

Informační systém pro podporu řízení, správu a zjišťování aktuálního stavu rozvrhované výuky

Information System For Computer Aided Course Planning and Scheduling

Petr Čápek



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2010/2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Petr ČÁPEK
Osobní číslo: A08613
Studijní program: B 3902 Inženýrská informatika
Studijní obor: Informační a řídicí technologie

Téma práce: Informační systém pro podporu řízení, správu a zjišťování aktuálního stavu rozvrhované výuky

Zásady pro vypracování:

1. Zpracujte literární rešerši na téma tvorby databázových aplikací.
2. Analyzujte možnosti implementace.
3. Navrhněte vlastní implementaci včetně grafického uživatelského rozhraní.
4. Demonstrujte ukázkovou aplikaci.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

1. BILL EVJEN, Christian Nagel, et al. C 2008 Programujeme profesionálně. Praha : Computer Press, 2009. 1904 s. ISBN 978-80-251-2401-7.
2. LAIR, Robert. Beginning Silverlight 4 in C. Is.I.J : Apress, 2010. 395 s. ISBN 978-1-4302-2988-9.
3. FAROULT, Stéphane; ROBSON, Peter. The Art of SQL. Sebastopol : O'Reilly Media, Inc, 2006. 349 s. ISBN 978-0-596-00894-9.
4. HOLZNER, Steve. Design Patterns For Dummies. Indianapolis : For Dummies, 2006. 308 s. ISBN 978-0-471-79854-5.
5. Microsoft Developer Network [online]. 2011 [cit. 2011-01-31]. Windows Communication Foundation. Dostupné z WWW: [http://msdn.microsoft.com/en-us/library/ms735119(v=vs.90).aspx].

Vedoucí bakalářské práce:

Ing. Erik Král

Ústav bezpečnostního inženýrství

Datum zadání bakalářské práce:

25. února 2011

Termín odevzdání bakalářské práce:

7. června 2011

Ve Zlíně dne 25. února 2011

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Práce se zabývá implementací informačního systému pro podporu řízení, správu a zjišťování aktuálního stavu rozvrhované výuky. Uživatelské rozhraní je realizované v aplikační platformě Microsoft Silverlight s využitím technologie Windows Communication Foundation Rich Internet Application (WCF RIA), objektově relačním mapování Entity Framework a integrovaném jazyku pro dotazování LINQ. Datový model je implementovaný v relačním databázovém systému Microsoft SQL Server. V práci je zpracován přehled použitých technologií a možnosti implementace. Výstupem práce je funkční aplikace s grafickým rozhraním. Aplikace byla použita při tvorbě úvazků na Ústavu bezpečnostního inženýrství Univerzity Tomáše Bati.

Klíčová slova: WCF RIA, Rich Internet Application, Silverlight, LINQ, Entity Framework, rozvrhování výuky, webová aplikace

ABSTRACT

The aim of this work is design and realization of information system for computer aided course planning and scheduling. User interface is implemented in Microsoft Silverlight application platform using Windows Communication Foundation Rich Internet Application (WCF RIA), Entity Framework object-relation mapping and Language-Integrated Query (LINQ). Data model is implemented in Microsoft SQL relational model database server. This work includes description of selected technologies and analysis of implementation. The result of this work is functional application with user interface which was used for course planning and scheduling at Department of Security Engineering of Thomas Bata University.

Keywords: WCF RIA, Rich Internet Application, Silverlight, LINQ, Entity Framework, Course Scheduling, Web Application

Zde bych chtěl poděkovat vedoucímu práce Ing. et Ing. Eriku Královi za odborné vedení, rady a konzultace bakalářské práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 DATOVÝ MODEL.....	11
1.1 LINQ	11
1.2 ENTITY FRAMEWORK.....	15
2 UŽIVATELSKÉ ROZHŘANÍ A APLIKAČNÍ LOGIKA.....	18
2.1 SILVERLIGHT	18
2.2 WCF RIA.....	21
II PRAKTICKÁ ČÁST	23
3 ANALÝZA PROBLÉMU	24
3.1 SOUČASNÝ STAV ŘEŠENÍ	24
3.2 OBJEKTY A JEJICH VZTAHY	25
3.2.1 Vyučující	25
3.2.2 Předměty	27
3.2.3 Studijní obory	28
3.2.4 Základní vztahy	29
3.2.5 Požadované funkce.....	30
4 IMPLEMENTACE DATOVÉHO MODELU	31
4.1 DATOVÝ MODEL	31
4.1.1 Tabulka „kecatko“	32
4.1.2 Tabulka „koeficienty“	33
4.1.3 Tabulka „kvalifikace“	33
4.1.4 Tabulka „období“	34
4.1.5 Tabulka „obor“	34
4.1.6 Tabulka „permis“	35
4.1.7 Tabulka „semestr“	35
4.1.8 Tabulka „predmet“	36
4.1.9 Tabulka „semestr“	37
4.1.10 Tabulka „semestr_predmet“	37
4.1.11 Tabulka „sloucena_vyuka“	37
4.1.12 Tabulka „ustav“	38
4.1.13 Tabulka „vyucujici_obdobi“	38
4.1.14 Tabulka „vyucujici“	39
4.1.15 Tabulka „vyuka“	40
5 IMPLEMENTACE UŽIVATELSKÉHO ROZHŘANÍ A APLIKAČNÍ LOGIKY.....	41
5.1 STRUKTURA APLIKACE	41
5.1.1 Klient.....	42
5.1.2 Server	43
5.2 BEZPEČNOST APLIKACE	44
5.2.1 Autorizace a autentizace	44
5.2.2 Zabezpečení připojení	47

5.3	UŽIVATELSKÉ PRVKY	48
5.3.1	Přihlašovací formulář	48
5.3.2	Zobrazení domovské stránky	49
5.3.3	Editace oborů	49
5.3.4	Editace práv	50
5.3.5	Editace předmětů	51
5.3.6	Editace vyučujících	52
5.3.7	Editace výuky	53
5.3.8	Zobrazení vyučujících	56
5.3.9	Zobrazení výuky	57
5.4	DESKTOPOVÁ APLIKACE	58
6	ZHODNOCENÍ PŘÍNOSU APLIKACE	60
	ZÁVĚR	61
	ZÁVĚR V ANGLIČTINĚ	62
	SEZNAM POUŽITÉ LITERATURY	63
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	65
	SEZNAM OBRÁZKŮ	66
	SEZNAM TABULEK	67
	SEZNAM PŘÍLOH	68

ÚVOD

Cílem práce je vytvoření informačního systému pro podporu řízení, správu a zjišťování aktuálního stavu rozvrhované výuky. Tento systém momentálně na Univerzitě Tomáše Bati ve Zlíně (UTB) neexistuje a jeho implementace by mohla zjednodušit práci zaměstnancům UTB. Systém umožňuje přiřazování výukových jednotek vyučujícím na základě informací o počtu studijních skupin v jednotlivých studijních oborech a programech. Součástí systému je vytváření přehledu o výuce jednotlivých vyučujících a předmětech.

Uživatelské rozhraní je realizované v aplikační platformě Microsoft Silverlight s využitím technologie Windows Communication Foundation Rich Internet Application (WCF RIA) [1], objektově relačním mapování Entity Framework [2] a integrovaném jazyku pro dotazování LINQ [3]. Datový model je implementovaný v relačním databázovém systému Microsoft SQL Server [4]. V práci je zpracován přehled použitých technologií a možnosti implementace. V práci je realizovaná vlastní implementace včetně grafického uživatelského rozhraní a jsou prezentovány dosažené výsledky. Aplikace byla použita při tvorbě úvazků na Ústavu bezpečnostního inženýrství Univerzity Tomáše Bati.

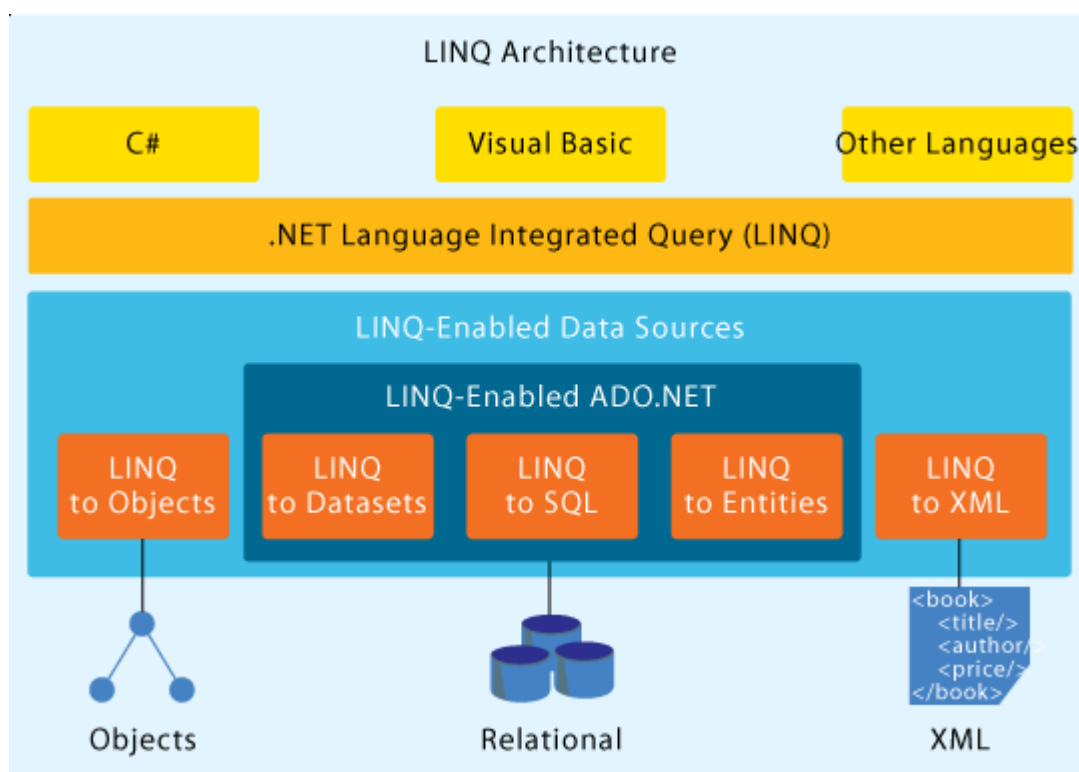
I. TEORETICKÁ ČÁST

1 DATOVÝ MODEL

V následující kapitole budou rozebrány technologie, které byly zvoleny pro vytvoření řešeného informačního systému pro podporu řízení, správu a zjišťování aktuálního stavu rozvrhované výuky. Tyto technologie představují obecné řešení databázových aplikací. Zvolené technologie umožňují rychlý vývoj aplikací a jejich snadnou údržbu. Nejprve jsou popsány technologie pro implementaci datového modelu a poté technologie pro řešení aplikační logiky a uživatelského rozhraní.

1.1 LINQ

Language-Integrated Query (LINQ) představuje nejvýznamnější změnu v C# 3.0 a .NET 3.5. S pomocí LINQ je možné integrovat syntaxi dotazů přímo do jazyka C#. LINQ umožňuje přístup k různým datovým zdrojům bez změny syntaxe. Tento přístup k různým datovým zdrojům je umožněn aplikací abstraktní vrstvy.[5][14]



Obr. 1. Architektura LINQ [21]

Standardně firma Microsoft dodává několik implementací LINQ. Mezi nejdůležitější lze zařadit [12][14]:

- **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

V další části textu bude popsáno využití integrovaného jazyka na ukázkovém příkladu. Nechť je definována následující třída *MojeTrida* následujícím způsobem:

```
public class Mojetrida
{
    public string jmeno { get; set; }
    public int cislo { get; set; }
}
```

Třída *mojetrida* obsahuje dvě vlastnosti (properites). Vlastnost *jmeno*, což je řetězec a vlastnost *cislo*, což je celočíselný datový typ. Vytvoříme instanci typu `List<mojetrida>` a naplníme ji daty.

```
List<mojetrida> list = new List<mojetrida>();
list.Add(new mojetrida() { jmeno = "tom", cislo = 1 });
list.Add(new mojetrida() { jmeno = "jon", cislo = 2 });
list.Add(new mojetrida() { jmeno = "jan", cislo = 1 });
list.Add(new mojetrida() { jmeno = "tim", cislo = 2 });
```

Máme za úkol vyhledat v našem listu ty záznamy, které mají vlastnost *cislo* rovnu 2, výsledky seřadit abecedně podle vlastnosti *jmeno* a vypsát tato data.

Řešení bez použití technologie LINQ

V době, kdy neexistovalo LINQ, by byl problém implementován nejspíše takto:

```
List<mojetrida> vysledek = new List<mojetrida>();
foreach (mojetrida p in list)
if (p.cislo == 2)
    vysledek.Add(p);

vysledek.Sort(jaktridit);

foreach (mojetrida p in vysledek)
    Console.WriteLine(p.jmeno + " " + p.cislo);
```

K setřídění by byla použita funkce *jaksetridit*:

```
public int jaktridit(mojetrída a, mojetrida b)
{
    return a.jmeno.CompareTo(b.jmeno);
}
```

Zkušenější programátor by využil možnosti delegátů a implementace by nejspíše vypadala takto:

```
List<mojetrida> vysledek = list.FindAll(
    delegate(mojetrída p)
    {
        return p.cislo == 2;
    });
// v tomto miste jsou již data v proměně vysledek

vysledek.Sort(
    delegate(mojetrída p1, mojetrida p2)
    {
        return p1.jmeno.CompareTo(p2.jmeno);
    });

foreach (mojetrida p in vysledek)
    Console.WriteLine(p.jmeno + " " + p.cislo);
```

Hlavní nevýhoda toho řešení spočívá v tom, že data jsou v proměnné *vysledek* ještě před tím, než je skutečně potřebujeme. Zatěžujeme tedy zbytečně paměť a také pokud dojde ke změně dat v proměnné *list* potom, co jsme provedli metodu *FindAll*, neprojeví se tato změna ve *vysledek*.

Řešení s použitím extensit metod pro LINQ bez lambda výrazů

LINQ s sebou přináší implementaci extension pro vybrané třídy a interface. Jeden z interface, který implementuje extension metody, je *IEnumerable*. Jedna z možností jak implementovat zadaný příklad je např.:

```
IEnumerable<mojetrida> vysledek = list.Where(
    delegate(mojetrída p)
    {
        return p.cislo == 2;
    }).OrderBy(
    delegate(mojetrída p)
    {
        return p.jmeno;
    }).Select(
    delegate(mojetrída p)
    {
        return p;
    });

foreach (mojetrida p in vysledek) Console.WriteLine(p.jmeno + " " + p.cislo);
```

Oproti předešlým implementacím poskytuje tato implementace tu výhodu, že data proměnná *vysledek* se daty naplní až při výpisu, což tolik nezatěžuje paměť a data jsou vždy aktuální.

S použitím extensit metod pro LINQ s lambda výrazy

S příchodem platformy .NET 3 se objevují tzv. lambda výrazy, což není nic jiného, než anonymní funkce. Tyto výrazy slouží hlavně ke zkrácení kódu a zlepšení přehlednosti. Předešlé řešení je možné přepsat pomocí lambda výrazů následovně:

```
IEnumerable<mojetrida> vysledek = list.Where(p => p.cislo == 2).OrderBy(p =>
p.jmeno).Select(p => p);

foreach (mojetrida p in vysledek)
    Console.WriteLine(p.jmeno + " " + p.cislo);
```

Můžeme vidět, že tento kód je výrazně kratší a přehlednější oproti minulé ukázce.

S použitím klíčových slov LINQ a anonymních tříd

Poslední příklad je ukázkou předešlého dotazu implementovaného pomocí klíčových slov LINQ. Dále proměnná *vysledek* již není typu *IEnumerable<mojetrida>* ale je deklarovaná jako tzv. anonymní třída. Výhoda anonymních tříd je, že nemusí obsahovat stejné vlastnosti (properties) jako třída, z níž je dotaz generován (což u *IEnumerable<>* není možné).

```
var vysledek = from o in list
               where o.cislo == 2
               orderby o.jmeno
               select o;

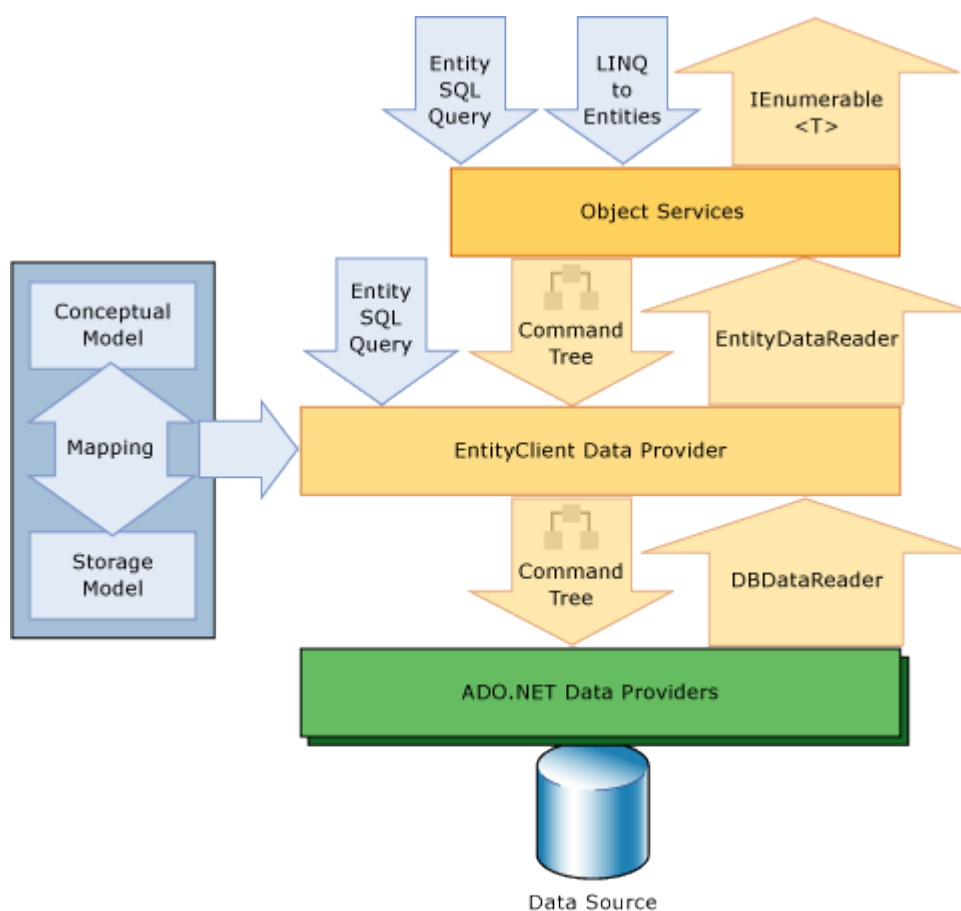
foreach (var p in vysledek)
    Console.WriteLine(p.jmeno + " " + p.cislo);
```

Toto řešení na první pohled připomíná „obrácený“ SQL dotaz [7]. Pomocí klíčových slov LINQ je možné implementovat libovolný výběrový dotaz SQL. Oproti úplně první implementaci je zde vidět výrazné zkrácení kódu, výrazné zlepšení přehlednosti a výrazný nárůst rychlosti při zachování minimálních paměťových požadavků.

1.2 Entity framework

ADO.NET Entity Framework je objektově relační mapovací prostředí, které vzniklo s příchodem platformy .NET 3.5. Principem objektově relačního mapování je, že prvky databáze jsou převedeny na objekty. V Entity Frameworku (EF) jsou tabulky převedeny na Entity set, řádky tabulky na Entity, sloupce tabulky jsou převedeny na Properties. Struktura EF se dělí na následující vrstvy [5][15]:

- Logická – logická vrstva je definována pomocí SSDL (Store schema definition language) a popisuje fyzickou databázi (úložiště dat)
- Konceptuální – vrstva je definována pomocí CSDL (Conceptual sheme definition language) a popisuje třídy .NET (programátor pracuje pouze s touto vrstvou)
- Mapovací – vrstva je definována jazykem MSL (Mapping specification language) a mapuje objekty konceptuální vrstvy na logickou



Obr. 2. Entity framework architektura [2]

S daty je možné pracovat před EF následujícími způsoby[15]:

- Pomocí standardních tříd ADO .NET (connection, comand, datareader) – tento způsob komunikace umožňuje pouze čtení dat, nikoli však modifikaci. Příkazy je třeba definovat jako řetězce (bez podpory IntelliSense) a výsledky jsou vráceny v netypové formě.
- Pomocí eSQL vůči kontextu – umožňuje plně ovládat databázi. Příkazy je opět třeba definovat jako řetězce, avšak výsledky jsou již vráceny v typové formě.
- Pomocí LINQ vůči kontextu – umožňuje plné ovládání databáze. Také je zde podpora při psaní dotazů (IntelliSense) a kontrola syntaxe dotazů. Výsledky jsou vráceny v typové formě. Jedná se o nejčastější případ komunikace.

Mezi klíčové vlastnosti EF patří:

- Plná podpora LINQ
- Modifikace dat
- Odložené vyhodnocení dotazu
- Každá entita se načítá v rámci kontextu pouze jedenkrát (neexistují zde dva objekty pro jednu entitu)
- Podpora detekce konfliktů (Managing Concurrency)
- Podpora transakcí (Transaction)
- Řízení načítání dat z vnořených tabulek

Mezi hlavní konkurenty EF patří LINQ to SQL. EF je založen na modelu vycházejícího z poskytovatele, tudíž se do něj mohou zapojit i jiní výrobci databází. LINQ to SQL pracuje pouze s Microsoft SQL. EF nabízí také mnohem pokročilejší mapovací funkce a je nezávislý na skutečné databázi (logická vrstva se může lišit od konceptuální). Oproti jiným nástrojům pro objektově relační mapování má tu výhodu, že je standartní součástí .NET Frameworku 3.5 a vyšší a umožňuje použití LINQ (pomocí LINQ to Entity).

LINQ to SQL podporuje i jiné výrobce databází jako např. [15][16]:

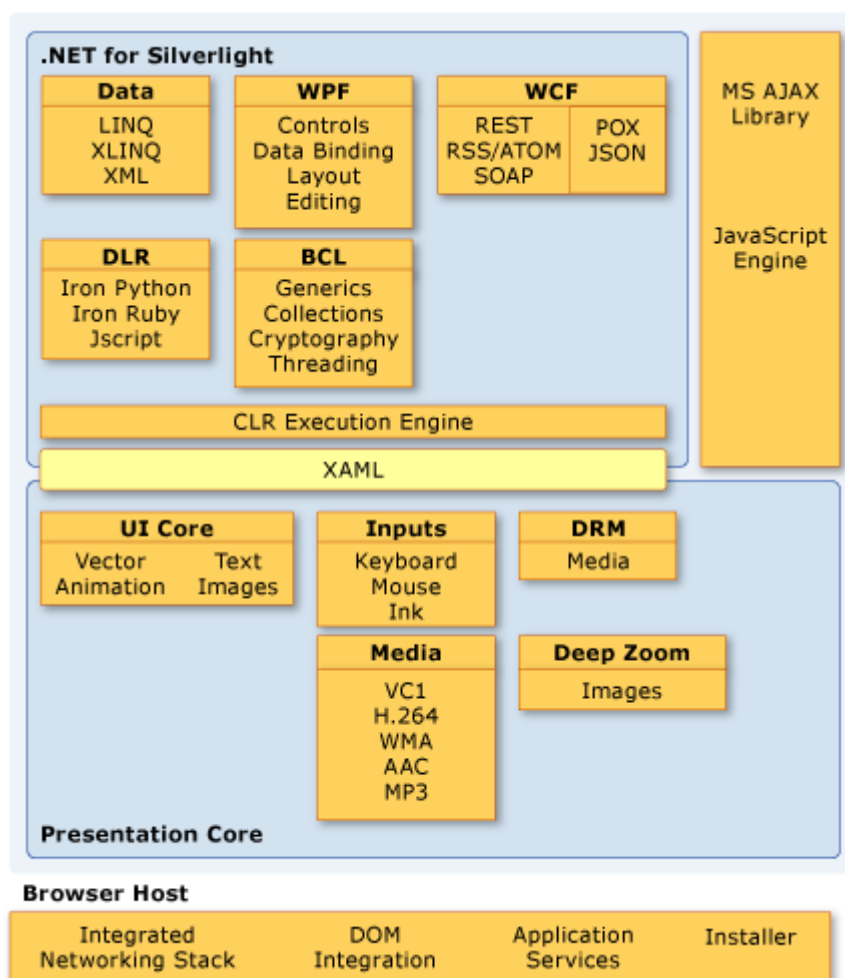
- DB2
- EffiProz
- Firebird
- Informix
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL
- SQLite
- Sybase
- VistaDB

2 UŽIVATELSKÉ ROZHRANÍ A APLIKAČNÍ LOGIKA

2.1 Silverlight

Silverlight je aplikační platforma pro tvorbu webových aplikací, které svým chováním a vzhledem připomínají desktopové aplikace. Tyto aplikace bývají označovány jako Rich Internet Application (RIA). Silverlight běží v rámci webového prohlížeče a používá plug-in, aby se aplikace správně zobrazila [6]. Je možné najít mnoho společného mezi Silverlightem a Adobe Flash. Oba běží jako plug-in poskytující platformu pro vlastní aplikaci, která běží na klientovi. Flash můžeme najít zhruba na 99% počítačů, zatímco Silverlight zhruba na 60% počítačů (k 4 dubnu 2010) [18][19] .

Silverlight je vhodný nejen pro tvorbu multimediálních aplikací, ale také pro podnikové aplikace. Jeho výhodou jsou možnosti integrace dat, mnoho existujících komponentů a doplňků, jako je například RIA Services Framework.



Obr. 3. Architektura platformy Silverlight [20]

Silverlight plug-in pracuje se všemi hlavními prohlížeči na dvou hlavních operačních systémech (Windows a Macintosh – pouze x86). Také existuje opensource projekt Moonlight [22], který má za cíl implementaci Silverlightu pro Linux. Také je možnost psát Silverlight aplikace pro Windows Phone 7 (pouze v Silverlight 3). Podpora ostatních mobilních operačních systémů se plánuje v budoucích verzích.

Rozhraní Silverlightu má velikost 6MB, což je opravdu málo vzhledem k tomu, že obsahuje část .NET Frameworku a plnou implementaci CLR (common language runtime). To je obrovská výhoda při psaní kódu, protože je možné použít jakýkoliv jazyk platformy .NET (standardně C#, VB) [13].

Obecné výhody Silverlightu:

- Jednoduché nasazení – aplikace je doručena přes internet skrze webový prohlížeč
- Není třeba plný .NET Framework – postačuje pouze malé rozhraní
- Aplikaci je možné napsat v libovolném jazyce .NETu (C#, VB)
- Aplikace může běžet na více platformách (Windows, Mac)
- Rychlý vývoj a lepší možnosti než HTML aplikace

Výhody Silverlightu oproti ASP.NET:

- Odpadá problém s různým vzhledem aplikace mezi prohlížeči i operačními systémy. Velká výhoda Silverlightu je, že aplikace se renderuje stejně ve všech prohlížečích na všech operačních systémech. To ušetří spoustu času s testováním a řešením takto vzniklých problémů.
- Silverlight aplikace jsou bohatší než HTML a vytvářejí se rychleji
- I když je Silverlight bohatý klient, uživatelé nemusí posílat konstantní postbacky na server, a to činí aplikaci svižnější
- S CLR a částí .NET Frameworku běžícím na klientovi již není třeba psát nic ve JavaScriptu a kód je znovupoužitelný v jiné Silverlight aplikaci
- Aplikace může běžet v offline módu
- Aplikace může být oddělena a může běžet jako standardní aplikace

Výhody Silverlightu oproti WinFormům:

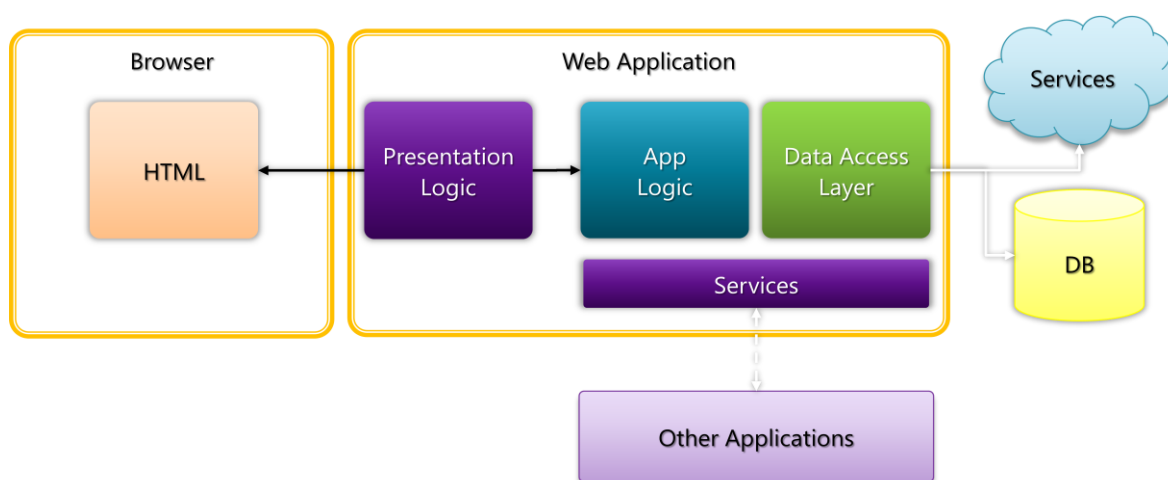
- Silverlight používá XAML na definování uživatelského rozhraní. To umožňuje vytvořit unikátní a flexibilní uživatelské rozhraní, které nabízí podporu pro data binding, vektorovou grafiku a animace
- RIA Services umožňují vytvořit aplikaci, která bude jednoduše a čistě komunikovat se serverem

Srovnání Silverlightu a Adobe Flash:

- Stejně vlastnosti
 - Oba patří do RIA platformy a můžou být nasazeni přes webový prohlížeč
 - Oba mohou fungovat mimo webový prohlížeč
 - Oba používají pro definice uživatelského rozhraní XML jazyk
- Výhody Flash
 - Flash je rozšířenější
 - Jádro Flashe je menší (1.8MB vs. 6 MB), ale pro spuštění mimo prohlížeč je třeba nainstalovat Adobe Air (15MB)
- Nevýhody Flash
 - Kód musí být psaný v ActionScriptu, který vnikl z JavaScriptu, a který není tak kvalitní jako managed jazyky (neexistuje zde LINQ ani generika). Kód je specifický pro Flash/Flax a nemůže být použit pro jiné aplikace
 - Potřebuje další aplikaci (AdobeAir), aby bylo možné spouštět aplikace mimo webový prohlížeč. I tak je třeba (pokud chceme spustit aplikaci ve Flashi bez webového prohlížeče) distribuovat aplikaci zvlášť (Silverlight umožňuje pouštět aplikace jak v rámci webového prohlížeče tak i mimo něj)

2.2 WCF RIA

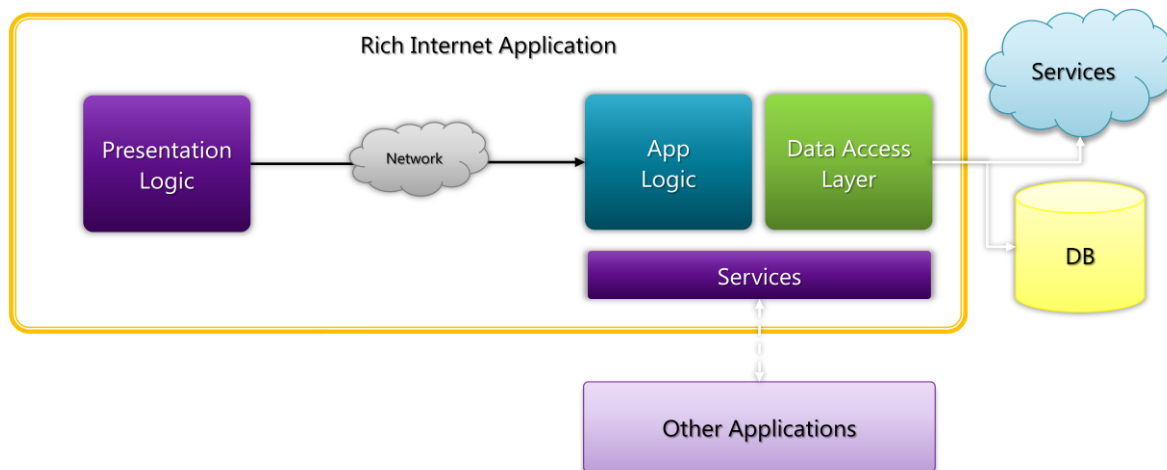
V aplikacích typu WPF, WinFrom je možné přistupovat přímo do databáze (např. pomocí SqlConnection, OdbcConnection ...). V Silverlightu toto není možné, protože Silverlight je klientská platforma, která může běžet v rámci prohlížeče kdekoli na světě a tudíž není možné se přímo napojit na databázi. Data musí být poskytnuta v rámci webové služby, která zajišťuje výměnu dat mezi databází a aplikací. Existuje mnoho způsobů, jak realizovat webovou službu. Jeden z nejznámějších a nejlepších je využití technologie WCF [9].



Obr. 4. Schéma standardní webové aplikace [17]

V Silverlightu je sice možné využít technologii WCF, ale ta s sebou přináší několik nevýhod. Použitím WCF narůstá množství opakujícího se kódu a objem komunikace, zvláště pokud je použito filtrování, třídění, seskupování a stránkování. Dále, pokud má aplikace používat validaci a přístupová práva, je třeba implementace na straně serveru i na straně klienta [10].

Ve snaze zjednodušit přístup k datům pomocí Silverlightu Microsoft představil novou technologii WCF RIA Services. RIA je nejvyšší vrstvou WCF, která poskytuje framework pro vytváření aplikací, které využívají datových služeb serveru. RIA také významným způsobem zjednodušuje proces vytváření webových služeb nad databází a jejich využívání v rámci Silverlightu.



Obr. 5. Schéma RIA aplikace [17]

Klíčové vlastnosti RIA:[12]

- Endpoints – WCF RIA podporuje konfiguraci endpointů. Standardně je použit binární endpoint, ale je možné přidat, či použít i jiné typy endpointů jako např. OData, Soap, Json
- DomainDataSource – DomainDataSource (DDS) umožňuje jednoduše provádět databinding nebo operace jako filtrování, třídění a stránkování.
- DomainService operace – DomainService poskytuje vysokoúrovňové programování, které může být použito při vytváření aplikační logiky. Také podporuje LINQ, které umožňuje vytvářet flexibilní a velmi efektivní způsob získávání dat. DomainService podporuje také CRUD operace (create, read, update, delete). Také nabízí možnost regulace zátěže serveru skrze nastavení maximálního počtu výsledků pro jednotlivé dotazy.
- Authentizace a autorizace – metody DomainService mohou být zabezpečeny pomocí autentizačních a autorizačních (na základě role) atributů. Také je zde integrován bezpečnostní model z ASP.NET, který významným způsobem usnadňuje zabezpečení služeb.
- Validate a metadata – validate může být použita na entity, jejich členy, DomainService metody a její parametry. Všechny validate definované na straně serveru mohou být automaticky převedeny na klienta. Silverlight komponenty jako datagrid či dataform automaticky podporují data anotaci bez potřeby složité implementace.

II. PRAKTICKÁ ČÁST

3 ANALÝZA PROBLÉMU

Používají se datově i formátově nekompatibilní formuláře, do kterých se stejná data přepisují ručně, formuláře se většinou tisknou nebo posílají emailem, neexistuje centrální evidence platných verzí. Každý tajemník používá vlastní decentralizovaný systém, většinou založený na programu Microsoft Excel, kdy evidence není vzájemně propojená

3.1 Současný stav řešení

Decentralizované vstupy potřebné pro rozvrhování

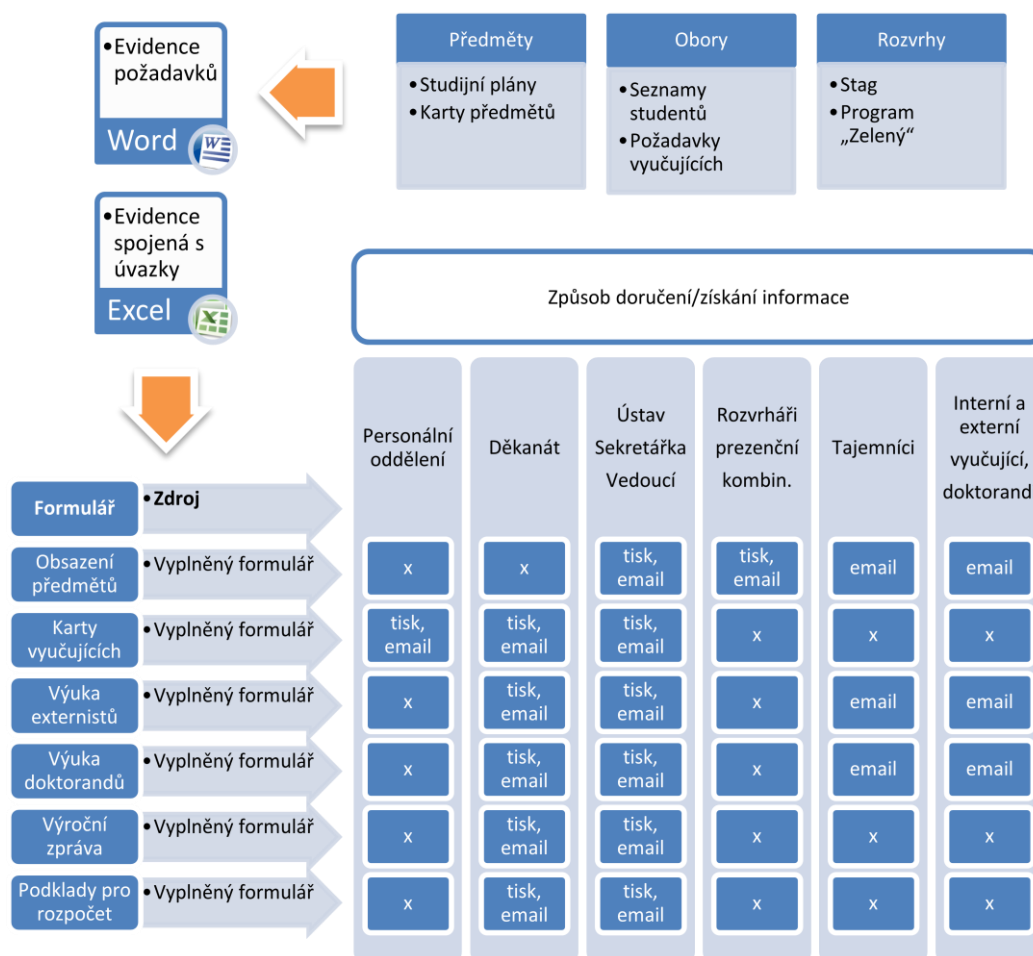
- Studijní plány (v pdf, nebo tištěné).
- Karty předmětů v systému STAG, data se přepisují ručně.
- Seznamy studentů, každá fakulta má vlastní evidenci, neexistuje jeden důvěryhodný zdroj.
- Rozvrh zadaný v desktopovém programu, data se posílají emailem.
- Rozvrh ručně přepsaný do systému STAG.

Některé z používaných formulářů

- Požadavky vyučujících k rozvrhu FAI
- Požadavky vyučujících k předmětům FAI
- Karta vyučujícího
- Přehled výuky externistů a doktorandů
- Přehled výuky pro výroční zprávu a přehled výuky pro tvorbu rozpočtu

Při změně vstupních dat musí tajemník ručně změnit formuláře a v písemné podobě nebo emailem je odevzdat:

- sekretářce ústavu
- rozvrhářům všech fakult
- interním zaměstnancům, externím zaměstnancům a doktorandům
- tajemníkům všech ústavů a fakult
- vedoucím ústavů všech zaměstnanců
- personálnímu oddělení



Obr. 6. Současný stav řešení

3.2 Objekty a jejich vztahy

3.2.1 Vyučující

Pro Vyučujícího $V_j, j = 1, \dots, n_V$ jsou uvedeny v Tab. 1 dynamické parametry, které se určují výpočtem.

Tab. 1. Dynamické parametry vyučujícího

Parametr	Popis
ZH_j	Aktuální úvazek vyučujícího za školní rok v započítatelných hodinách (ZH).
A_j	Počet započítatelných hodin přímé výuky za školní rok v ZH..
B_j	Počet ZH zkoušení a tutoringu za školní rok v započítatelných hodinách.

V Tab. 2 jsou uvedeny statické parametry vyučujícího.

Tab. 2. Statické parametry vyučujícího

Parametr	Popis
C_j	Počet započítatelných hodin ostatní pedagogické činnosti za školní rok v započítatelných hodinách.
$úvazek_j$	Naplánovaný úvazek vyučujícího za školní rok.
$jméno_j$	Jméno a příjmení vyučujícího.
$vztah_j$	Vztah k Univerzitě Tomáše Bati ve Zlíně. Možné hodnoty: interní zaměstnanec, interní doktorand, externí doktorand, externista.
$ústav_j$	Ústav zaměstnance.

V Tab. 3 jsou uvedeny kolekce vyučujícího.

Tab. 3. Kolekce vyučujícího

Parametr	Popis
$Kvalifikován_j$	Množina předmětů, které je vyučující kvalifikován vyučovat.
$Předměty_j$	Množina předmětů vyučovaných vyučujícím.
$Garantuje_j$	Množina předmětů garantovaných vyučujícím.

3.2.2 Předměty

Pro Předmět $P_i, i = 1, \dots, n_p$ jsou v Tab. 4 uvedeny statické parametry předmětu, a v Tab. 5 kolekce předmětu.

Tab. 4. Statické parametry předmětu

Parametr	Popis
nt_i	Počet výukových týdnů předmětu za semestr.
h_i^p	Počet hodin jedné přednášky za výukový týden t_{P_i} .
h_i^c	Počet hodin jednoho cvičení za výukový týden t_{P_i} .
h_i^s	Počet hodin jednoho semináře za výukový týden t_{P_i} .
ns_i	Počet studentů zapsaných na předmět t_{P_i} .
$kredit_i$	Počet kreditů na předmět.
$ústav_i$	Ústav garantující předmět.
$zkratka_i$	Zkratka předmětu.
$název_i$	Název předmětu.
$Kvalifikování_i$	Množina vyučujících, kteří mohou vyučovat tento předmět.
$garant_i$	Garant předmětu.
$ukončení_i$	Typ ukončení předmětu, možné hodnoty: zkouška, klasifikovaný zápočet, zápočet.
$Obory_i$	Množina oborů, ve kterých se předmět vyučuje.
$kapacita_i^c$	Počet kroužků na jedno cvičení.
$kapacita_i^s$	Počet kroužků na jeden seminář.

Tab. 5. Kolekce předmětu

Parametr	Popis
$Vyučující_i$	Množina vyučujících, kteří předmět vyučují.

3.2.3 Studijní obory

V Tab. 6 jsou uvedeny statické parametry Studijní obor $O_k, k = 1, \dots, n_k$.

Tab. 6. Statické parametry studijního oboru

Parametr	Popis
ns_k	Počet studentů
nk_k	Počet studijních kroužků (cca 12 studentů)
$ročník_k$	Ročník oboru
$semestr_k$	Semestr oboru
$zkratka_k$	Zkratka oboru
$název_k$	Název oboru.
$forma_k$	Forma studia, možné hodnoty: prezenční, kombinované.
typ_k	Typ studia, možné hodnoty: bakalářské, magisterské.
$Místo_k$	Místo výuky oboru.

V Tab. 7 jsou uvedeny kolekce oboru.

Tab. 7. Kolekce oboru

Parametr	Popis
$Předměty_k$	Množina předmětů oboru.

3.2.4 Základní vztahy

Vztah (1) popisuje výpočet započitatelných hodiny vyučujícího.

$$ZH_j = A_j + B_j + C_j \cong \text{úvazek}_j \quad (1)$$

Vztah (2) popisuje výpočet započitatelných hodin za přímou výuku.

$$\begin{aligned} A_j = ZH_j^p \sum_{i=1}^{n_p} h_i^p n t_i \sum_{k=1}^{n_k} q_{i,j,k}^p \\ + ZH_j^c \sum_{i=1}^{n_p} h_i^c n t_i \sum_{k=1}^{n_k} q_{i,j,k}^c + ZH_j^s \sum_{i=1}^{n_p} h_i^s n t_i \sum_{k=1}^{n_k} q_{i,j,k}^s \end{aligned} \quad (2)$$

Vztahy (3), (4) a (5) popisují výpočet započitatelných hodin za zkoušení a tutoring.

$$\begin{aligned} w_{i,j,k}^{zk} \\ \cong \begin{cases} ns_k \frac{q_{i,j,k}^p}{\sum_{j=1}^{n_v} q_{i,j,k}^p}, & ukončení_i = zk \\ ns_k \frac{q_{i,j,k}^c}{\sum_{j=1}^{n_v} q_{i,j,k}^c}, & ukončení_i = klz \bigwedge klz \text{ se uděluje na cvičení} \\ ns_k \frac{q_{i,j,k}^s}{\sum_{j=1}^{n_v} q_{i,j,k}^s}, & ukončení_i = klz \bigwedge klz \text{ se uděluje na semináři} \end{cases} \end{aligned} \quad (3)$$

$$B_j = ZH^{zk} \sum_{i=1}^{n_p} \sum_{k=1}^{n_k} w_{i,j,k}^{zk} + ZH^{klz} \sum_{i=1}^{n_p} \sum_{k=1}^{n_k} w_{i,j,k}^{klz} \quad (4)$$

$$\begin{aligned} \sum_j q_{i,j}^c &= kapacita_{p_i}^c \sum_k nk_k, O_k \in Obory_i \\ \sum_j q_{i,j}^s &= kapacita_{p_i}^s \sum_k nk_k, O_k \in Obory_i \end{aligned} \quad (5)$$

Kde:

$q_{i,j,k}^p$ Je počet přednáškových skupin všech oborů předmětu P_j vyučovaných vyučujícím V_j .

$q_{i,j,k}^c$ Je počet skupin cvičení všech oborů předmětu P_j vyučovaných vyučujícím V_j . *skupina cvičení = kapacita $_{P_i}^c \times$ kroužek.*

$q_{i,j}^s$ Je počet skupin seminářů všech oborů předmětu P_j vyučovaných vyučujícím V_j . *skupina seminářů = kapacita $_{P_i}^s \times$ kroužek.*

ZH^p Započítatelné hodiny za hodinu přednášky.

ZH^c Započítatelné hodiny za hodinu cvičení.

ZH^s Započítatelné hodiny za hodinu semináře.

$w_{i,j,k}^{zk}$ Je počet studentů oboru O_k , předmětu P_j , kterým udělí zkoušku vyučující V_j .

$w_{i,j,k}^{klz}$ Je počet studentů oboru O_k , předmětu P_j , kterým udělí klasifikovaný zápočet vyučující V_j

ZH^{zk} Započítatelné hodiny za udělení zkoušky.

ZH^{klz} Započítatelné hodiny za udělení klasifikovaného zápočtu.

3.2.5 Požadované funkce

Požadované funkce systému:

- Udržování verzí v jednotlivých obdobích (školní rok)
- Editace $q_{i,j,k}^p$, $q_{i,j,k}^c$ a $q_{i,j,k}^s$ včetně zobrazení neobsazených skupin.
- Editace $w_{i,j,k}^{zk}$ a $w_{i,j,k}^{klz}$ včetně zobrazení neobsazených studentů.
- Editace statických (zadaných) parametrů P_i , V_j a O_k .

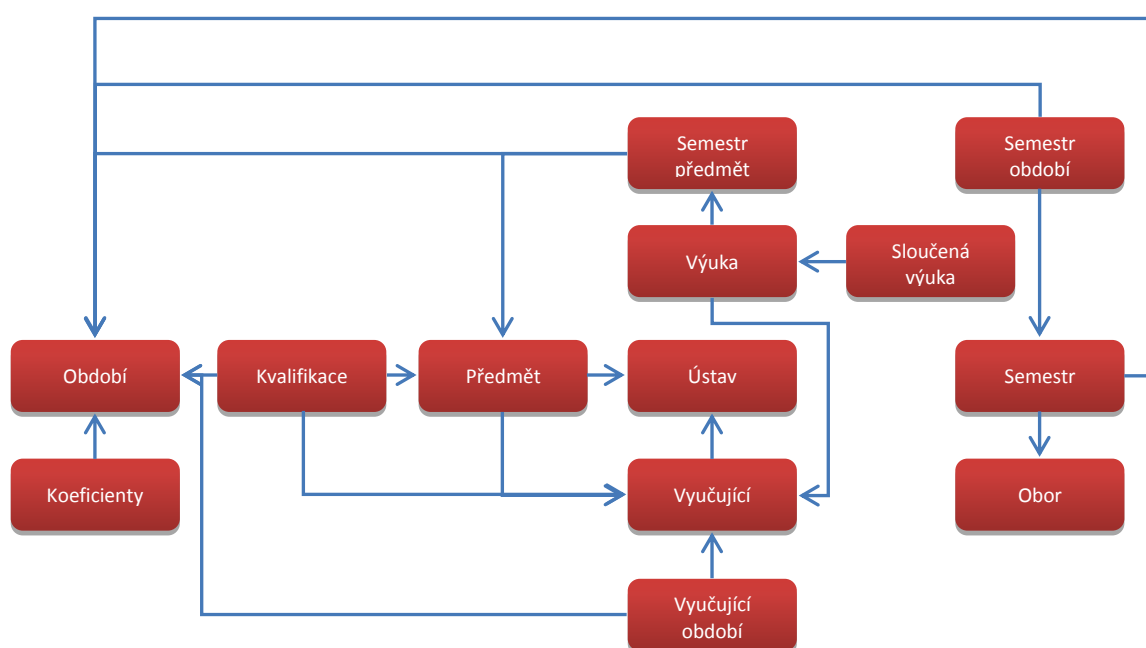
V aplikaci budou uživatelé „Host“ - varianta s heslem i bez hesla, práva pouze pro čtení, „Tajemník“ - práva pro editační funkce, přibližně 6 uživatelů. Každý tajemník má vlastní heslo a „Supervisor“ – právo editace uživatelů.

4 IMPLEMENTACE DATOVÉHO MODELU

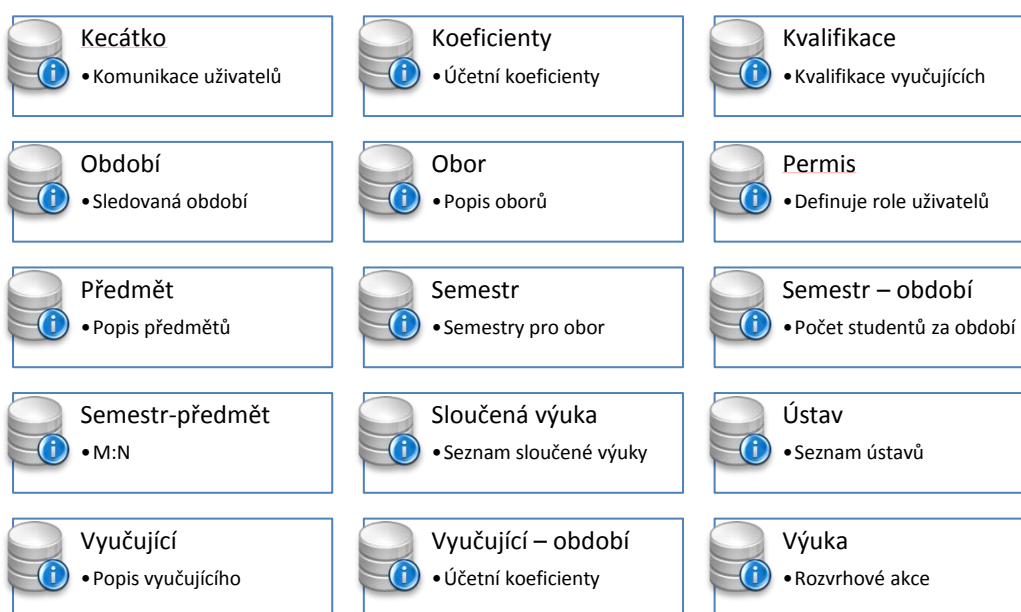
Následující kapitola popisuje implementaci v relačním databázovém systému Microsoft SQL Server

4.1 Datový model

Na obrázku 7 je zobrazeny relace mezi tabulkami bez detailního popisu sloupců pro každou tabulku. Šipka znázorňuje relaci 1:∞ (1 u šipky).



Obr. 7. Relační schéma databáze [16]



Obr. 8. Přehled tabulek

4.1.1 Tabulka „kecatko“

Tabulka slouží k zobrazení a uložení jednoduchých zpráv a vzkazů, umístěných na domovské stránce projektu. Systém je koncipován jako „shoutbox“, který můžeme vidět v různých webových aplikacích.

Tab. 8. Popis tabulky „kecatko“

Název sloupce	Datový typ	Popis
id	Int	PK, Auto inkrement
time	Datetime	Čas vložení zprávy
username	Nvarchar(50)	Jméno uživatele
role	Nvarchar(50)	Role uživatele
message	Nvarchar(150)	Zpráva, kterou uživatel napsal

4.1.2 Tabulka „koeficienty“

Slouží k uložení účetních koeficientů, které se používají pro výpočty *A*, *B*, *C*. Pro každé období by měli být definovány následující koeficienty:

- *c* počet započítatelných hodin za hodinu cvičení
- *klz* počet započítatelných hodin za udělený klasifikovaný zápočet
- *p* počet započítatelných hodin za hodinu přednášky
- *s* počet započítatelných hodin za hodinu semináře
- *z* počet započítatelných hodin za udělený zápočet
- *zk* počet započítatelných hodin za udělenou zkoušku

Tab. 9. Popis tabulky „koeficienty“

Název sloupce	Datový typ	Popis
id_obdobi	Int	PK, Odkaz na tabulku období
nazev	Nvarchar(50)	PK, Typ koeficientu
hodnota	float	Hodnota koeficientu

4.1.3 Tabulka „kvalifikace“

Tato tabulka uchovává informaci o tom, který vyučující je upřednostňován pro vytvoření úvazku pro daný předmět v daném období. Prakticky uchovává informace pro tabulku *editace/výuka*, aby při přiřazení výuky vyučujícímu byli upřednostněni vyučující, kteří jsou „kvalifikovaní“. Zároveň umožňuje vytvořit poznámku k dané kvalifikaci prostřednictvím sloupce poznámka.

Tab. 10. Popis tabulky „kvalifikace“

Název sloupce	Datový typ	Popis
id	Int	PK, Auto inkrement
id_vyucujici	Int	Odkaz na tabulku <i>vyucujici</i>
id_predmet	Int	Odkaz na tabulku <i>predmet</i>
poznámka	Nvarchar(200)	Poznámka k dané kvalifikaci
id_obdobi	Int	Odkaz na tabulku <i>obdobi</i>

4.1.4 Tabulka „období“

Slouží k uchování období, které databáze obsahuje. Pojmem období je myšlen kalendářní rok. Tabulka byla vytvořena pro zlepšení konzistence dat. Umožňuje správci vytvořit nové období a to dovolí tajemníkům vytvářet nové úvazky.

Tab. 11. Popis tabulky „období“

Název sloupce	Datový typ	Popis
id	Int	PK,Auto inkrement
rok	Int	Kalendářní rok

4.1.5 Tabulka „obor“

Popisuje obory pro jednotlivé fakulty. Vytvářet/modifikovat záznamy v této tabulce může pouze administrátor. Ostatní uživatelé mají právo pouze na čtení.

Tab. 12. Popis tabulky „obor“

Název sloupce	Datový typ	Popis
id	Int	PK,Auto inkrement
nazev	Nvarchar(100)	Název oboru
zkratka	Nvarchar(50)	Zkratka oboru
forma	Nvarchar(50)	Forma výuky oboru
typ	Nvarchar(50)	Typ výuky oboru
misto	Nvarchar(50)	Místo výuky oboru
fakulta	Nvarchar(50)	Fakulta oboru
cisloOboru	Nvarchar(50)	Katalogové číslo oboru
id_stag	int	ID oboru na stagu
planovat	bit	Informace o tom, jestli je možné pro obor plánovat výuku

4.1.6 Tabulka „permis“

Tato tabulka slouží pro definování práv uživatelů aplikace. Po úspěšné autentizaci s univerzitním LDAP serverem je potřeba načíst, v jaké roli vystupuje v této aplikaci daný uživatel. V případě, že uživatel je v roli tajemníka, je třeba načíst ústav, který spravuje.

Tab. 13. Popis tabulky „permis“

Název sloupce	Datový typ	Popis
username	Nvarchar(50)	PK, stagovský login
role	Nvarchar(50)	Role, v jaké vystupuje uživatel v této aplikaci
id_ustav	Int	Ústav, pro který má tajemník oprávnění vytvářet výuku (pokud je uživatel tajemník)

4.1.7 Tabulka „semestr“

Tabulka obsahuje seznam ročníků a semestrů daného oboru, pro které je možné vytvářet výuku.

Tab. 14. Popis tabulky „semestr“

Název sloupce	Datový typ	Popis
id	Int	PK, Auto increment
id_obor	Int	Odkaz na tabulku <i>obor</i>
rocnik	Int	Číslo ročníku
semestr	Nvarchar(50)	Informace, zda se jedná o letní nebo zimní semestr

4.1.8 Tabulka „predmet“

Tabulka slouží k uložení předmětu a informací o něm.

Tab. 15. Popis tabulky „predmet“

Název sloupce	Datový typ	Popis
id	int	PK, Auto increment
zkratka	Nvarchar(50)	Zkratka předmětu
nazev	Nvarchar(50)	Název předmětu
kredit	Int	Počet kreditů za předmět
id_garant	Int	Odkaz na tabulku <i>vyucujici</i>
ukonceni	Nvarchar(50)	Jakým způsobem je předmět ukončen
pocet_p	Int	Počet hodin přednášky
pocet_c	Int	Počet hodin cvičení
pocet_s	Int	Počet hodin seminářů
kapacita_c	Int	Počet studijních kroužků, které je možno vyučovat v jedné výukové akci cvičení
kapacita_s	Int	Počet studijních kroužků, které je možno vyučovat v jedné výukové akci semináře
tydnu	Int	Počet týdnů výuky
id_stag	Int	ID předmětu na stagu
jednotka_p	Nvarchar(50)	Jednotka hodin přednášky
jednotka_c	Nvarchar(50)	Jednotka hodin cvičení
jednotka_s	Nvarchar(50)	Jednotka hodin semináře
planovat	Bit	Informace, zda se má pro předmět plánovat výuka
id_ustav	Int	Odkaz na tabulku <i>ustav</i> , informace, pod který ústav předmět spadá

4.1.9 Tabulka „semestr“

Obsahuje informace o počtech studentů a počtech kroužků pro všechna plánovací období. Podle počtu skupin se generují výukové akce, zatímco podle počtu studentů se počítá účetní koeficient B .

Tab. 16. Popis tabulky „semestr“

Název sloupce	Datový typ	Popis
id	Int	PK, Auto increment
id_semestr	Int	Odkaz na tabulku <i>semestr</i>
id_obdobi	Int	Odkaz na tabulku <i>obdobi</i>
pocet_studentu	Int	Počet studentů pro daný semestr v daném období
pocet_krouzku	Int	Počet kroužků pro daný semestr v daném období

4.1.10 Tabulka „semestr_predmet“

Obsahuje informace o tom, ve kterém oboru a semestru se má daný předmět vyučovat.

Tab. 17. Popis tabulky „semestr_predmet“

Název sloupce	Datový typ	Popis
id	Int	PK, Auto increment
id_predmet	Int	Odkaz na tabulku <i>predmet</i>
id_semestr	Int	Odkaz na tabulku <i>semestr</i>
id_obodbi	Int	Odkaz na tabulku <i>obdobi</i>

4.1.11 Tabulka „sloucena_vyuka“

Tabulka obsahuje informace o sloučených výukách. Vyučuje-li vyučující 2 výukové akce ve stejný čas ve stejné místnosti, vytvoří se v této tabulce informace o sloučení. Toto sloučení je zohledněno při výpočtu úvazku.

Tab. 18. Popis tabulky „sloucena_vyuka“

Název sloupce	Datový typ	Popis
id	Int	PK, Číslo sloučení
id_vyuka	Int	PK, Odkaz na tabulku <i>vyuka</i>

4.1.12 Tabulka „ustav“

Tabulka obsahující seznam ústavů.

Tab. 19. Popis tabulky „ustav“

Název sloupce	Datový typ	Popis
id	Int	PK, Auto increment
nazev	Nvarchar(50)	Název ústavu

4.1.13 Tabulka „vyucujici_obdobi“

Obsahuje účetní informace pro všechna období pro daného vyučujícího. Obsahuje informaci o plánovaném úvazku, statickém koeficientu C a vztahu vyučujícího s univerzitou.

Tab. 20. Popis tabulky „vyucujici_obdobi“

Název sloupce	Datový typ	Popis
id	Int	PK, Auto increment
id_vyucujici	Int	Odkaz na tabulku <i>vyucujici</i>
id_obdobi	Int	Odkaz na tabulku <i>obdobi</i>
uvazek	Int	Plánovaný úvazek pro dané období
c	Int	Přepočítaný koeficient c
vztah	Nvarchar(50)	Vztah vyučujícího k univerzitě

4.1.14 Tabulka „vyucujici“

Obsahuje informace o vyučujících. Také obsahuje informaci o tom, pod jaký ústav vyučující spadá a také dodatečnou informaci, jaký jiný tajemník (z jiného ústavu) může vytvářet výuku pro daného zaměstnance.

Tab. 21. Popis tabulky „vyucujici“

Název sloupce	Datový typ	Popis
id	Int	PK, Auto increment
jmeno	Nvarchar(50)	Jméno vyučujícího
prijmeni	Nvarchar(50)	Příjmení vyučujícího
titul	Nvarchar(50)	Dosažené tituly vyučujícího
email	Nvarchar(50)	Email
id_stag	Int	Id vyučujícího na stagu
planovat	Bit	Informace, zda se má pro vyučujícího v daném období vytvářet výuka
telefon	Nvarchar(50)	Telefonní číslo
adresa	Nvarchar(50)	Adresa
id_ustav_planovaci	Int	Odkaz na tabulku <i>ustav</i> , informace, který další tajemník plánuje vyučujícímu výuku
id_ustav	Int	Odkaz na tabulku <i>ustav</i> , informace o ústavu, pod který tajemník spadá

4.1.15 Tabulka „vyuka“

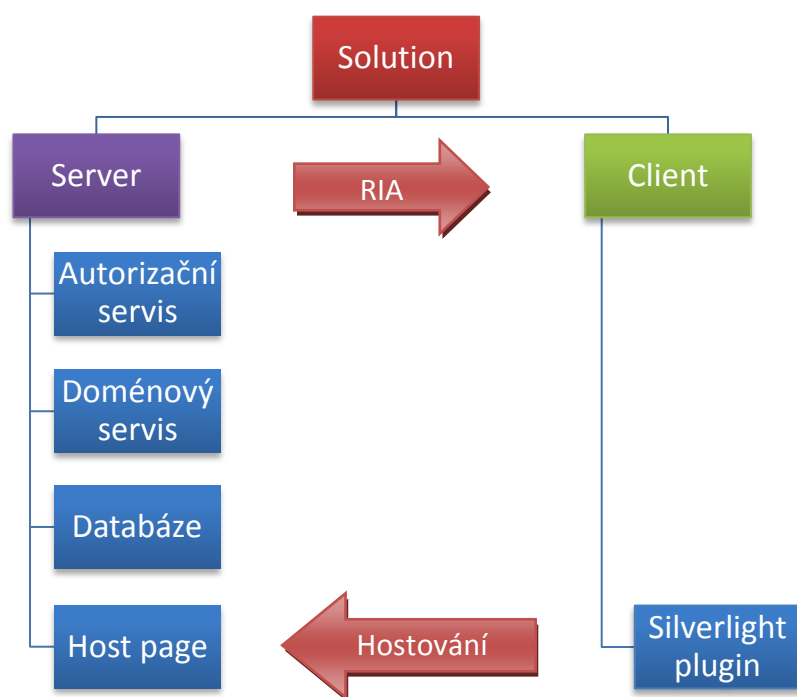
Popisuje plánované výukové akce. Obsahuje informace o tom, jaký předmět bude daný vyučující vyučovat v daném období a jestli se tato výuková akce započítává do účetního koeficientu *A*. Také obsahuje informaci o typu výukové akce (Přednáška, Cvičení, Seminář).

Tab. 22. Popis tabulky „vyuka“

Název sloupce	Datový typ	Popis
id	Int	PK, Auto increment
id_vyucujici	Int	Odkaz na tabulku <i>vyucujici</i>
typ	Nvarchar(50)	Typ výukové akce (p,c,s)
id_obdobi	Int	Odkaz na tabulku <i>obdobi</i>
id_semestr_predmet	Int	Odkaz na tabulku <i>semestr_predmet</i> , obsahuje informace o předmětu, oboru a ročníku dané výukové akce
zapocitavat	Bit	Informace o tom, zda se bude započítávat daná výuková akce do účetního koeficientu <i>A</i>

5 IMPLEMENTACE UŽIVATELSKÉHO ROZHRANÍ A APLIKAČNÍ LOGIKY

5.1 Struktura aplikace



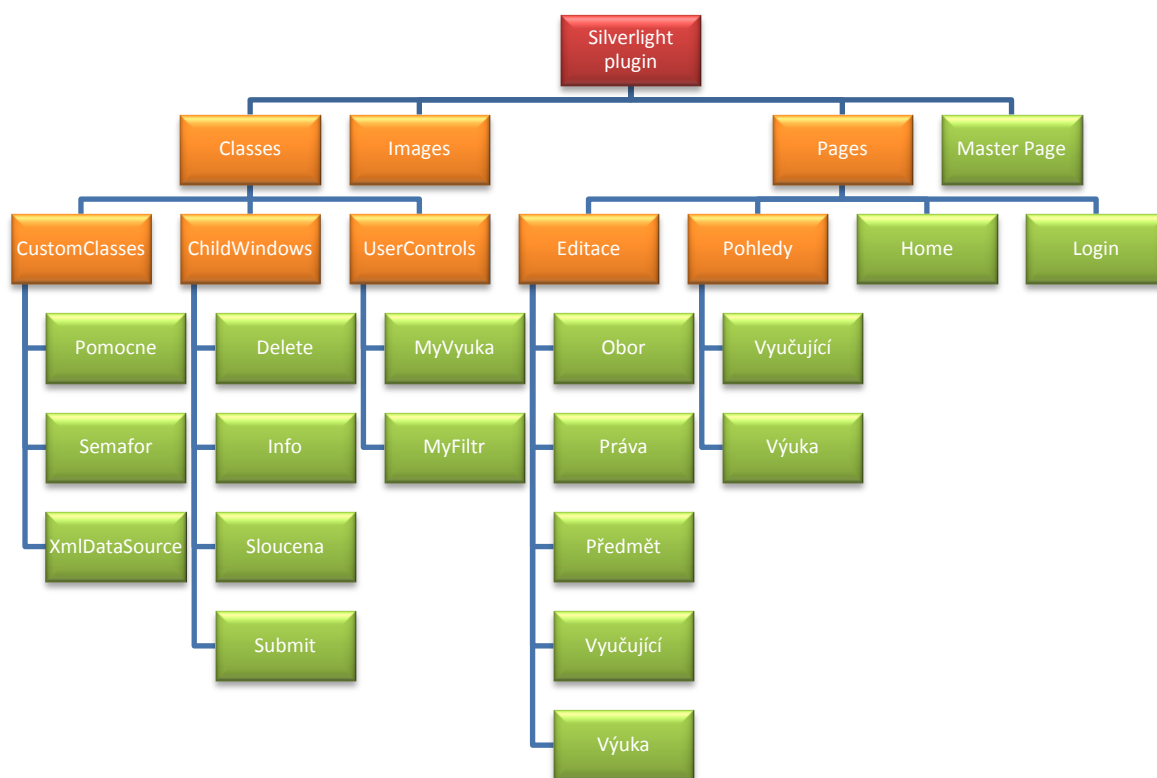
Obr. 9. Základní struktura aplikace

Celé řešení je rozděleno do 2 částí - klient a server. Jako klientskou část můžeme chápat Silverlight plugin, který běží kompletně na klientském PC. Tvorba Silverlight pluginu s použitím RIA služeb je velmi pohodlná, protože Visual Studio umožňuje v rámci vytváření aplikace debugovat zároveň Silverlight plugin a služby poskytované serverem. Také RIA automaticky kopíruje strukturu serverových tříd na klienta, což omezuje počet syntaktických chyb při vytváření aplikace. Spolu s třídami se do Silverlightu zkopírují také anotační atributy, což usnadní vytváření klientské části validace.

Serverová část aplikace se skládá ze samotné databáze, doménových služeb, autorizačních služeb a hostující stránky. Protože klient nemůže mít přímý přístup do databáze, je třeba použít autorizační služby pro zjištění identity a poté může klient volat doménové služby, které pracují s databází. Navíc je možnost, že server, který poskytuje služby, může zároveň hostovat silverlight plugin, čímž se stává řešení ucelenější.

5.1.1 Klient

Klientská část projektu je celý Silverlight plugin. Následující diagram popisuje strukturu, použitou při implementaci klientské části.



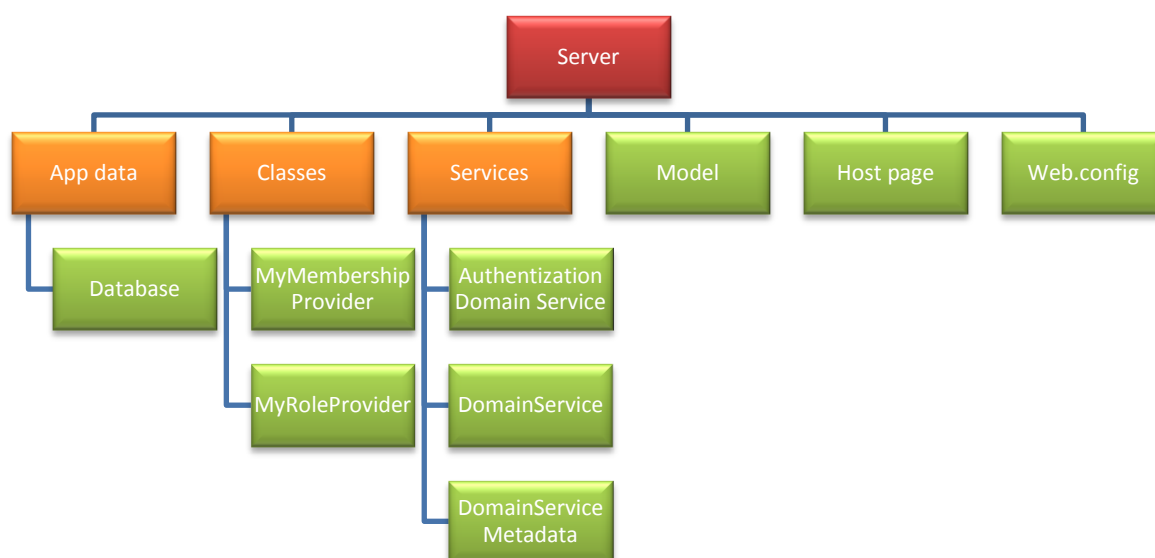
Obr. 10. Struktura klientské části projektu

Tab. 23. Popis objektů klientské části aplikace

Soubor	Stručný popis
Pomocne	Obsahuje pomocné třídy pro zobrazení v datagridech
Semafor	Třída pro synchronizaci asynchronně vrácených dat ze serveru
XmlDataSource	Třída pro načtení statických hodnot do comboboxů z xml souboru
Delete	Třída popisující dialogové okno zobrazené při mazání objektů
Info	Třída popisující dialogové okno zobrazující doplňující informace
Sloucena	Třída popisující dialogové okno, které se stará o sloučení výuky
Submit	Třída popisující dialogové okno při odesílání dat na server
MyVyuka	Třída popisující uživatelský prvek pro tvorbu výuky
MyFiltr	Třída popisující uživatelský prvek pro filtrování kolekcí
Master Page	Třída popisující hlavní okno aplikace
Pages/*	Třídy popisující konkrétní stránky aplikace

5.1.2 Server

Serverová část řešení poskytuje autorizační služby, doménové služby a webovou stránku, která hostuje samotný Silverlight plugin.



Obr. 11. Struktura serverové části projektu

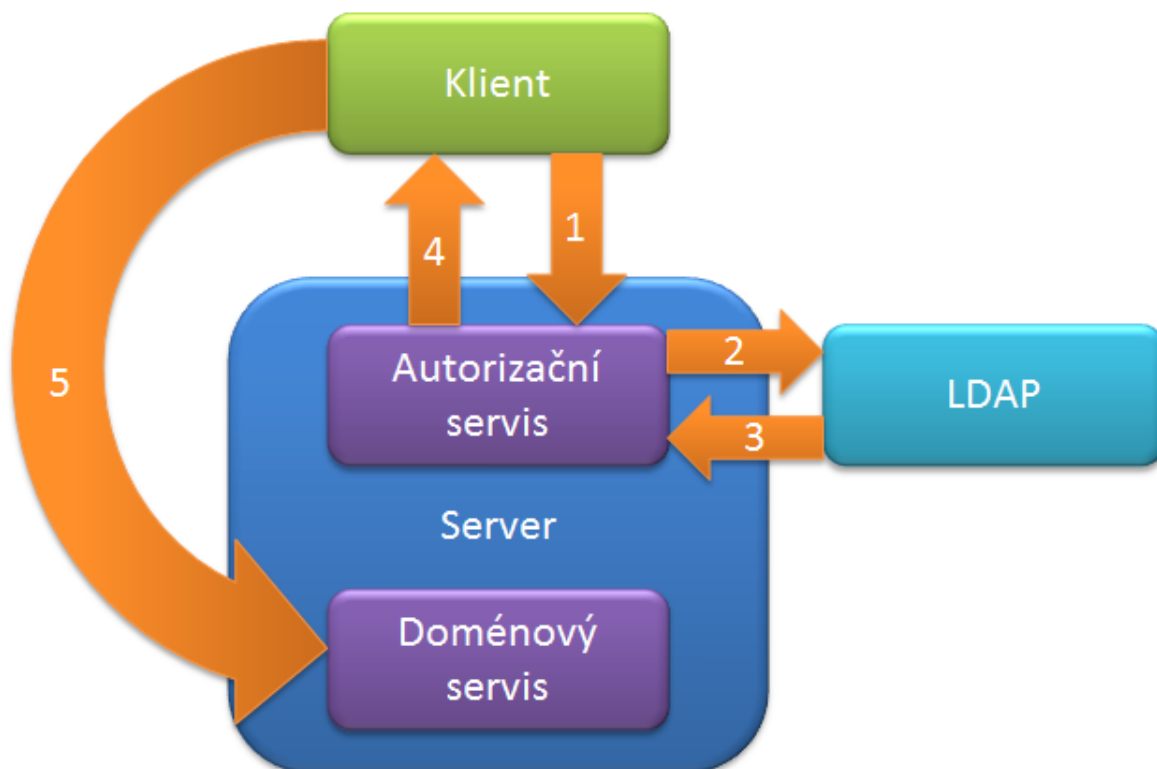
Tab. 24. Popis objektů serverové části aplikace

Soubor	Stručný popis
Database	Microsoft SQL Server Database soubor, kde jsou uložena všechna data
MyMembershipPrivider	Třída, která je zodpovědná za ověření uživatele přes LDAP
MyRoleProvider	Třída pro načtení uživatelské role
AutentizationDomainService	Třída, která se stará o veškerou autentizaci klienta
DomainService	Doménové služby pro komunikaci klienta s databází
DomainService.Metadata	Obsahuje třídy + anotaci těch tříd, které jsou použity v doménových službách
Model	ADO.NET Entity Data model
Host page	Stránka, která hostuje silverlight plugin
Web.config	Konfigurační soubor popisující chování celého serveru

5.2 Bezpečnost aplikace

5.2.1 Autorizace a autentizace

Aplikace používá ověření identity uživatele. Aplikaci tedy nemůže používat kdokoliv, ale jen student nebo zaměstnanec školy. Samotná autentizace probíhá tak, že uživatel zadá své přihlašovací údaje, které jsou zabezpečeně odeslány na server. Server se poté pokusí ověřit přihlašovací údaje pomocí LDAP serveru umístěného na univerzitě. V případě úspěšného ověření se aplikace pokusí načíst roli uživatele v rámci této aplikace z tabulky *permis*. Pokud uživatel nemá definovanou roli, je považován jako host.



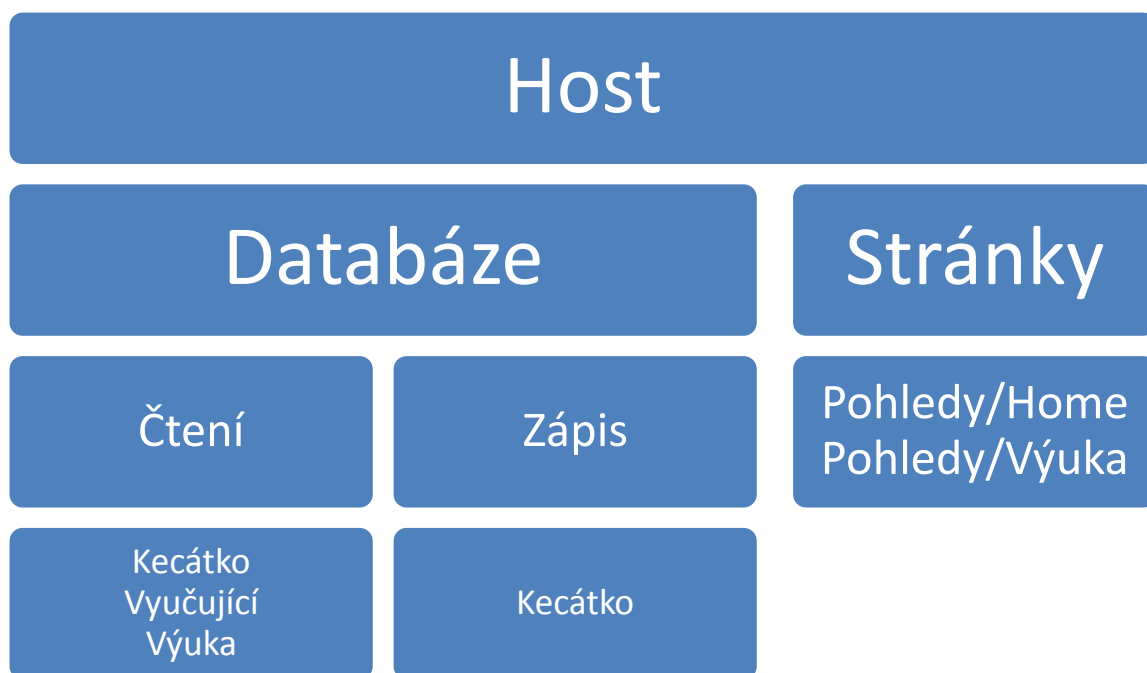
Obr. 12 Schéma autorizace a autentizace vytvořené aplikace

Aplikace je přizpůsobena pro 3 skupiny uživatelů:

- Host
- Tajemník
- Administrátor

Host

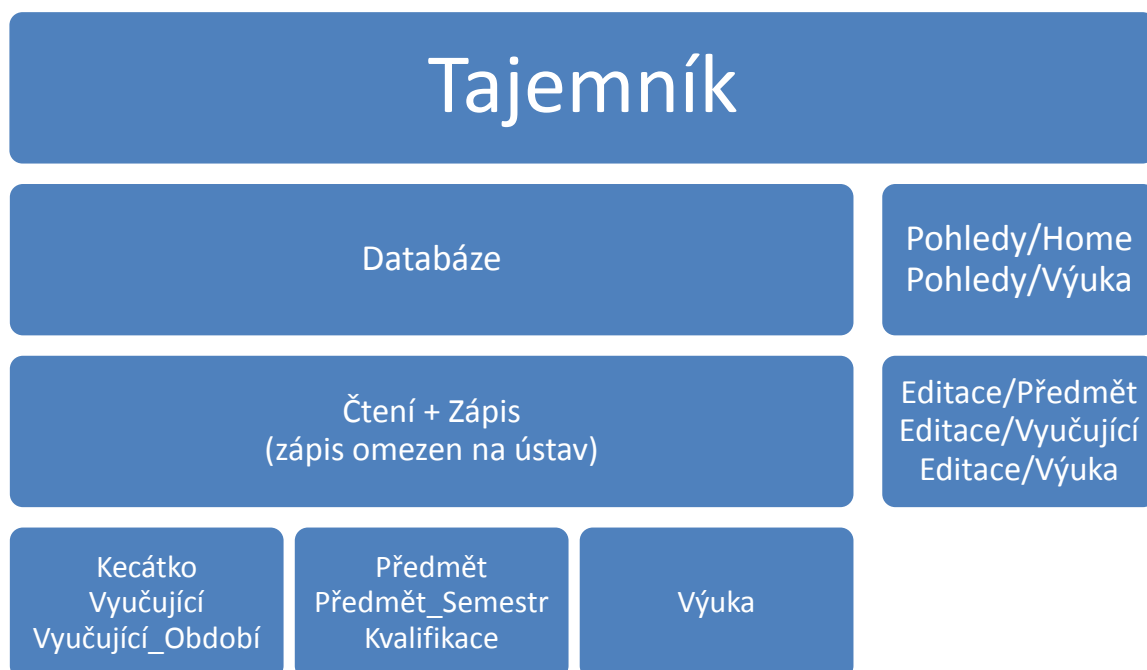
Hostem se myslí jakýkoliv subjekt, který projde ověřením skrze LDAP, ale nemá definovanou roli v této aplikaci.



Obr. 13. Práva uživatele Host

Tajemník

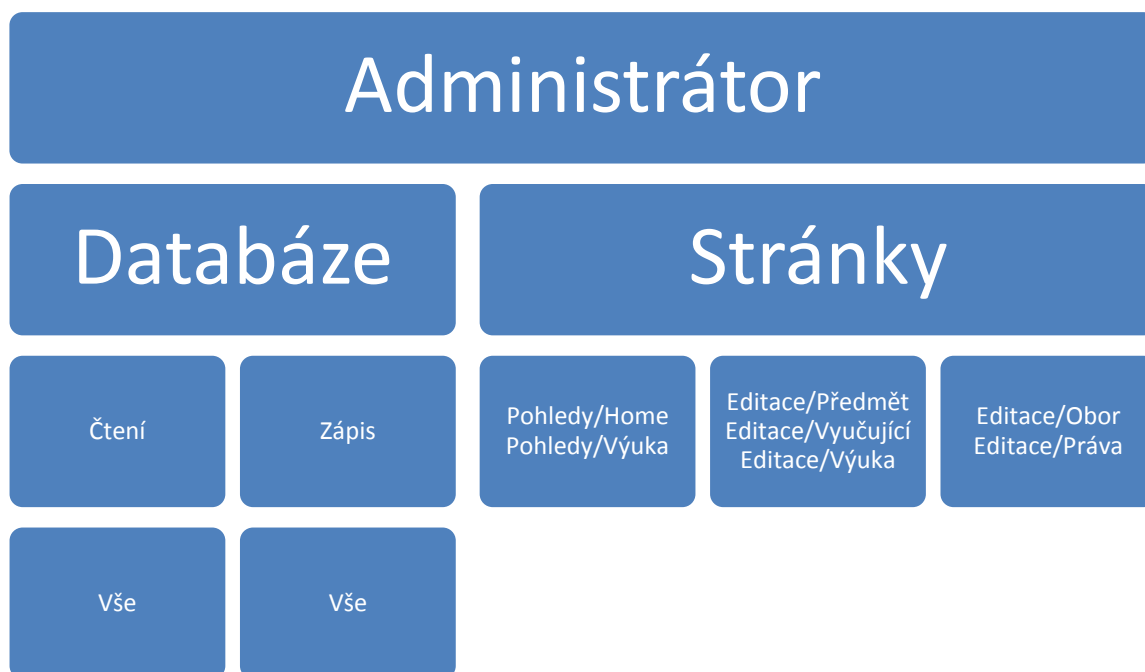
Tajemník je uživatel aplikace, který má právo vytvářet nové zaměstnance a předměty, které spadají pod jeho ústav. Dále může tajemník svým zaměstnancům definovat výuku. Každý tajemník může mít právo pouze pro jeden ústav.



Obr. 14. Práva uživatele tajemník

Administrátor

Administrátor je správce aplikace. Má právo měnit libovolnou tabulku. Určuje role ostatních uživatelů systému a má k dispozici důležité systémové funkce.



Obr. 15. Práva uživatele administrátor

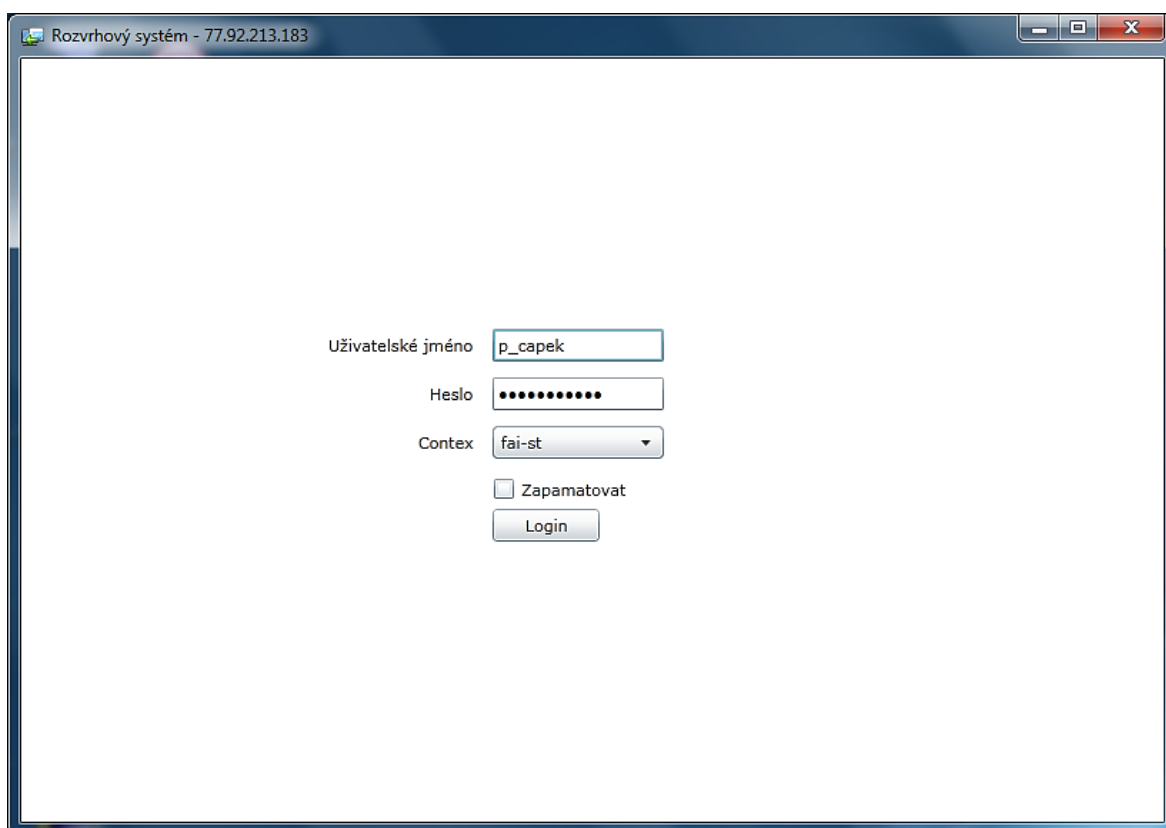
5.2.2 Zabezpečení připojení

K aplikaci je možné přistupovat pomocí protokolu http nebo zabezpečeně pomocí protokolu https. Pro použití zabezpečeného připojení je třeba na straně uživatele přijmout certifikát, aby mohlo být navázáno šifrované spojení. Šifrované spojení je realizováno pomocí SSL, tudíž není možnost spojení odposlouchávat či podvrhnout data.

5.3 Uživatelské prvky

5.3.1 Přihlašovací formulář

Pomocí tohoto formuláře se uživatel přihlašuje do systému. Zadá zde své uživatelské jméno a heslo (Stag, novel login). Kontext se vybere stejný jako u přihlašování Novelu. Zaškrtnutí pole *Zapamatovat* slouží k tomu, že pokud jej uživatel vybere a přihlásí se do aplikace a poté aplikaci uzavře (bez odhlášení), při příštím spuštění aplikace je automaticky přihlášen do systému bez nutnosti ověřování přes LDAP. Přináší to výhodu pohodlnějšího přístupu k aplikaci a možnosti používat aplikaci i v případě výpadku školního ověřovacího serveru. Zapamatování je prováděno pomocí standardní ASP cookies s délkou platnosti 7 dnů. Po přihlášení je uživatel přesměrován na stránku *Pohledy/Domů*.

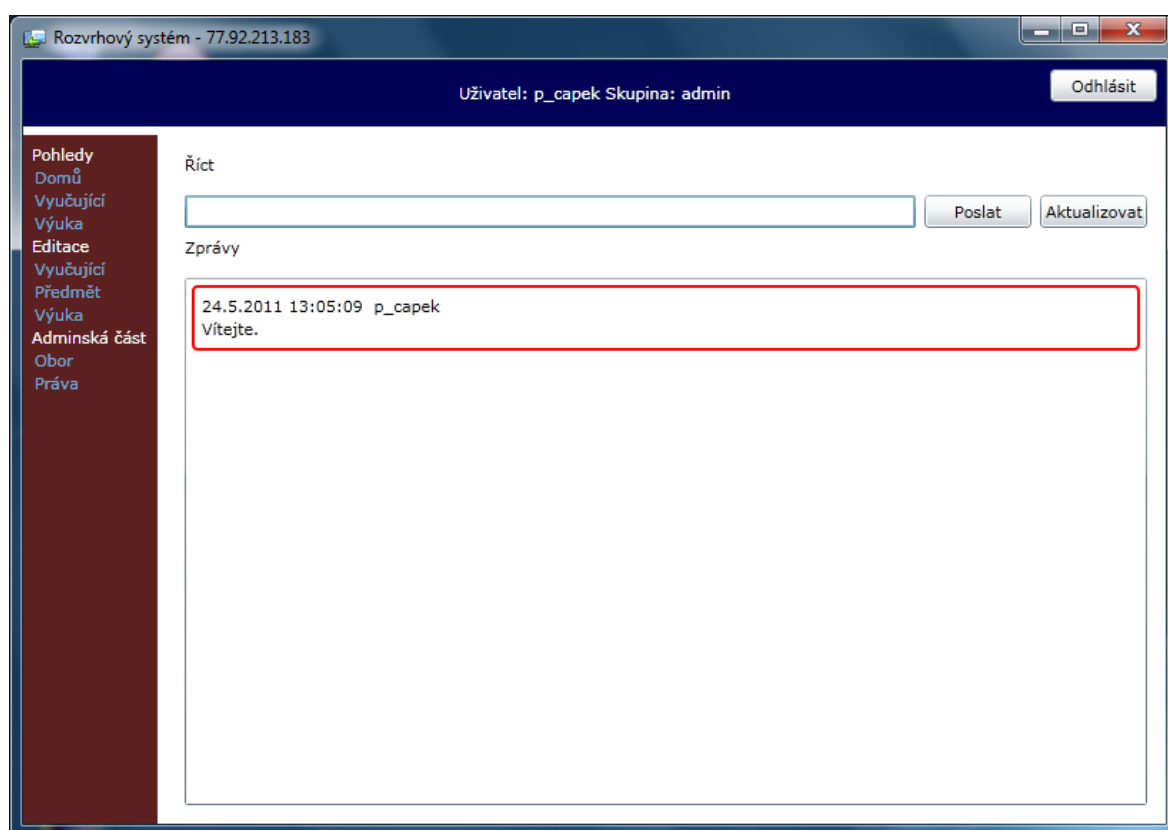


The screenshot shows a web browser window with the title "Rozvrhový systém - 77.92.213.183". The main content area displays a login form. The form has three input fields: "Uživatelské jméno" (Username) containing "p_capek", "Heslo" (Password) masked with dots, and "Contex" (Context) set to "fai-st" via a dropdown menu. Below these fields is an unchecked checkbox labeled "Zapamatovat" (Remember) and a "Login" button.

Obr. 16. Přihlašovací formulář

5.3.2 Zobrazení domovské stránky

Domovská stránka je koncipována jako tzv. „shoutbox“ (zjednodušená verze chatu). Při příchodu uživatele na stránku se vždy načte posledních 20 zpráv. Uživatel má možnost aktualizovat si seznam zpráv pomocí tlačítka *Aktualizovat*. Zprávy se vypisují ve formátu čas vložení příspěvku, uživatelské jméno a na další řádek samotná zpráva pro ostatní uživatele. Zprávy jsou barevně orámovány podle práv uživatele. Host má šedou barvu orámování, tajemník zelenou barvu a administrátor červenou barvu.



Obr. 17. Domovská stránka aplikace

5.3.3 Editace oborů

Tato stránka je rozdělena do dvou tabulek. V první tabulce je možné zobrazit informace o oborech a upravit tyto informace. Pomocí horních tlačítek je možné přidat či smazat obor. Smazat obor lze jen za předpokladu, že na něj nejsou vázány žádné semestry. Pokud si vybereme v horní tabulce obor, zobrazí se nám výpis semestrů patřících k danému oboru ve spodní tabulce. Poslední 2 sloupce spodní tabulky jsou aktuální, vždy podle daného období.

Rozvrhový systém - 77.92.213.183

Uživatel: p_capek Skupina: admin Odhlásit

Pohledy
Domů
Vyučující
Výuka
Editace
Vyučující
Předmět
Výuka
Adminská část
Obor
Práva

Zkratka	Název	Místo	Fakulta	Forma
knBT-tech	Bezpečnostní technologie, systémy a management - technická specializace	Zlín	FAI	Kombi
knIT	Informační technologie	Zlín	FAI	Kombi
pbBT	Bezpečnostní technologie, systémy a management	Zlín	FAI	Prezen
pbIRT	Informační a řídicí technologie	Zlín	FAI	Prezen
pbITA	Informační technologie v administrativní praxi	Zlín	FAI	Prezen
pnARI	Automatické řízení a informatika	Zlín	FAI	Prezen
pnBT	Bezpečnostní technologie, systémy a management	Zlín	FAI	Prezen

Přidat
Smazat
Odeslat

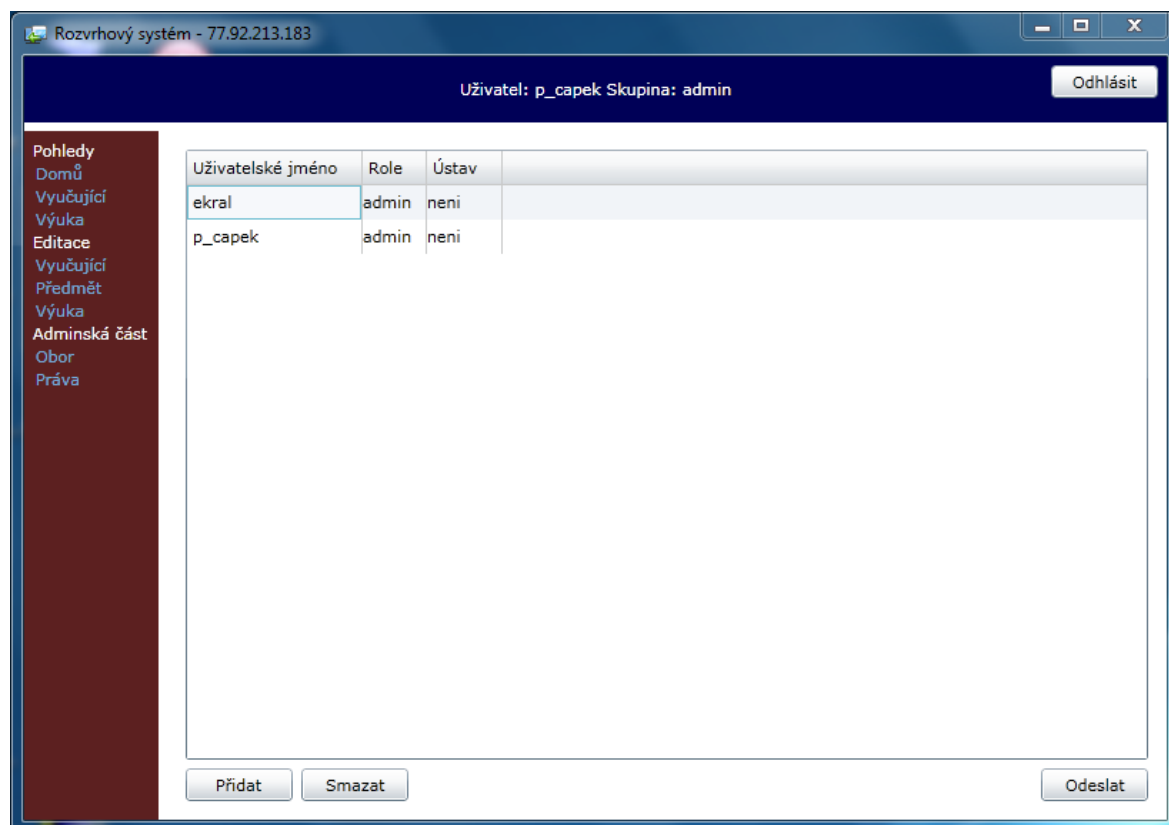
Ročník	Semestr	Počet studentů	Počet kroužků
1	LS	168	12
1	ZS	168	14
2	LS	101	8
2	ZS	101	8
3	LS	74	8
3	ZS	74	8

Přidat
Smazat

Obr. 18. Editace oborů

5.3.4 Editace práv

Editace/Práva slouží k definování rolí uživatelů v tomto systému. Jenom administrátor může měnit role uživatelů v systému. Role se vytvoří tak, že do pole *Uživatelské jméno* se zadá nové uživatelské jméno a v seznamu rolí se vybere příslušná role. Poslední sloupec má význam jen tehdy, pokud je role uživatele *Tajemník*, jinak je hodnota v tomto poli ignorována.



Obr. 19. Editace práv

5.3.5 Editace předmětů

Tato stránka slouží k editaci informací o předmětu. Je rozdělena do 3 tabulek. Horní největší tabulka zobrazuje všechny předměty z ústavu tajemníka. V této tabulce jsou důležité sloupce *Kapacita s* a *Kapacita c*, což jsou koeficienty vyjadřující, kolik 12cti členných skupin je možné vyučovat na jedné výukové akci semináře nebo cvičení.

Spodní levá tabulka *Kvalifikace* slouží k editaci kvalifikovaných vyučujících k vybranému předmětu. Pole *Poznámka* je zobrazeno při tvorbě výuky na straně Editace/Výuka. *Kvalifikace* je zobrazená vždy pro aktuální období.

Spodní pravá tabulka *Semestry* slouží k definování v jakém semestru a pro jaký obor se bude daný předmět vyučovat.

Rozvrhový systém - 77.92.213.183

Uživatel: p_capek Skupina: admin Odhlásit

Pohledy
Domů
Vyučující
Výuka
Editace
Vyučující
Předmět
Výuka
Adminská část
Obor
Práva

Zkratka	Název	Garant	Ústav	Týdnů	Počet p	Počet s	Počet c
A2ADS	Algoritmy a datové struktury	Dulík Tomáš	AUIUI	14	2	0	2
A2ARP	Architektura počítačů	Sysel Martin	AUPKS	14	2	0	1
A2EMV	Elektřina, magnetismus a vlnění	Schauer František	AUEM	14	2	2	2
A2IAM	Instrumentace a měření	Křesálek Vojtěch	AUEM	14	2	0	2
A2MAT	Matematika II	Fialka Miloslav	AUM	14	2	3	0
A2MZS	Mechanické zábranné systémy	Ivanka Ján	AUBI	14	2	0	1
A2POE	Podniková ekonomika	Kuncová Jana	AUBI	14	2	1	1

Přidat
Smazat
Odeslat

Kvalifikace

Vyučující	Poznámka
Bělašková Silvie	
Fialka Miloslav	
Chudá Hana	
Ostravský Jan	
Polášek Vladimír	
Řezníčková Jana	

Přidat
Smazat

Semestr

Semestry
Obor: pbBT Rocnik: 1 Semestr: LS
Obor: pbIRT Rocnik: 1 Semestr: LS

Přidat
Smazat

Obr. 20. Editace předmětů

5.3.6 Editace vyučujících

Pomocí této stránky je možné zadávat a upravovat informace o vyučujících. Důležité jsou sloupce *Úvazek* a *C*, které jsou zobrazeny vždy pro aktuálně dané období. *Úvazek* se myslí plánovaný úvazek pro dané období a slouží při plánování výuky pro zobrazení procentuálního naplnění úvazku vyučujícího. *C* je statický koeficient, který se započítává do úvazku.

Rozvrhový systém - 77.92.213.183

Uživatel: p_capek Skupina: admin Odhlásit

Pohledy
Domů
Vyučující
Výuka
Editace
Vyučující
Předmět
Výuka
Adminská část
Obor
Práva

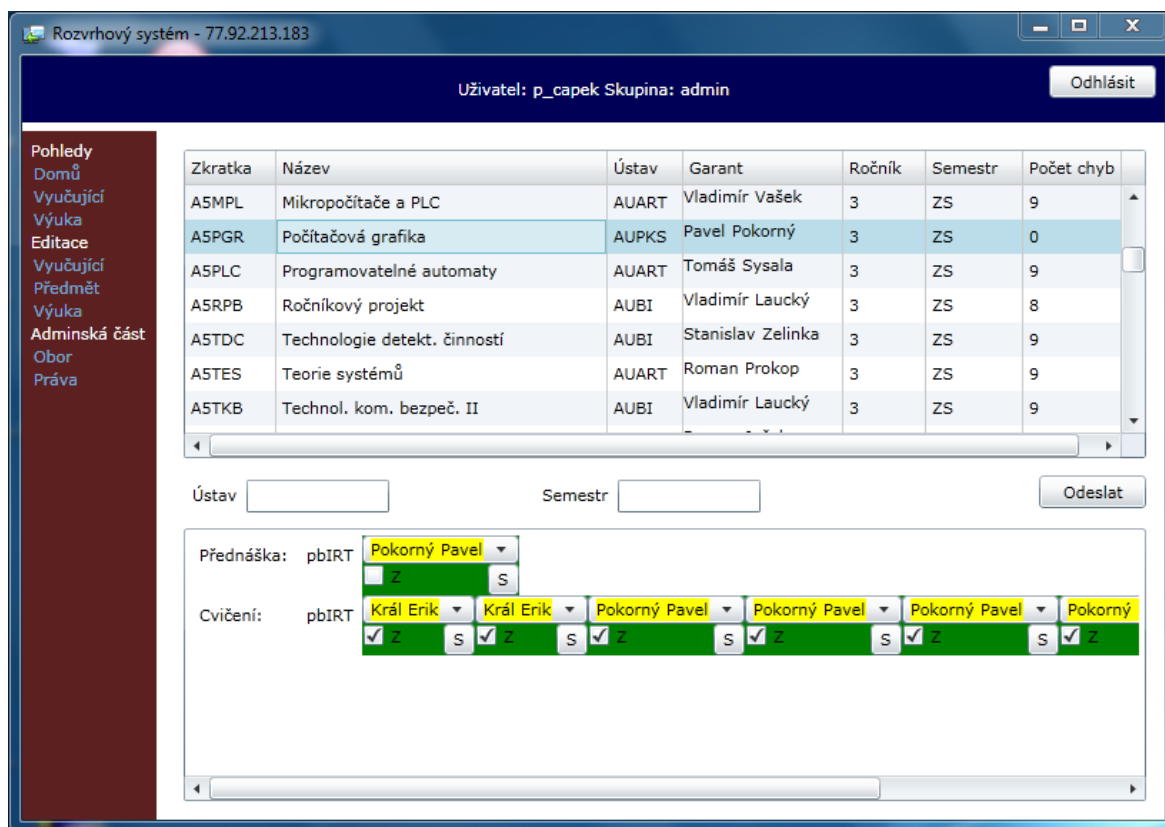
Jméno	Příjmení	Titul	Úvazek	C	Vztah	Ústav	Email	Telefon	Adresa	Ústa
Milan	Adámek	doc. Mgr., Ph.D.	900	30	Interní	AUBI				AUBI
Zdeněk	Babík	Ing.,	900	30	Doktorand	AURP				AURP
Hana	Báčová	Ing.,	900	30	Doktorand	UAI				UAI
Monika	Bakošová	doc. Ing., CSc.	900	30	Interní	UAI				UAI
Jaroslav	Balátě	prof. Ing., DrSc.	900	30	Interní	AUBI		5		AUBI
Tomáš	Barot	Ing.,	900	30	Doktorand	AURP				AURP
Michal	Bavšenkov	JUDr.,	900	30	Interní	AUEM				AUEM
František	Bednařík	, Ph.D.	900	30	Interní	AUM				AUM
Ladislav	Běhal	Ing.,	900	30	Doktorand	AUPKS				AUPK
Silvie	Bělašková	Mgr.,	900	30	Doktorand	AUM				AUM
Martin	Beneda	Ing.,	900	30	Doktorand	AUBI				AUBI
Václav	Bezděk	Mgr.,	900	30	Doktorand	AUM				AUM
Karel	Bílek	Ing.,	900	30	Doktorand	AUEM				AUEM
Radovan	Binder	Ing.,	900	30	Doktorand	UAI				UAI
Michal	Bližňák	Ing., Ph.D.	900	30	Interní	AUIUI				AUIU
Vladimír	Bobál	prof. Ing., CSc.	900	30	Interní	AURP				AURP

Přidat
Smazat
Odeslat

Obr. 21. Editace vyučujících

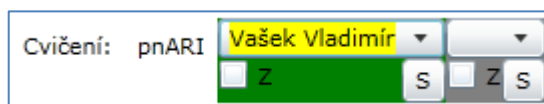
5.3.7 Editace výuky

Asi nejdůležitější stránka v aplikaci. Na této stránce je možné vytvářet výuku vyučujícím. V horní tabulce jsou zobrazeny předměty, pro které je možné vytvářet výuku.



Obr. 22. Editace výuky

Zajímavý je sloupec *Počet chyb*, který zobrazuje, kolik výukových akcí v daném předmětu nebylo obsazeno, či kolik výukových akcí přebývá. V dolní části okna jsou vygenerovány výukové akce pro předmět vybraný v horní tabulce. Výukové akce se počítají z počtu skupin oboru a kapacity seminářů/cvičení. Aplikace dynamicky reaguje na zvýšení/snížení počtu skupin oborů v průběhu semestru. V případě snížení počtu skupin dojde k označení přebytečné výuky červeným křížkem, což umožňuje odstranit přebytečnou výuku. Při vytváření výuky si uživatel vybere typ výukové akce, kterou chce obsadit. K dispozici jsou přednáška, cvičení a seminář. Dále určí obor, pro který je výuková akce určena. Touto selekcí dostaneme řádek ovládacích prvků pro tvorbu výuky. Na řádku se mohou nacházet šedě podbarvené (neobsazená výuková akce) a zeleně podbarvené komponenty.



Obr. 23. Uživatelský prvek výuka

Samotná komponenta pro tvorbu výuky je složena s ovládacích prvků rolety, zaškrťovacího pole a Tlačítka.

Po rozbalení rolety se zobrazí seznam zaměstnanců, kterým může tajemník plánovat výuku.

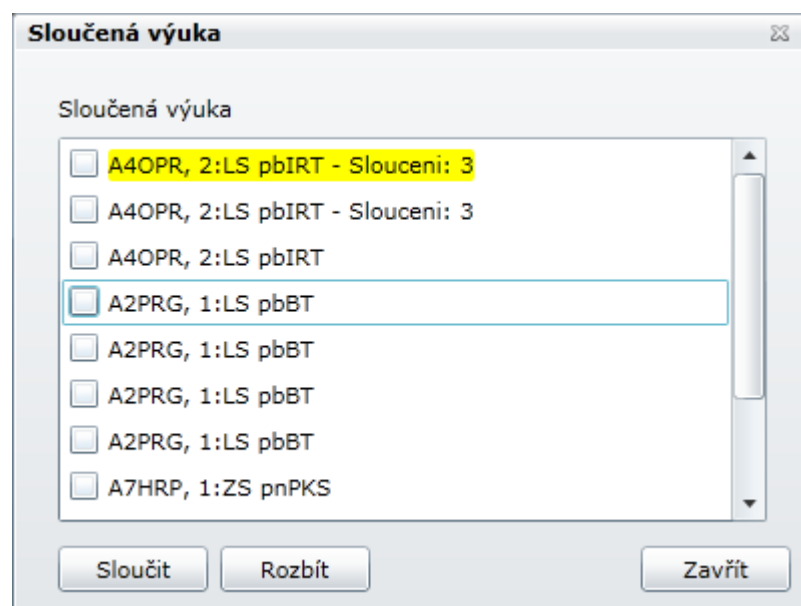


Obr. 24. Uživatelský prvek
pro volbu vyučujícího

Seznam je rozdělen na 2 části. Horní část je podbarvena žlutě a je pro každý předmět jiná. Žlutě podbarvení zaměstnanci se načítají z vytvořených kvalifikací předmětu, což znamená, že jsou nejvhodnější pro obsazení výuky daného předmětu. Zbytek seznamu je tvořen zaměstnanci, které může tajemník plánovat. Po přesunu kursoru na jméno se spočítá a zobrazí aktuální úvazek zaměstnance spolu s počtem naplánovaných hodin v letním a zimním semestru.

Dále je v komponentě pro tvorbu výuky zaškrťovací pole Z, které ovlivňuje, zda se výuková akce započítává do koeficientu B.

Poslední ovládací prvek komponentu je tlačítko *S*, které umožňuje sloučit výukovou akci s jinou výukovou akcí daného vyučujícího.



Sloučená výuka

Sloučená výuka

- ☐ A4OPR, 2:LS pbIRT - Sloučení: 3
- ☐ A4OPR, 2:LS pbIRT - Sloučení: 3
- ☐ A4OPR, 2:LS pbIRT
- ☒ A2PRG, 1:LS pbBT
- ☐ A2PRG, 1:LS pbBT
- ☐ A2PRG, 1:LS pbBT
- ☐ A2PRG, 1:LS pbBT
- ☐ A7HRP, 1:ZS pnPKS

Sloučit Rozbít Zavřít

Obr. 25. Formulář sloučené výuky

5.3.8 Zobrazení vyučujících

Tato strana je nejdůležitější pro uživatele, kteří jsou přihlášení jako host. Pokud je uživatel vyučující, může si jednoduše zadat své jméno do stránky a systém mu vypíše plánovaný úvazek, předměty, pro které je vhodné, aby je vyučoval, předměty, které garantuje a nakonec jsou vypsány i předměty, které by měl vyučovat. U vypsání výuky jednotlivých předmětů je zobrazeno v jakém semestru, v jakém oboru je naplánována výuka. Dále je zde uveden počet skupin, jež má vyučující vyučovat, délka trvání výukové akce a předběžný počet studentů, kterým bude udělovat zápočet/zkoušku.

Rozvrhový systém - 77.92.213.183

Uživatel: p_capek Skupina: admin Odhlásit

Pohledy
Domů
Vyučující
Výuka
Editace
Vyučující
Předmět
Výuka
Adminská část
Obor
Práva

Zadejte přímení:

Úvazek %	A	B	C	h/t v ZS	h/t v LS	Pl. úvazek
117,6	812	47,5	199	8	21	900

Kvalifikace

Zkratka	Název
A2PRG	Programování
A4OPR	Objektové programování
A7HRP	Hradlová pole

Garant

Zkratka	Název

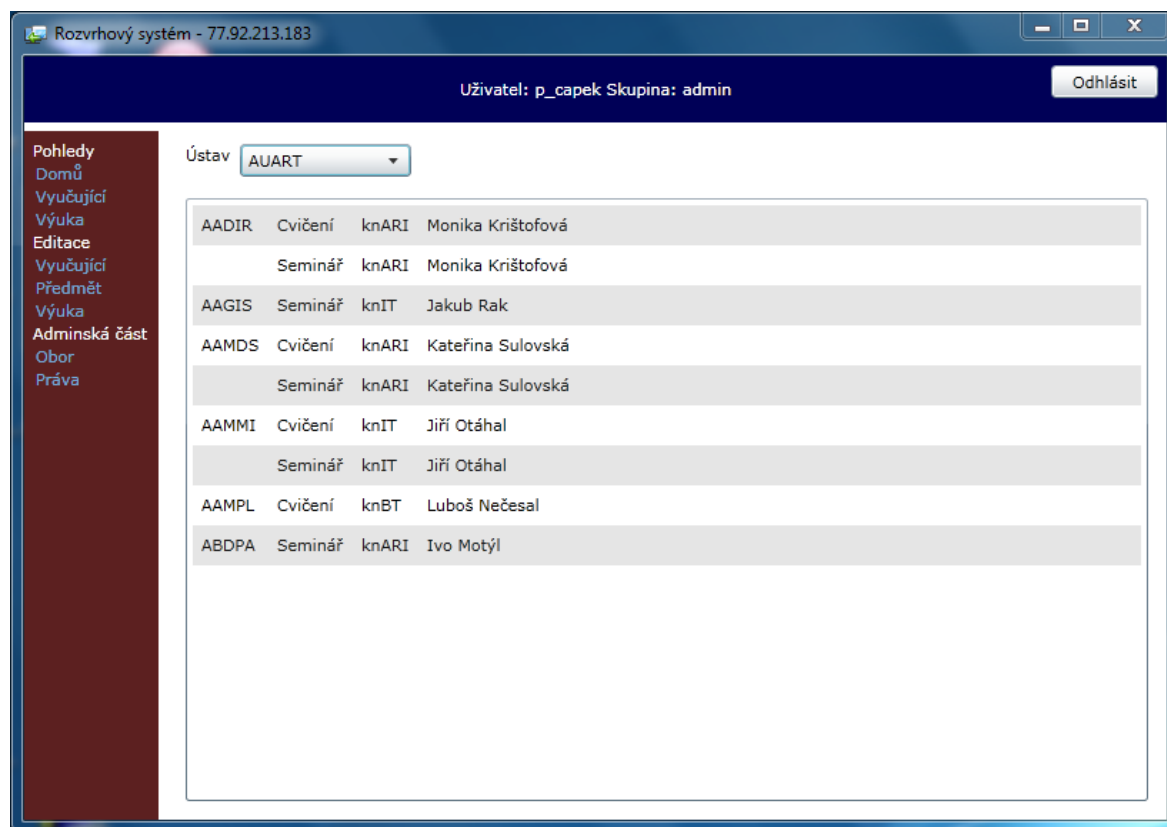
Výuka

Zkratka	Název	Obor	Ročník	Semestr	Typ	Počet skupin	Délka v h/t	Počet zkoušek
A4OPR	Objektové programování	pbIRT	2	LS	c	3	3	0
A2PRG	Programování	pbBT	1	LS	c	4	3	0
A7HRP	Hradlová pole	pnPKS	1	ZS	c	2	2	0
A5PGR	Počítačová grafika	pbIRT	3	ZS	c	2	2	0

Obr. 26. Zobrazení vyučujících

5.3.9 Zobrazení výuky

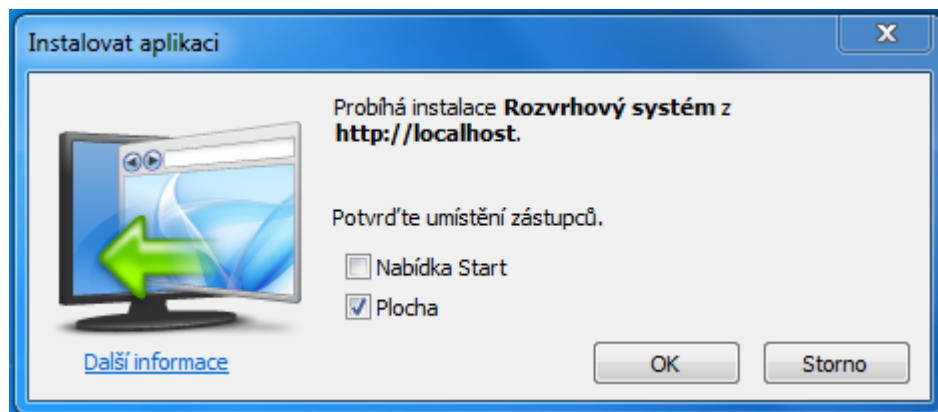
Pohledy/Výuka je zjednodušené zobrazení *Editace/Výuka* přístupné všem hostům. Zobrazuje předměty pro jednotlivé ústavy a také, kteří vyučující budou vyučovat daný předmět.



Obr. 27. Zobrazení výuky

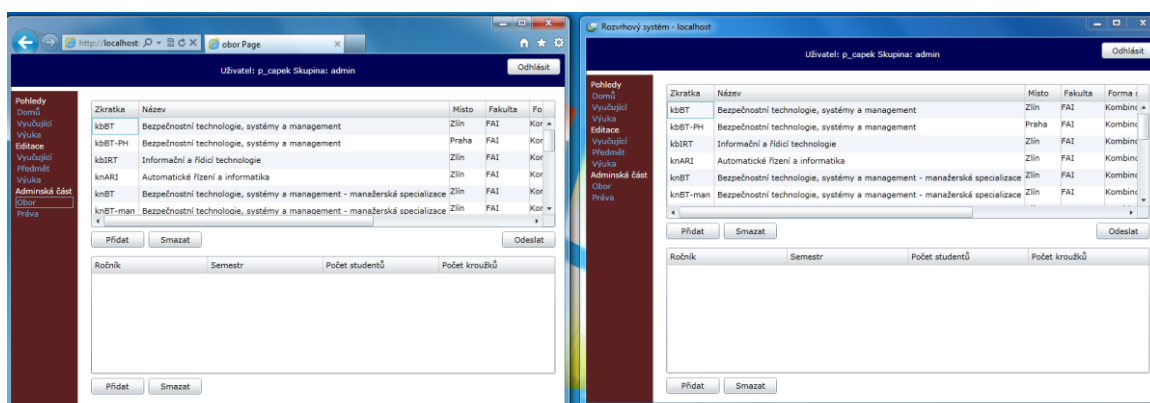
5.4 Desktopová aplikace

Silverlight podporuje tzv. Out of browser, což je běh Silverlight pluginu bez prohlížeče. Tato aplikace je nakonfigurována pro využití této vlastnosti Silverlightu. Je nutné ji nejprve nainstalovat do počítače. To se provede tak, že kdekoli v aplikaci je možné v kontextovém menu vybrat možnost „Instalovat Rozvrhový systém do tohoto počítače ...“.



Obr. 28. Instalace „Out of browser“ aplikace

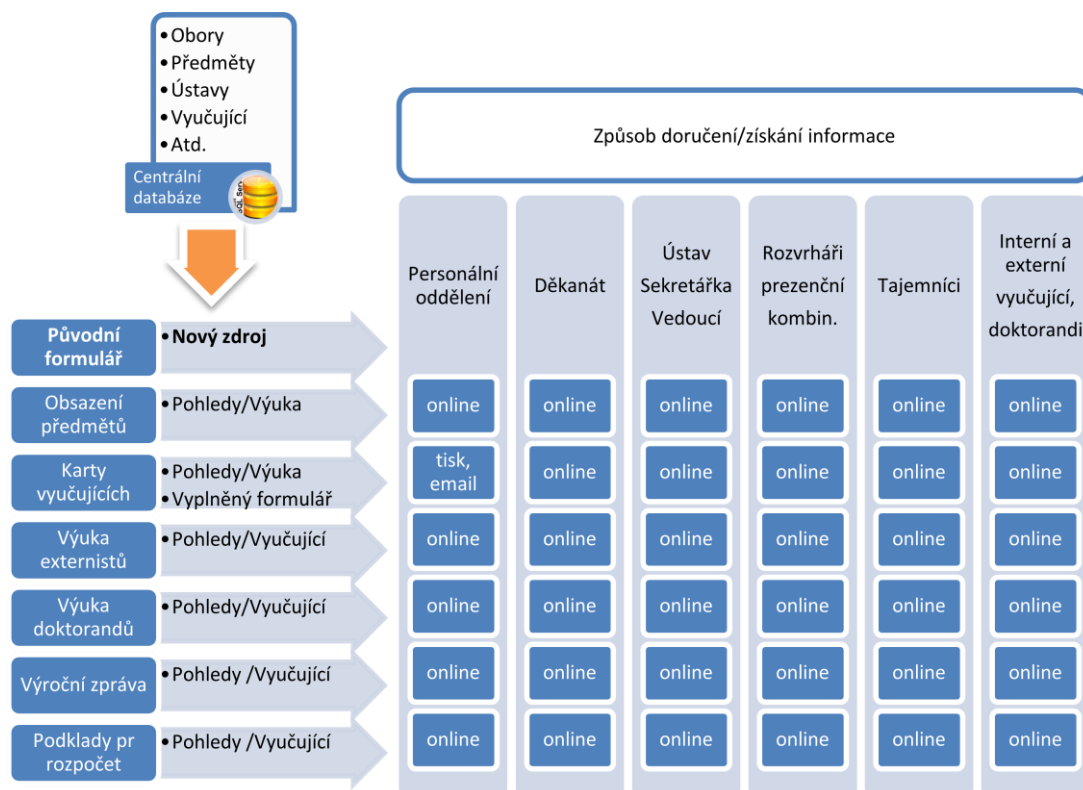
Po té je zobrazeno dialogové okno o možnosti umístění zástupců aplikace na plochu a nabídky start. V dalším kroku se provede samotná instalace aplikace do počítače. Nyní je možné aplikaci spustit mimo webový prohlížeč.



Obr. 29. Okno aplikace v prohlížeči a okno „Out of browser“ aplikace

6 ZHODNOCENÍ PŘÍNOSU APLIKACE

Výchozí stav je zobrazen na obrázku 6 a na obrázku 30 je zobrazený stav podpory řízení, správy a zjišťování aktuálního stavu rozvrhované výuky při využití vytvořené aplikace. Výrazným způsobem se ulehčí práce tajemníků a dalších zaměstnanců, kdy většinu času trávili ručním přepisováním údajů z jednoho formuláře do druhého. Nezanedbatelná je i úspora papíru a omezení chyb způsobených decentralizovaným přístupem.



Obr. 30. Zjednodušení procesu

ZÁVĚR

Cílem práce bylo vytvoření informačního systému pro podporu řízení, správu a zjišťování aktuálního stavu rozvrhované výuky. Tento systém momentálně na Univerzitě Tomáše Bati ve Zlíně (UTB) neexistuje a jeho implementace výrazně ulehčí práci zaměstnancům UTB. Systém umožňuje přiřazování výukových jednotek vyučujícím na základě informací o počtu studijních skupin v jednotlivých studijních oborech a programech. Součástí systému je vytváření přehledu o výuce jednotlivých vyučujících a předmětech.

Systém splňuje zadané požadavky. Uživatelské rozhraní je realizované v aplikační platformě Microsoft Silverlight s využitím technologie Windows Communication Foundation Rich Internet Application (WCF RIA), objektově relačním mapování Entity Framework a integrovaném jazyku pro dotazování LINQ. Datový model je implementovaný v relačním databázovém systému Microsoft SQL Server.

Práce je koncipována tak, aby ji bylo možné v budoucnosti rozšířit o další funkce, jako je doplněk pro prostředí Microsoft Excel pro automatickou tvorbu formulářů [23], automatické obsazování předmětů podle zadaných pravidel, rozšíření o základní tvorbu rozvrhů, verze optimalizovaná pro mobilní zařízení a dotykové obrazovky a vytváření tiskových sestav.

Aplikace byla použita při tvorbě úvazků na Ústavu bezpečnostního inženýrství Univerzity Tomáše Bati.

ZÁVĚR V ANGLIČTINĚ

The aim of this work is design and realization of information system for computer aided course planning and scheduling. This system does not exist on Thomas Bata University in Zlín yet. This system shall significantly simplify administrative work at UTB. Using this system, teaching units can be assigned to the lecturers knowing the number of group of students and syllabus. System includes online views with information about lecturers and subjects.

The designed system meets given requirements. User interface is implemented using Microsoft Silverlight application platform, Windows Communication Foundation Rich Internet Application (WCF RIA), Entity Framework object-relation mapping and Language-Integrated Query (LINQ). Data model is implemented in Microsoft SQL relational model database server.

System is designed to be extendable, for example there is a Microsoft Excel add-in for automatic form creation [23]. There are planned features as automatic scheduling, optimization for touch devices and print views creation.

The result of this work was used for course planning and scheduling at Department of Security Engineering of Thomas Bata University.

SEZNAM POUŽITÉ LITERATURY

- [1] *Msdn* [online]. 2011 [cit. 2011-05-28]. WCF RIA Services. Dostupné z WWW: <[http://msdn.microsoft.com/en-us/library/ee707344\(VS.91\).aspx](http://msdn.microsoft.com/en-us/library/ee707344(VS.91).aspx)>
- [2] *Msdn* [online]. 2011 [cit. 2011-05-28]. Entity Framework Overview. Dostupné z WWW: <<http://msdn.microsoft.com/en-us/library/bb399567.aspx>>
- [3] *Msdn* [online]. 2011 [cit. 2011-05-28]. Language-Integrated Query (LINQ). Dostupné z WWW: <<http://msdn.microsoft.com/library/bb397926.aspx>>
- [4] *Msdn* [online]. 2011 [cit. 2011-05-28]. SQL Server. Dostupné z WWW: <<http://msdn.microsoft.com/en-us/sqlserver/default.aspx>>
- [5] BILL EVJEN, Christian Nagel, et al. *C# 2008 Programujeme profesionálně*. Praha : Computer Press, 2009. 1904 s. ISBN 978-80-251-2401-7.
- [6] LAIR, Robert. *Beginning Silverlight 4 in C#*. [s.l.] : Apress, 2010. 395 s. ISBN 978-1-4302-2988-9.
- [7] FAROULT, Stéphane; ROBSON, Peter. *The Art of SQL*. Sebastopol : O'Reilly Media, Inc, 2006. 349 s. ISBN 978-0-596-00894-9.
- [8] HOLZNER, Steve. *Design Patterns For Dummies*. Indianapolis : For Dummies, 2006. 308 s. ISBN 978-0-471-79854-5.
- [9] Microsoft Developer Network [online]. 2011 [cit. 2011-01-31]. *Windows Communication Foundation*. Dostupné z WWW: <[http://msdn.microsoft.com/en-us/library/ms735119\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms735119(v=vs.90).aspx)>
- [10] WCF PRO ZAČÁTEČNÍKY – 2. DÍL: CONTRACT, BINDING, SERVICE BEHAVIOR. *.netstudent* [online]. 13.2.2009, [cit. 2011-05-31]. Dostupný z WWW: <<http://www.netstudent.cz/Articles/wcf-pro-zacatecniky-2-dil-contract-binding-service-behavior>>
- [11] *Asynchronous vs Synchronous Silverlight* [online]. 30.12.2010 [cit. 2011-05-31]. .NET Framework Developer Center. Dostupné z WWW: <<http://social.msdn.microsoft.com/Forums/en-US/wcf/thread/a8189fe2-c999-47c7-a81f-8174554255eb>>
- [12] *Silverlight 4 Overview*. [s.l.] : Microsoft, 2010. 64 s. Dostupné z WWW: <<http://download.microsoft.com/documents/uk/mediumbusiness/products/silverlight/WhatsNewInSilverlight4.pdf>>

- [13] ANDERSON, Chris. *Pro Business Applications with Silverlight 4*. United States of America : Apress, 2010. 576 s. ISBN 978-1-4302-7207-6.
- [14] Úvod do LINQ. *Vyvojar.cz* [online]. 28.1.2008, [cit. 2011-05-31]. Dostupný z <WWW: <http://www.vyvojar.cz/Articles/563-uvod-do-linq.aspx>>
- [15] *Vyvojar.cz* [online]. 7.5.2008 [cit. 2011-05-31]. ADO.NET Entities: 1. část - k čemu je to dobré?. Dostupné z WWW: <http://blog.vyvojar.cz/mjurek/archive/2008/04/07/ADO.NET-Entities_3A00_-1.-cast_-2D00_-K-cemu-je-to-dobre.aspx>
- [16] *Data Developer Center* [online]. 7.5.2008 [cit. 2011]. ADO.NET Data Providers. Dostupné z WWW: <<http://msdn.microsoft.com/en-us/data/dd363565>>
- [17] BARNES, Jeff. *Blogs.msdn.com/jbarnes* [online]. 2009 [cit. 2011-05-04]. Dostupné z WWW: <<http://blogs.msdn.com/b/jbarnes/archive/2009/05/20/net-ria-services-building-data-driven-business-applications-with-microsoft-silverlight3-and-microsoft-asp-net.aspx>>
- [18] *Riastats* [online]. 2011 [cit. 2011-05-28]. Ritch Internet Application Statistics. Dostupné z WWW: <<http://riastats.com>>
- [19] *StatOWL* [online]. 2011 [cit. 2011-05-28]. Web Browser Plugin Market Share. Dostupné z WWW: <http://www.statowl.com/plugin_overview.php>
- [20] *C-sharpcorner.com* [online]. 2011 [cit. 2011-05-28]. Introduction to Microsoft Silverlight. Dostupné z WWW: <<http://www.c-sharpcorner.com/UploadFile/GemingLeader/7216/>>
- [21] The Evolution Of LINQ And Its Impact On The Design Of C#. *MSDN Magazine* [online]. 2007, June, [cit. 2011-05-28]. Dostupný z WWW: <<http://msdn.microsoft.com/en-us/magazine/cc163400.aspx>>
- [22] Mono [online]. 2011 [cit. 2011-05-28]. Moonlight. Dostupné z WWW: <<http://www.mono-project.com/Moonlight>>
- [23] LOS, Radek. *Implementace podpory rozvrhování výuky v prostředí Microsoft Excel*. Zlín, 2011. 80 s. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

WCF RIA	Windows Communication Foundation Rich Internet Application.
LINQ	Language-Integrated Query.
SQL	Structured Query Language.
XML	Extensible Markup Language.
ADO.NET	ActiveX Data Object for .NET.
SSDL	Store schema definition Language.
CSDL	Conceptual sheme definition language.
MSL	Mapping specification language.
EF	Entity Framework

SEZNAM OBRÁZKŮ

Obr. 1. Architektura LINQ [21].....	11
Obr. 2. Entity framework architektura [2]	15
Obr. 3. Architektura platformy Silverlight [20].....	18
Obr. 4. Schéma standardní webové aplikace [17]	21
Obr. 5. Schéma RIA aplikace [17].....	22
Obr. 6. Současný stav řešení.....	25
Obr. 7. Relační schéma databáze [16]	31
Obr. 8. Přehled tabulek	32
Obr. 9. Základní struktura aplikace	41
Obr. 10. Struktura klientské části projektu	42
Obr. 11. Struktura serverové části projektu	43
Obr. 12 Schéma autorizace a autentizace vytvořené aplikace	45
Obr. 13. Práva uživatele Host	46
Obr. 14. Práva uživatele tajemník.....	46
Obr. 15. Práva uživatele administrátor	47
Obr. 16. Přihlašovací formulář	48
Obr. 17. Domovská stránka aplikace	49
Obr. 18. Editace oborů.....	50
Obr. 19. Editace práv	51
Obr. 20. Editace předmětů	52
Obr. 21. Editace vyučujících.....	53
Obr. 22. Editace výuky	54
Obr. 23. Uživatelský prvek výuka	54
Obr. 24. Uživatelský prvek pro volbu vyučujícího.....	55
Obr. 25. Formulář sloučené výuky	56
Obr. 26. Zobrazení vyučujících	57
Obr. 27. Zobrazení výuky	58
Obr. 28. Instalace „Out of browser“ aplikace	59
Obr. 29. Okno aplikace v prohlížeči a okno „Out of browser“ aplikace	59
Obr. 30. Zjednodušení procesu	60

SEZNAM TABULEK

Tab. 1. Dynamické parametry vyučujícího.....	25
Tab. 2. Statické parametry vyučujícího	26
Tab. 3. Kolekce vyučujícího	26
Tab. 4. Statické parametry předmětu	27
Tab. 5. Kolekce předmětu.....	27
Tab. 6. Statické parametry studijního oboru.....	28
Tab. 7. Kolekce oboru.....	28
Tab. 8. Popis tabulky „kecatko“	32
Tab. 9. Popis tabulky „koeficienty“	33
Tab. 10. Popis tabulky „kvalifikace“	33
Tab. 11. Popis tabulky „období“.....	34
Tab. 12. Popis tabulky „obor“	34
Tab. 13. Popis tabulky „permis“	35
Tab. 14. Popis tabulky „semestr“.....	35
Tab. 15. Popis tabulky „predmet“.....	36
Tab. 16. Popis tabulky „semestr“.....	37
Tab. 17. Popis tabulky „semestr_predmet“	37
Tab. 18. Popis tabulky „sloucena_vyuka“	38
Tab. 19. Popis tabulky „ustav“	38
Tab. 20. Popis tabulky „vyucujici_obdobi“	38
Tab. 21. Popis tabulky „vyucujici“	39
Tab. 22. Popis tabulky „vyuka“	40
Tab. 23. Popis objektů klientské části aplikace	43
Tab. 24. Popis objektů serverové části aplikace	44

SEZNAM PŘÍLOH

P I: Zdrojové kódy aplikace na přiloženém CD-ROM

