak dynamicky generovaly ovládací prvky, protože zde mohl být výčet nebo výpis textu napsané studentem.

Tak jako u jiných stránek zde bylo potřeba, aby při kliknutí na ikonu pro editaci se nenačítala celá stránka. Protože se zde otázky vytvářely dynamicky, tak místo dříve použitého `AjaxManagerProxy` byl použit ovládací prvek `UpdatePanel`. U tohoto panelu do vlastnosti `ContentTemplate` se vložil ovládací prvek `Repeater` s otázkami. Díky této kombinaci se při editaci otázky obnovovala jen ta část stránky v ovládacím prvku `UpdatePanel`. Na konci stránky se pak nachází tlačítko pro uzavření výsledku. Tlačítko na obrázku č. 20 není vidět, protože se při editaci otázky schválně schová. Důvod byl, aby lektor nezapomněl uložit hodnocení aktuálně upravované otázky.

### **Statistiky testu (`UCReportExamIteration`)**

Grafy na stránce byly vytvořeny pomocí ovládacího prvku `RadChart`. Oběma grafům jsem nastavil předdefinovaný skin `WebBlue` a hlavně orientaci grafu pomocí vlastnosti `seriesorientation` na `Horizontal`. Nejtěžší úlohou zde bylo co nejefektivněji získání dat a to ve formě, která by se dala rovnou použít. Zde bylo zvoleno získání dat pomocí jazyka LINQ.

Příkladem může být první graf, kde po získání dat se tato data vložila do grafu a nastavila se, že na y-ové ose se nachází výpisy studentů s číslem pokusu a na x-ové

souřadnici se vypisují procenta úspěšnosti studenta v daném testu. Dále byla ještě implementována metoda `rchartStudents_OnItemDataBound`, kde se zjistí u každého sloupce jeho procentuální hodnota a porovná se s hodnotou potřebnou k úspěšnému vykonání testu. Pokud je rovna či vyšší, nastaví se pozadí sloupce na zelenou barvu, pokud je nižší, tak barva pozadí sloupce je červená.

Dále je zde vidět jediná ukázka kódu. Tento kód slouží k získání průměrné procentuální úspěšnosti studentů v jednotlivých otázkách pro daný test. V kódu je vidět, že se data berou z tabulky otázek. Berou se v úvahu jen ty otázky, které se nachází v testu s ID rovno vstupnímu parametru `examId`. Tyto otázky se pak seskupí podle nadpisu otázky a vytvoří se nová anonymní proměnná s vlastností `key`, kde je nadpis otázky a vlastností `value`. Ve vlastnosti `value` se uloží výpočet procentuální úspěšnosti, kde se sečtou všechny výsledky otázek pro danou otázku a vydělí jejich počtem. Objevuje se zde přetypování a kontrola zda výsledek má hodnotu `null`. Je to vzhledem k tomu, že otázka nemusí mít žádnou instanci, což nastane, pokud nebylo na žádnou otázku odpovězeno. Nakonec se výsledek převede do kolekce typu slovník, kde se název otázky, neboli vlastnost `key`, dekoduje z případných zástupných HTML tagů do pouze textové podoby.

```
using (var context = new elearningDatabase())
{
    return (from s in context.Questions
            where s.Exam.ID == examId
            group s by s.Caption
            into g
            select new
                {
                    key = g.Key,
                    value = (int)((int?)g.Sum(s => (int)s.QuestionIterations.Sum(qi =>
(int?)qi.QuestionResults.PercCorrect ?? 0) / (s.QuestionIterations.Count != 0 ?
s.QuestionIterations.Count : 1)) ?? 0)
                }
            ).ToDictionary(x => HttpUtility.HtmlDecode(x.key), x => x.value);
}
```

### **Dotazy na lektora (UCResponseMessages)**

V první části tvorby této stránky bylo potřeba zajistit, aby se přihlášenému uživateli zobrazily dotazy pro něj určené nebo jím vytvořené. Například může nastat, že student může být v jiném kurzu i lektorem. Proto jsem vytvořil LINQ dotaz, kde vrátím všechny dotazy jím vytvořené a k tomu všechny dotazy z kurzu, kde vystupuje v roli lektora.

Jako další bylo potřeba zajistit, aby uživatel rozpoznal dotazy, u kterých je nová odpověď. Na obrázku č. 23 je to vidět v podobě žlutého trojúhelníku s vykřičníkem. To bylo zajištěno přidáním vlastnosti `IsNew` k entitám pro dotaz `Message` a odpověď `MessageResponse`. Tato vlastnost byla typu `bool` a nesla informaci, jestli ještě nebyla přečtená. Při vytvoření entity je tato vlastnost nastavena na `True`. Její přenastavení pak dochází při zobrazení uživatelem, který tuto entitu nevytvořil.

Nakonec bylo potřeba implementovat možnost vytváření nových dotazů. Tento formulář se zobrazí na pravé části při stisknutí tlačítka přidat nad tabulkou. Po odeslání dotazu se uloží do databáze a zobrazí jak v tabulce dotazů, tak i v pravé části, jak je vidět na obrázku č. 23. Pod tímto dotazem se pak pomocí ovládacího prvku `Repeater`, vypisují jednotlivé odpovědi. Načítání dotazů a zobrazení formuláře pro nový dotaz se zobrazuje pomocí ovládacího prvku `AjaxManager`, aby nedocházelo k načítání celé stránky.

## 9.2 Business logic layer

Tato část aplikace zprostředkovává komunikaci mezi datovou a aplikační vrstvou. Nachází se zde také logika rozhodování a manipulace s daty.

### 9.2.1 Root

Zde se nachází základní třída `BLL`, od které jsou všechny ostatní business třídy odvozeny. Základem této třídy je vlastnost, která zpřístupňuje providera dat. V aplikaci je implementován `EntityDataProvider` provider.

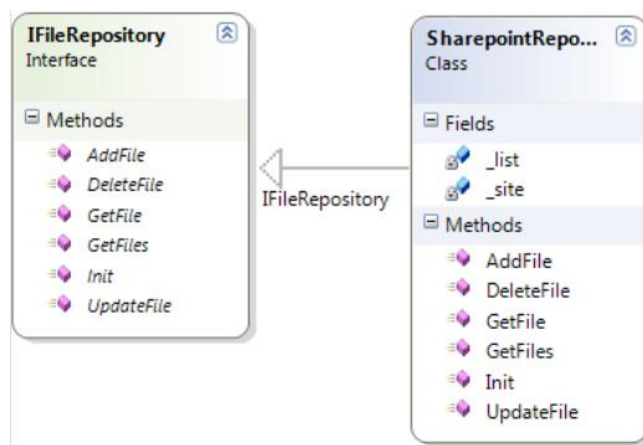
### 9.2.2 Entity BLL

Tato složka obsahuje třídy s logikou zpracování dat pro důležité entity v návrhu Entity Frameworku. Všechny tyto třídy mají společnou metodu `Save`, kde se rozhoduje, jestli daný objekt už v databázi existuje nebo se teprve musí vytvořit. Dále obsahují metody jak získat případný referenční klíč, pokud nemáme načtenou celou referenci. To se provádí pomocí třídy `EntityHelper`, která je implementovaná v datové vrstvě.

#### **Ukládání souborů:**

Ukládání souborů bylo vytvořeno tak, aby šlo jednoduše rozšířit a mohl být způsob ukládání jednoduše změněn. K tomu se vytvořila nová `Solution` a v ní projekt `IFileRepositories`. V tomto projektu se nachází interface téhož jména, kde je deklarace základních operací viz obrázek č. 26. Dále se v projektu nachází třída `RepositoryException`, která dědí od základní třídy pro výjimky `Exception`. Tato třída byla vytvořena, aby bylo možné v aplikaci odchytit výjimky vzniklé při manipulaci se soubory.

Dále se vytvořil další projekt, z interface `IFileRepository`, pro který se vytvořila implementace `SharepointRepository`. Tato implementace pracuje s Microsoft Sharepoint, kam se data mohou ukládat, získávat a mazat. Implementace má dvě základní proměnné a to `_site`, kde se nachází adresa Sharepointu a `_list`, kde je uložen název knihovny, kam se soubory uloží. Tyto dvě proměnné se nastaví při volání metody `Init`, jejíž parametrem je `Dictionary`, kde jsou uloženy všechny proměnné pro danou implementaci.



Obrázek 26 - Diagram tříd pro ukládání souborů

### DocumentBLL:

Příkladem může být třída `DocumentBLL`, která má za úkol správu souborů. Jak již bylo zmíněno, tak třída je podle návrhového vzoru Singleton. To má za následek to, když je poprvé volána tato třída, tak se teprve vytvoří její instance. Instance `SharepointRepository` pro ukládání dat do Sharepointu také vytvoří podle návrhového vzoru Singleton, ale použije se zde i návrhový vzor Dependency Injection.

Návrhový vzor Dependency Injection snižuje závislost mezi třídami. Znamená to, pokud bude potřeba v proměnné typu `IFileRepository` vytvořit instanci `SharepointRepository`, tak se to nebude psát napevno do kódu. Například třídy `DocumentBLL`, ale pouze se zavolá metoda `Resolve` s parametrem interface, pro který je potřeba vytvořit instance. Metoda `Resolve` je uložena v třídě `Container` v `Hermes Frameworku`. Aby toto vše fungovalo, musí být v konfiguračním souboru uveden interface a k němu daná implementace s cestou, kde se nachází, viz příklad:

```

<configuration>
  <hermes.unity>
    <alias alias="IFileRepository"
type="Xperiensis.common.IFileRepositories.IFileRepository,
Xperiensis.common.IFileRepositories, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=a4f6e85f24ffb0e9" />

    <container name="appContainer">
      <register type="IFileRepository"
mapTo="Xperiensis.common.SharepointRepositories.SharepointRepository,
Xperiensis.common.SharepointRepositories, Version=1.0.0.0,
Culture=neutral, PublicKeyToken=a4f6e85f24ffb0e9"/>
    </container>
  </hermes.unity>
</configuration>

```

### 9.2.3 Helpers

V této složce je implementovaná statická třída `RandomHelper`, která má za úkol rozšířit rozhraní `IEnumerable` o metodu `AsRandom`. Tato metoda slouží k náhodnému promíchání pořadí v prvku v dané kolekci. Provádí se to pomocí třídy `Random` a cyklu, ve kterém náhodně prohazuje prvky.

## 9.3 Data access layer

V datové vrstvě se nachází vše, co je spojené s vkládáním, mazáním a úpravou dat do databáze nebo jiného úložiště dat. V této aplikaci je to pomocí databáze s použitím `Entity Framework`.

### 9.3.1 Interface

Složka obsahuje interface `IDataProvider`, které obsahují deklaraci všech potřebných metod, které se v aplikaci používají. Z této třídy se vytvoří již určitá implementace ukládání dat do nějakého úložiště.

### 9.3.2 Data

Zde se nachází implementace interfacu `IDataProvider`, a to pomocí `Entity Frameworku`. Tato třída se jmenuje `EntityDataProvider` a dotazy do databáze jsou zde vytvářeny pomocí technologie `LINQ`. Další třídou, která se zde nachází, je `DataManager`. Ten je implementován jako návrhový vzor `Singleton` a prostřednictvím vlastnosti `Instance` nám vrací instanci třídy `EntityDataProvider`.

Ve třídě `EntityDataProvider` se dotazujeme do databáze, jak bylo zmíněno, a v případě, že nastane chyba, je odchycena v podobě výjimky. Tato výjimka se pak předá metodě `Error` v `logProvider`, která se nachází v `Hermes Frameworku`. Do jakého souboru se bude logovat je nastaveno v konfiguračním souboru `hermes_1.3.4.config` v sekci `loggingConfiguration`.

### 9.3.3 Entity

Složka obsahuje třídu `LessonWithCount`, která dědí od entity `Lesson` z návrhu `Entity Frameworku`. Rozšiřuje tuto třídu o vlastnosti, do kterých se ukládá počet studijních materiálů, počet studentů a počet testů vztahujících se k danému kurzu neboli `Lesson`. Třída byla implementována z důvodu snížení počtu dotazů do databáze.

### 9.3.4 Helpers

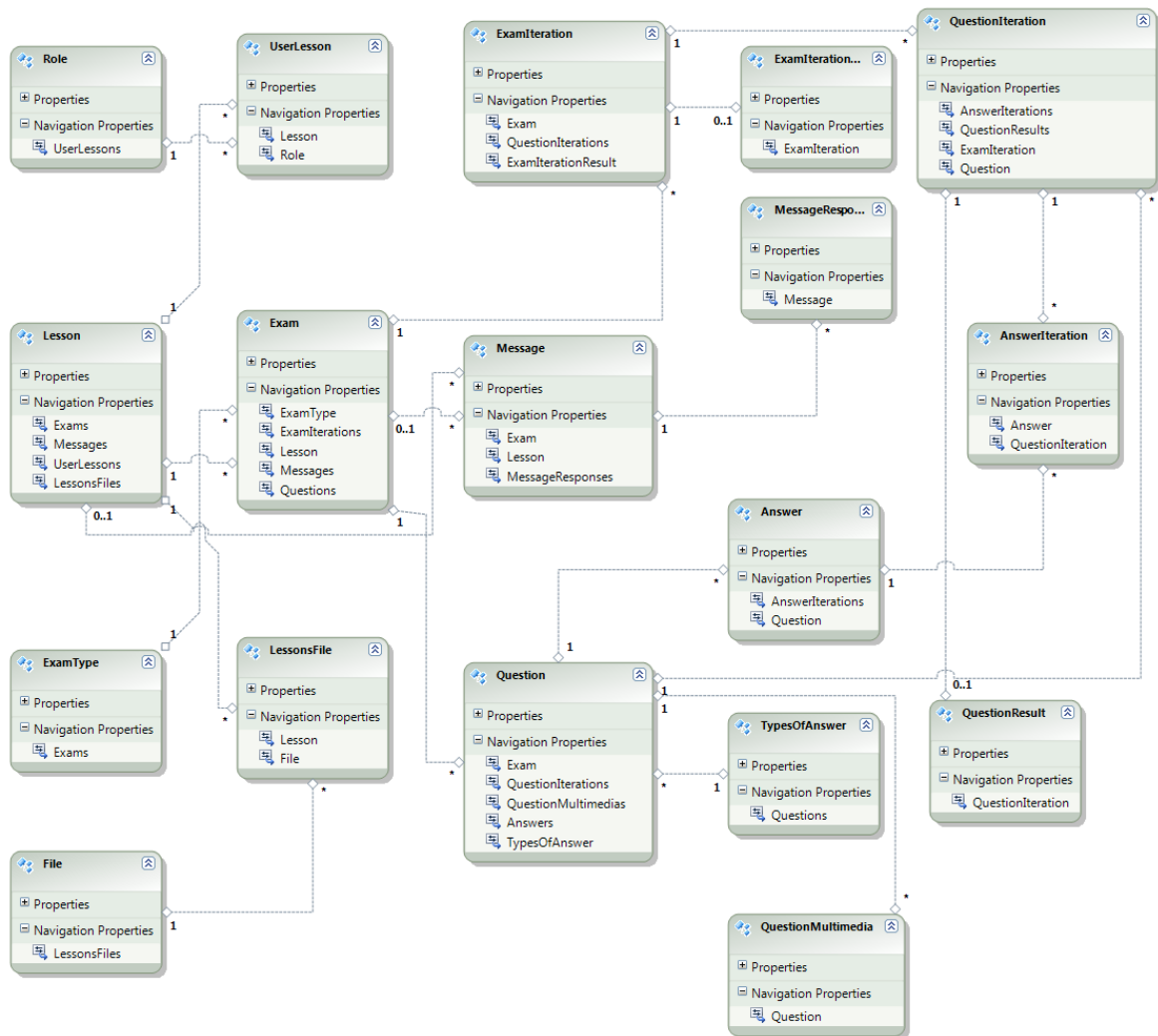
Důležitou třídou je zde `EntityHelper`, tato třída obsahuje metody, díky kterým je možné získat nebo nastavit klíč referenčního objektu. Příkladem může být test, který má v sobě uložený ID kurzu, ke kterému je přiřazen. Jelikož `Entity Framework` funguje na základě entit, tak nebýt pomocných metod v `EntityHelper`, by se musel získávat z databáze celý záznam o kurzu, i když nám stačí pouze jeho ID.

Dalším pomocníkem je zde třída `UserHelper`, která obsahuje konstanty s názvy rolí v databázi. Ty jsou používány při dotazování, kdy potřebujeme získat data pro určitou roli.

### 9.3.5 ElearningModel

V tomto souboru najdeme entitně-relační diagram, který je znázorněn na obrázku č. 25. Kvůli přehlednosti jsou v diagramu schovány u všech entit vlastnosti a jsou zobrazeny pouze navigační vlastnosti pro přehlednost diagramu. Všechny tabulky mají společné tři vlastnosti. První dvě jsou `ChangeDate` a `ChangeUser`, ty slouží k zjištění, kdo naposled s danou entitou manipuloval a kdy. Třetím je `RowVersion`, který se využije v případě použití transakcí na odrolování změn.





Obrázek 27 - Entitně-relační diagram

**Lesson** – Tato entita je základním kamenem celé aplikace. Přestavuje kurz, ke kterému máme přiřazené testy, uživatele, studijní materiály a dotazy od studentů. Tyto všechny objekty jsou zde zastoupeny kolekcí typu `EntityCollection`. Dále samozřejmě obsahuje vlastnosti jako je název, popis, apod.

**UserLesson** – Entita kde je uloženo přihlašovací jméno uživatele, roli kterou má přidělenou a kurz, ke kterému je přidělen.

**Role** – Entita obsahuje ID a název role. Jsou zde předem nadefinovány dvě role a to lektor a student.

**LessonFile** – Entita je obdobou M:N relace v databázi a propojuje kurz a dokument v podobě studijního materiálu. Toto propojení má výhodu, že můžeme jeden materiál přiřadit více kurzům.

**File** – Zde jsou uchovány informace o souboru. Soubor se nejdříve nahraje pomocí třídy `SharepointRepository` metodou `Add`, která má jako návratovou hodnotu ID souboru, pod kterým se vytvoří entita v databázi.

**Message** – Entita sloužící k vytvoření dotazu na lektora. Tento dotaz musí mít referenci na kurz, ke kterému se vztahuje. Může obsahovat i referenci na určitý test v daném kurzu, ale to není povinné. Dále obsahuje kolekci odpovědí typu `MessageResponse`.

**MessageResponse** – Entita představuje odpověď na dotaz, kde se uchovává například samotná odpověď, přihlašovací jméno lektora, který odpověděl a referenci na dotaz.

**Exam** – Entita představuje test, kde jsou uloženy například informace o počtu otázek, kolik je maximální a minimální počet bodů pro splnění testu, apod. Dále obsahuje reference na:

- typ testu,
- kolekci dotazů vztahených k tomuto testu,
- kurz, ve kterém se nachází,
- kolekci instancí testu pro jednotlivé studenty,
- otázky v tomto testu.

**ExamType** – Představuje typ testu, kde v databázi má dva zástupce a to `Exam` pro zkuškový test, který představuje závěrečný test. Druhý typ je `Preparation`, pro výukové testy sloužící k výuce.

**ExamIteration** – Entita zastupuje vytvořenou instanci testu daným studentem. Obsahuje informace jako start testu, očekávaný čas ukončení a skutečné dokončení testu. Dále obsahuje referenci na test, z kterého byl vytvořen odkaz na výsledek této instance a kolekci instancí otázek, na které student odpověděl.

**ExamIterationResult** – Představuje výsledek instance testu, kde je uloženo například celkové skóre, které student získal v testu. Čas, kdy byl výsledek vypočítán a příznak, zda student uspěl v testu či ne.

**Question** – Entita zastupující otázku v testu. Otázka obsahuje samotnou otázku s případným popisem, pořadím, v jakém bude zobrazeno, bodové ohodnocení a další. Dále obsahuje reference na:

- test, ke kterému je vztažen,
- typ odpovědi,
- multimediální otázku,
- instanci otázky,
- kolekci odpovědí k dané otázce.

**TypeOfAnswer** – Typ otázky, neboli typ odpovědi. Typy zde mám tři, z toho dva se vyhodnotí automaticky po skončení testu. První typ automaticky vyhodnocený je `CheckBox`, u kterého je povoleno více odpovědí. Druhý je `RadioButton`, kde je povolena pouze jedna odpověď. Posledním typem je `Text`, kdy student napíše vlastní odpověď a lektor po dokončení testu musí ohodnotit.

**Answer** – Odpověď na otázku, která obsahuje samotnou textovou odpověď a případně její popis. Dále obsahuje i příznak, jestli je odpověď správná či ne. Samozřejmě v sobě má referenci na otázku, ke které se vztahuje a instanci této odpovědi.

**AnswerIteration** – Instance odpovědi, kde je nejdůležitější vlastnost příznak `Selected`, který nám říká, že byla zaškrtnuta.

**QuestionMultimedia** – Entita sloužící například pro zobrazení multimediálního souboru, na jeho základě pak student odpoví.

**QuestionIteration** – Instance dané otázky, která v sobě obsahuje čas zobrazení nebo psanou odpověď, v případě, že typ otázky je `Text`. V opačném případě obsahuje referenci na odpověď, kterou student zvolil. Dále má reference na otázku, která je instance. Stejně tak i instance testu, ke kterému se vztahuje a nakonec na výsledek této instance otázky.

**QuestionResult** – Entita představuje výsledek dané instance otázky a obsahuje příznak, zda potřebuje opravit od lektora, komentář nebo konečné skóre za odpověď.

## 10 Závěr

Cílem této práce bylo vytvoření aplikace, která by podpořila elektronické vzdělávání a seznámit čtenáře postupem návrhu a vývoje.

Před samotným vývojem bylo potřeba zjistit, co vše v sobě obsahuje elektronické vzdělávání a v jakých formách je možné se s ním setkat. Tyto formy byly zastoupeny mnoha aplikacemi. Proto jsem vytvořil samostatnou kapitolu pro porovnání nejčastěji nasazovaných aplikací pro elektronické vzdělávání. Na základě tohoto srovnání jsem se rozhodl vytvořit aplikaci, kterou je možné použít jako samostatnou webovou stránku, ale i jako modul pro Microsoft Office Sharepoint (MOS). Tím byla dána i technologie pro vývoj a to ASP.NET spolu s Hermes Frameworkem pro integraci do MOS.

Samotný vývoj aplikace pak probíhal ve vývojovém prostředí Visual Studio 2010 v šabloně Hermes Framework za použití technologie ASP.NET s jazykem C#. K zajištění uživatelsky přívětivého a efektivního prostředí se použila knihovna ovládacích prvků od firmy Telerik. Uložiště dat zde bylo rozděleno na 2 části. První část je databáze, do které se přistupuje pomocí dotazovacího jazyka LINQ v rámci Entity Frameworku. V této databázi se pak nacházejí všechny data aplikace, kromě studijních materiálů. Druhá část je tedy určena pro ukládání studijních materiálů. Pro tyto studijní materiály byla vytvořena samostatná aplikace s interfacem, který deklaruje operace pro práci se studijními materiály. Na jeho základě se pak implementovalo ukládání do knihovny dokumentů v MOS. Jaká finální implementace bude použita v aplikaci, je vyřešeno návrhovým vzorem Dependency Injection, který značně snižuje závislost tříd.

Cíle práce byly splněny v celém rozsahu. Dále by se tato aplikace mohla rozšířit několika směry. Příkladem může být práce se studijními materiály, kde by se nemuselo pracovat pouze se soubory v rámci MOS. Mohly by zde být odkazy na samostatné stránky či knihovny dokumentů. Dále by bylo možné propojit aplikaci s fulltextovým vyhledáváním, který MOS podporuje. Uživatel by tak mohl ihned nalézt dokument, který obsahuje hledaný výraz. V případě, že na MOS je nainstalován modul Office Web Apps, mohl by si i tento dokument prohlédnout ve webovém prohlížeči.

Na konci bych chtěl jen zmínit, že tato práce pro mne, jako autora, byla velkým přínosem. Na této práci jsem se naučil technologie Entity Framework a ASP.NET, které mi budou velkou výhodou při hledání zaměstnání.

## 11 Literatura

- [1] E-learning. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-23]. Dostupné z: <http://cs.wikipedia.org/wiki/E-learning>
- [2] KVĚTOŇ, Karel. *Úloha e-learningu na školách: základní informace pro manažery vzdělávání*. Ostravská univerzita v Ostravě, 2005. [cit. 2012-04-07]. Dostupné z: [http://virtualni.osu.cz/e-learning\\_pro\\_skoly/Kveton-Uloha\\_e-learningu\\_na\\_skolach.pdf](http://virtualni.osu.cz/e-learning_pro_skoly/Kveton-Uloha_e-learningu_na_skolach.pdf)
- [3] PEJŠA, Jan. KONTIS. *LCMS a LMS, vývoj kurzů*. [cit. 2012-04-09] Dostupné z: [http://www.e-learn.cz/soubory/LMS\\_LCMS.pdf](http://www.e-learn.cz/soubory/LMS_LCMS.pdf)
- [4] MAZAL, Ferdinand. NET UNIVERZITY S.R.O. *VÝHODY A NEVÝHODY E- LEARNINGU*. 2006. [cit. 2012-04-09]. Dostupné z: [www.net-university.cz/download/seminar/2.ppt](http://www.net-university.cz/download/seminar/2.ppt)
- [5] ASP.NET. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-09]. Dostupné z: <http://cs.wikipedia.org/wiki/ASP.NET>
- [6] DILLINGEROVÁ, Monika. *SYSTÉMY PRE e-LEARNINGOVÉ VZDELÁVANIE*. 2007. Dostupné z: <http://www.ddm.fmph.uniba.sk/files/EMATIK/Dillingerova.pdf>. Univerzita Komenského.
- [7] Úvod do LINQ. In: *Www.vyvojar.cz* [online]. 24.1.2008 [cit. 2012-04-09]. Dostupné z: <http://www.vyvojar.cz/Articles/563-uvod-do-linq.aspx>
- [8] Microsoft SharePoint 2010. *Www.microsoft.com* [online]. 2010 [cit. 2012-04-09]. Dostupné z: <http://www.microsoft.com/cze/sharepoint/2010/>
- [9] ADO .NET Entity Framework. 2008. Dostupné z: [blog.vyvojar.cz/files/folders/228675/download.aspx](http://blog.vyvojar.cz/files/folders/228675/download.aspx)
- [10] Stříteská H. *Historie e-learningu v České republice* [online] 2007. [cit.10.4.2012] Dostupné z: <http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xstrites.htm>
- [11] ŠIKO, Petr. Moderní formy elektronického vzdělávání. In: *Ihned.cz* [online]. 12. 9. 2005 [cit. 2012-04-10]. Dostupné z: [http://ihned.cz/2-22676725-000000\\_d-92](http://ihned.cz/2-22676725-000000_d-92)
- [12] ŠKRABAL, Ondřej. XHTML. In: *Www.webtiger.cz* [online]. 27.3.2011 [cit. 2012-04-21]. Dostupné z: <http://www.webtiger.cz/Clanek/10/XHTML>
- [13] JavaScript. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-20]. Dostupné z: <http://cs.wikipedia.org/wiki/JavaScript>
- [14] Měníme stránky ASP na stránky ASP.NET. In: *Http://www.zive.cz* [online]. 11. 8. 2005 [cit. 2012-04-09]. Dostupné z: <http://www.zive.cz/clanky/menime-stranky-asp-na-stranky-aspnet/sc-3-a-126126/default.aspx>
- [15] T. Pitner. *E-learning na Masarykově univerzitě (2)*. Zpravodaj ÚVT MU. ISSN 1212-0901, 2003, roč. XIII, č. 3, s. 14-16.

[16] Jak volit vhodný SW nástroj pro tvorbu obsahu odpovídající výukovému modelu.  
In: *Www.inforama.cz* [online]. 2005 [cit. 2012-04-26]. Dostupné z:  
[http://www.inforama.cz/aktuality/zaujalo\\_mne/2009/009\\_sw-nastroj-pro-vyuku.htm](http://www.inforama.cz/aktuality/zaujalo_mne/2009/009_sw-nastroj-pro-vyuku.htm)

## **12 Seznam příloh**

### **Příloha č.1: CD obsahující**

- Tento dokument
- Zdrojové kódy aplikace

### **Příloha č.2: Srovnání e-learningových aplikací**