

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

Diplomová práce

Multiplatformní aplikace na zobrazování geografických dat s využitím Windows Azure

Plzeň, 2015

Martin Matas

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne

Martin Matas

Abstract

This thesis discusses three phenomenons in IT. The first phenomenon is cross-platform development – a way of creating a single application for multiple platforms simultaneously. The second phenomenon is cloud computing, and a representative of cloud computing – Microsoft Azure. The third phenomenon is displaying of geographic data on a map, which is a rapidly expanding area of IT, thanks to the rise of mobile devices. We will also look into KML – a file format for displaying geographic data on a map. The result of this thesis is the design, the creation and the testing of an application that displays geographic data on a map. There are two parts of the application – the server part and the client application for Windows Phone 8.1. Both parts of the application use Web services to communicate with each other.

Abstrakt

Tato práce pojednává o třech fenoménech v IT. Prvním fenoménem je multiplatformní programování, tedy možnost vytvořit jednu aplikaci pro více platforem současně. Druhým fenoménem je cloud computing a jeden zástupce cloud computingu – Microsoft Azure. Třetím fenoménem je zobrazování geografických dat v mapě, což je s nástupem mobilních zařízení velmi rychle se rozšiřující oblast IT. Podíváme se také na KML – formát souborů na zobrazování geografických dat v mapě. Výstupem této práce je návrh, vytvoření a otestování aplikace, která zobrazuje geografická data v mapě. Tato aplikace se dělí na dvě části, na serverovou část aplikace a na klientskou aplikaci na Windows Phone 8.1. Obě části aplikace spolu komunikují prostřednictvím webových služeb.

Obsah

1. Úvod.....	7
2. Multiplatformnost a webové služby.....	8
2.1 Multiplatformní programování	8
2.1.1 Příklady nástrojů.....	8
2.2 Multiplatformnost napříč typy zařízení.....	9
2.3 Multiplatformnost mezi Windows a Windows Phone	10
2.4 Webové služby.....	11
3. Microsoft Azure.....	13
3.1 Úvod do cloud computing.....	13
3.2 Microsoft Azure	14
3.2.1 Management Portal.....	14
3.3 Azure a vícevrstvé aplikace	18
3.3.1 Pojem vícevrstvá aplikace	18
3.3.2 Vícevrstvé aplikace na Microsoft Azure	18
3.3.3 Třívrstvá aplikace na Microsoft Azure.....	19
3.4 Podpora multiplatformnosti na Microsoft Azure	19
4. KML a zobrazení geografických dat.....	21
4.1 Standard KML	21
4.2 Soubor KMZ	22
4.3 Ustupování podpory KML.....	22
4.3.1 Seznam skutečností snižování podpory KML.....	22
4.4 Zobrazení KML na jiných platformách	23
4.4.1 Srovnání webových pluginů podporujících KML.....	23
4.4.2 Zobrazení na Androidu.....	23
4.5 Zobrazení KML na Windows Phone	23
4.5.1 Zobrazení v aplikaci	24
4.5.2 Zobrazení ve webovém prohlížeči.....	24
4.5.3 Windows Runtime API (verze Windows Phone 8.1).....	25
4.5.4 Bing Maps API.....	25
4.5.5 Závěr.....	25
4.6 Zobrazení geografických dat na Windows Phone.....	25
4.7 Možnosti uživatelského generování KML.....	26
4.7.1 Google Maps – Moje uložená místa.....	27
4.7.2 Tool for Google Maps	27
4.7.3 Free Map Tools.....	27
4.7.4 Earth Point	27
4.7.5 KML Tools	27
4.7.6 Mapmash	27
4.7.7 Spreadsheet Mapper 3	27
4.7.8 Google Earth.....	27
4.7.9 Srovnání aplikací	28
4.8 Knihovny pro generování KML.....	28
4.8.1 libkml.....	29
4.8.2 pyKML	30

4.8.3	SharpKml.....	30
4.8.4	JAK (Java API for KML)	30
5.	Analýza celého systému.....	31
5.1	Specifikace požadavků.....	31
5.2	Výběr technologií.....	31
5.2.1	Služba Microsoft Azure pro serverovou část aplikace	32
5.2.2	Výběr webových služeb	32
5.2.3	Přenos geografických dat	32
5.2.4	Způsob implementace mobilní aplikace.....	33
5.2.5	Technologie pro vykreslení geografických dat do mapy	34
5.2.6	Publikování na Windows Store	34
6.	Návrh serverové části aplikace	35
6.1	Automatické spouštění jobu.....	35
6.2	Načtení konfiguračního souboru	36
6.3	Parsování.....	36
6.3.1	Parsování hlavičky.....	36
6.3.2	Parsování jednotlivých řádek	37
6.4	Uložení změn do databáze	37
7.	Návrh mobilní aplikace.....	39
7.1	Asynchronní požadavek.....	39
7.2	Entity vrácené serverem.....	40
7.3	Navigace	40
7.4	První stránka – Vybrání města.....	40
7.5	Druhá stránka – Počasí ve městě	41
7.5.1	Funkce zatržítka ‚Vždy zobrazovat toto město‘	41
7.6	Třetí stránka – mapa.....	42
7.6.1	Zobrazení počasí pro chybějící data	43
7.6.2	Tabulka typů počasí.....	43
8.	Implementace serverové části aplikace.....	44
8.1	Obecný model	44
8.1.1	ProcessWeatherJob.....	44
8.1.2	TownConfig.....	44
8.1.3	Town.....	45
8.1.4	WeatherItem	45
8.1.5	TownController	45
8.1.6	WeatherItemController.....	45
8.1.7	pocasiczContext.....	45
8.2	Datové struktury.....	45
8.3	Použité technologie	45
8.4	Poznámky k realizaci	46
8.4.1	Automatická konverze času.....	46
8.4.2	Nastavení nového jobu na Microsoft Azure	47
9.	Implementace mobilní aplikace	48
9.1	Obecný model	48
9.1.1	App	48
9.1.2	MainPage.....	49

9.1.3	TownWeather	49
9.1.4	Map.....	49
9.1.5	Town.....	49
9.1.6	WeatherItem	49
9.2	Datové struktury.....	49
9.3	Použité technologie	49
9.4	Poznámky k realizaci	50
9.4.1	Práce s komponentou MapControl	50
9.4.2	Nasazení na Windows Store.....	50
10.	Ověření funkčnosti	51
10.1	Testování	51
10.2	Měření důležitých parametrů.....	52
10.2.1	Klientská část	52
10.2.2	Serverová část	52
10.3	Srovnání s dalšími aplikacemi, které umí zobrazit počasí v mapě.....	53
10.3.1	AccuWeather.....	53
10.3.2	Meteoservis Lite.....	53
10.3.3	Počasí (MSN Počasí).....	53
10.3.4	Weather (The Weather Channel)	53
10.3.5	Srovnání	53
10.4	Oblíbenost aplikace na Windows Store.....	53
10.5	Hodnocení multiplatformnosti.....	54
11.	Technická omezení a systémové požadavky	56
12.	Závěr.....	57
	Přehled zkratk a pojmů.....	58
	Literatura.....	60
	Příloha A – Uživatelská příručka	63
	Postup spuštění aplikace na lokálním stroji	63
	Instalace aplikace z Windows Store.....	66
	Uživatelská příručka pro aplikaci PočasíCZ.....	67
	Příloha B – Zobrazení KML ve webovém prohlížeči	70
	Příloha C - Konfigurační soubor načítaný jobem	71

1. Úvod

Tato práce se zabývá hned několika fenomény IT.

Multiplatformní aplikace je rychle se vyvíjející oblast IT. S příchodem nových druhů mobilních platform přišla také potřeba mít možnost spustit jednu aplikaci na co největším počtu dostupných platform. Taková aplikace se nazývá multiplatformní.

Cloud computing poskytuje hostování služeb, aplikací, nebo i celých sítí na cizí infrastruktuře, přičemž zajišťuje bezpečnost, stabilitu a škálovatelnost. Tato práce se zabývá jedním zástupcem cloud computingu – službou *Microsoft Azure* a její schopností hostovat vícevrstvé aplikace.

Zobrazování geografických dat v mapě na mobilních aplikacích je velice rychle se vyvíjející oblast IT. Většina dnešních mobilních zařízení má modul GPS, díky kterému stále více aplikací získává polohu zařízení, na kterém běží. Vzniká tak stále více příležitostí práce s mapou. Bude prozkoumán formát KML, což je formát souboru, který obsahuje geografická data.

Cílů práce je hned několik. Nejprve je nutné prozkoumat možnosti generování KML souborů na platformě Windows Phone 8.1 a prozkoumat možnosti Microsoft Azure pro tvorbu vícevrstevných aplikací. Poté bude vybrána jedna služba z nabídky Microsoft Azure, ve které bude naimplementována serverová část aplikace. Dalším cílem je vybrat způsob zobrazování geografických dat v mapě na platformě Windows Phone 8.1.

V praktické části bude poté vytvořena aplikace – jakési technologické demo, které zaintegruje tyto technologie: Službu Microsoft Azure pro implementaci serverové části aplikace, webové služby pro komunikaci mezi oběma částmi aplikace a klientskou aplikaci na Windows Phone 8.1, která zobrazí geografická data v mapě. Zároveň musí být aplikace smysluplná a také multiplatformní do té míry, že bude možné snadno přidat další klientské platformy bez nutnosti měnit serverovou část.

2. Multiplatformnost a webové služby

Počítačová platforma je definována jako pracovní prostředí, jak po stránce hardware, tak i software. Toto prostředí umožňuje bezproblémovou činnost programů. Z hardwarové stránky je platformou zejména stavba počítače, použité komponenty (procesor, paměti, připojené periférie). Ze softwarové stránky platforma určuje použitý operační systém, knihovny, ale i použité programovací jazyky či kompletní framework (vývojová a běhová platforma) [4].

Multiplatformnost je tedy schopnost aplikace běžet na více platformách [5]. Ta může být dosažena více vývojovými přístupy.

2.1 Multiplatformní programování

Existují různé způsoby přístupu k problému, jakým je vývoj multiplatformní aplikace. Jeden takový přístup je jednoduše vytvořit více verzí stejného programu v různých repozitářích, jinými slovy, verze pro Windows může mít jednu sadu souborů zdrojového kódu a verze pro Macintosh může mít jinou, zatímco verze pro systém na bázi UNIXu může mít ještě jinou. Ačkoliv toto je přímý přístup k problému, má potenciál být náročnější na cenu vývoje, čas vývoje, nebo obojí, obzvláště pro velké firmy. Je totiž potřeba vytvořit více než dva různé programy, které mají schopnost chovat se podobně. Také je možné, že tyto způsoby vývoje multiplatformní aplikace způsobí více problémů s hlášením chyb a jejich opravou, protože dva různé repozitáře budou mít různé programátory, a tím pádem různé chyby v každé verzi. Čím menší je vývojový tým, tím rychlejší bývají opravy chyb [5].

Jiný přístup, který bývá používán, je spolehnout se na již vytvořený software, který zakrývá rozdíly mezi platformami – zvaný abstrakce platformy – takže samotný program neví, na jaké platformě běží. Dalo by se říci, že takové programy jsou platformně nezávislé. Programy, které běží na Java Virtual Machine jsou vyvíjeny právě v tomto duchu. Programování ve WLanguage (používané ve WinDev, WebDev a WinDev Mobile) se používá k cílení na široký okruh platform: Windows, Linux, .Net, Java, Mac, Internet, Intranet, Tablet, Smartphone, Windows 8 RT, Windows Mobile, Windows Phone, Android, iOS iPhone/iPad [5].

Některé aplikace mixují různé metody multiplatformního programování k vytvoření výsledné aplikace. Příklad tohoto je webový prohlížeč Firefox, který používá abstrakci k vytvoření některých nízkourovňových komponent, oddělené podsložky repozitáře na implementaci funkcí závislých na platformě (např. GUI) a implementaci více než jednoho skriptovacího jazyka, aby byla usnadněna přenositelnost [5].

2.1.1 Příklady nástrojů

Níže jsou nastíněny příklady některých nástrojů, které umožňují nebo usnadňují vytváření multiplatformních aplikací [5, 6, 7, 8].

- Sencha
 - Nástroj k tvorbě aplikací v HTML5, poté je ještě nutné použít jiný nástroj k přeložení z HTML5 do Androidu nebo iOS
- PhoneGap

- Nástroj k přeložení kódu, který je v HTML5, CSS a JavaScriptu do kódu pro platformy: Android, iOS, BlackBerry a Windows
- Cocos2d
 - Nástroj využívaný především pro tvorbu 2D her, dává programátorovi možnost vývoje v kterémkoli z 5 podporovaných jazyků
- Ionic
 - Postaven na frameworku AngularJS
 - Mladý framework pro psaní GUI v HTML5, CSS a Javascriptu rychle získávající na popularitě
 - Vhodný především pro GUI, v případě složitějších aplikací není vhodný pro napsání celé aplikace
- Unity
 - Engine pro 3D hry, který lze spustit na mnoha platformách včetně Android, Windows, Windows Phone, Mac, či některých herních konzolích
- Corona
 - Engine pro 2D hry, který používá skriptovací jazyk Lua a podporuje iOS a Android
- Qt
 - Díky této službě je možné exportovat kód napsaný v C++ na různé platformy
 - Je možné vidět, jak aplikace funguje na různých zařízeních i bez nutnosti vlastnit tato zařízení
- Xamarin
 - Vývojová platforma, která umožňuje testování a monitorování výkonnosti a virtuální testy na více než tisíce zařízení
 - Je možné vyvíjet v jazycích Ruby nebo C#
 - Existuje doplněk pro Visual Studio, pro exportování do iOS je však stále nutný počítač s nainstalovaným Mac OS
- Alpha Anywhere
 - Nástroj s velkou podporou programovacích jazyků a množstvím tutoriálů
 - Vysoká cena

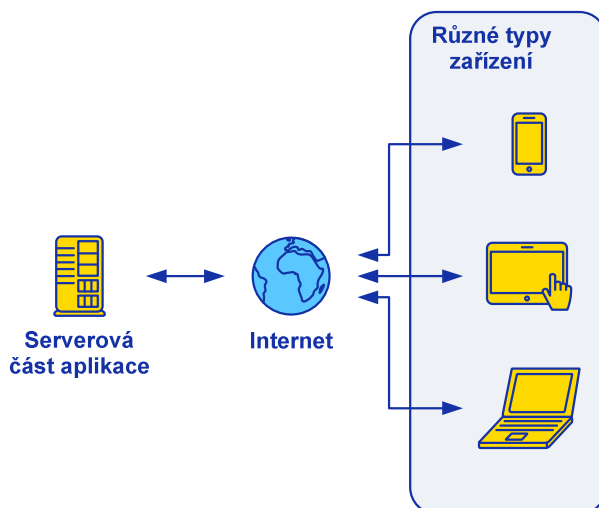
2.2 Multiplatformnost napříč typy zařízení

S příchodem nových zařízení a nových typů zařízení vznikla potřeba tvořit multiplatformní aplikace, které budou fungovat napříč více typy zařízení, tedy napříč:

- Počítači
- Tablety

- Mobilními telefony
- Dalšími zařízeními (např. různé typy herních konzolí)

V případě aplikací, které mají serverovou část (viz Obr. 1), je tendence udělat tuto serverovou část co nejvíce univerzální, tzn. udělat jakési API, ke kterému se připojují jednotlivé aplikace na různých platformách. Tyto aplikace však již mohou být velice odlišné, co do designu, tak i do samotné funkčnosti, která může být omezena možnostmi konkrétní platformy.



Obrázek 1 – Multiplatformní aplikace se serverovou částí

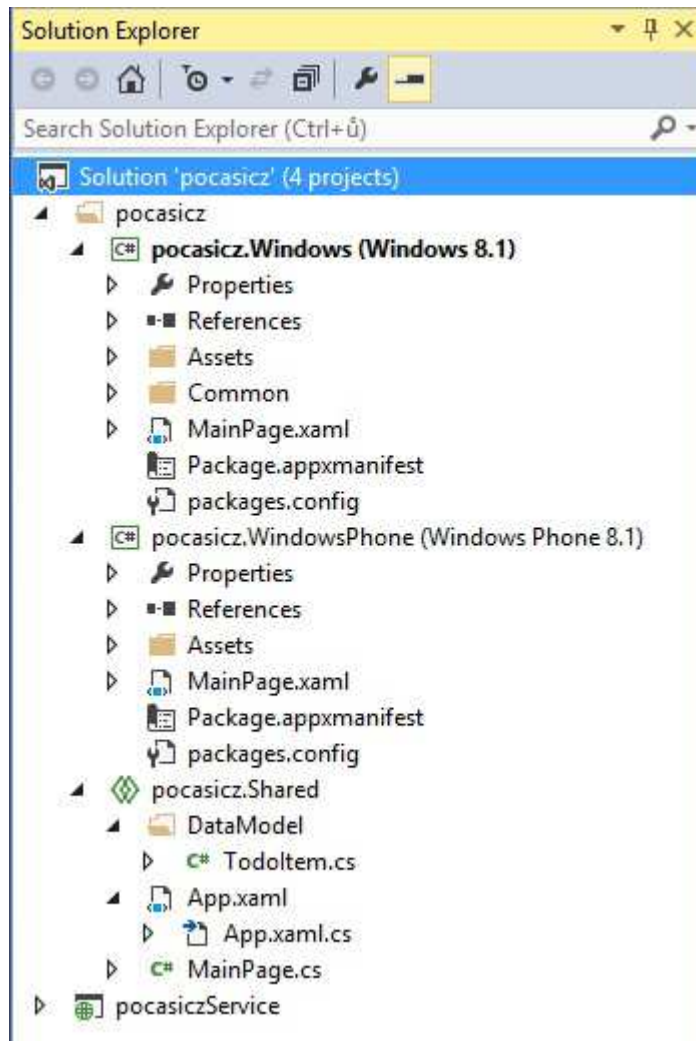
2.3 Multiplatformnost mezi Windows a Windows Phone

Přesto, že Windows a Windows Phone jsou různé operační systémy, Microsoft se snaží o jejich co největší sblížení. To je vidět i na možnostech multiplatformnosti napříč těmito dvěma systémy, resp. platformami.

To, co je mezi jinými platformami možné pouze za použití specializovaných nástrojů (viz kapitulu 2.1.1), je mezi Windows a Windows Phone možné standardně, za použití Visual Studia bez žádných doplňků.

Při vývoji multiplatformní aplikace pro Windows a Windows Phone je ve Visual Studiu možné rozdělení aplikace do těchto tří projektů (viz Obr. 2):

- Projekt specifický pro Windows
- Projekt specifický pro Windows Phone
- Sdílený projekt, používaný na obou platformách



Obrázek 2 - Projekt sdílený pro Windows a Windows Phone

Míra sdílení mezi Windows a Windows Phone může být až na takové úrovni, že je možné zvlášť vytvořit pouze GUI, vše ostatní můžeme vytvořit jen jednou – společný kód pro Windows i Windows Phone. Na Obr. 2 je toto vidět u třídy MainPage:

- MainPage.xaml
 - Obsahuje pouze GUI
 - Definována v projektech pro Windows a Windows Phone zvlášť
- MainPage.cs
 - Obsahuje code-behind společný pro obě výše zmíněné třídy definující GUI
 - Pouze v projektu pocasicz.Shared (sdílený projekt)

2.4 Webové služby

Webová služba je způsob síťové komunikace mezi dvěma elektronickými zařízeními. Je to softwarová funkce poskytnutá poskytnutá na určité webové adrese, která je stále zapnutá a přístupná [9].

U mobilních aplikací, které mají serverovou část (popsaných v kapitole 2.2), se ke vzájemné komunikaci používají právě webové služby. Serverová část aplikace je poskytovatelem webové služby, která je vždy přístupná, nikdy se nevypíná a je schopna zpracovat velké množství požadavků. Klientská (tzn. mobilní) část aplikace je žadatelem webové služby a typicky se připojuje k poskytovateli v momentě, kdy si uživatel vyžádá konkrétní data (ale může se připojovat i periodicky bez uživatelské interakce).

Webové služby se dělí do dvou skupin:

- Libovolné webové služby
- Služby kompatibilní s REST [9, 10, 11, 12]

Libovolné webové služby (přeloženo z angl. „Arbitrary Web services“) mohou provozovat libovolný seznam operací. Jejich nejvýznamnější zástupcem je protokol SOAP, který je proveden v syntaxi jazyka XML. Obvykle je doprovázen protokolem WSDL, pomocí kterého se definuje přípustný formát požadavku i odpovědi webové služby. WSDL je také proveden v syntaxi jazyka XML.

Služby kompatibilní s REST (přeloženo z angl. REST-compliant nebo RESTful) je další typ webových služeb. Zkratka REST je z angl. Representational State Transfer. Služby kompatibilní s REST obvykle využívají metod protokolu HTTP (GET, POST, PUT, DELETE, ...) a tím předávají data službě. Jsou vhodnější pro jednodušší služby, jelikož k vytvoření požadavku není potřeba vytvářet XML, jako je tomu u protokolu SOAP. Existují dokonce webové služby, které poskytují data ve formátech jako například CSS (Comma Separated Values), JSON (JavaScript Object Notation), nebo RSS (Really Simple Syndication).

3. Microsoft Azure

Tato kapitola nejprve popisuje cloud computing obecně a poté jednu platformu cloud computingu – Microsoft Azure.

3.1 Úvod do cloud computing

Cloud computing označuje souhrnně technologie a postupy používané v datových centrech a firmách pro zajištění snadné škálovatelnosti aplikací dodávaných přes internet. Zároveň je obecným názvem pro služby distribuované pomocí internetu [3].

Technologie umožňující cloud computing: [3]

- Virtualizace
- Servisně orientovaná architektura (SOA)
- Webové služby

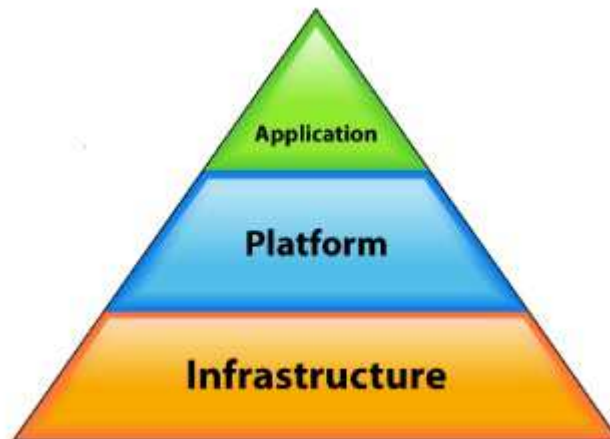
Výhody cloud computingu:

- Škálovatelnost
- Rychlost nasazení
- Bezpečnost dat
- Garantovaná dostupnost (obvykle 99,9%, Microsoft Azure se chlubí dostupností 99,95%)

Cloud computing rozdělujeme podle úrovně abstrakce do tří základních kategorií (viz Obr. 3):

- Software as a Service (SaaS)
 - Software jako služba. Představuje pronájem celé aplikace zákazníkem. Aplikace běží na softwarové i hardwarové infrastruktuře poskytovatele aplikace. Obvykle se jedná o webovou aplikaci nebo webovou službu. Příkladem podobné aplikace jsou emailové služby, např. Gmail nebo HotMail [3].
- Platform as a Service (PaaS)
 - Platforma jako služba. V tomto případě se jedná o outsourcing služeb potřebných pro provoz aplikace, ale ne samotné aplikace. Zákazník má aplikaci, ale nemá nebo nechce provozovat infrastrukturu nutnou pro její provoz, proto využije služeb cloudu a aplikaci nasadí a provozuje zde. Cloud poskytuje prostředky nutné pro běh aplikace a účtují se spotřebované prostředky za určité časové období [25].
- Infrastructure as a Service (IaaS)
 - Infrastruktura jako služba znamená používání vlastní virtuální infrastruktury na fyzické infrastruktuře poskytovatele. Použití IaaS je mnohem flexibilnější řešení, protože IaaS nabízí větší možnosti konfigurace virtuálních strojů [25].

- IaaS umožňuje kontrolovat virtuální stroje již od cloudové vrstvy operačního systému. Tím pádem má zákazník mnohem větší možnosti v konfiguraci a nastavení serverů. Je možné zvolit libovolný operační systém nebo nahrát již existující obraz OS [25].
- IaaS umožňuje provozovat již stávající aplikace bez nutnosti jejich úprav na virtuálních serverech v cloudu [3].

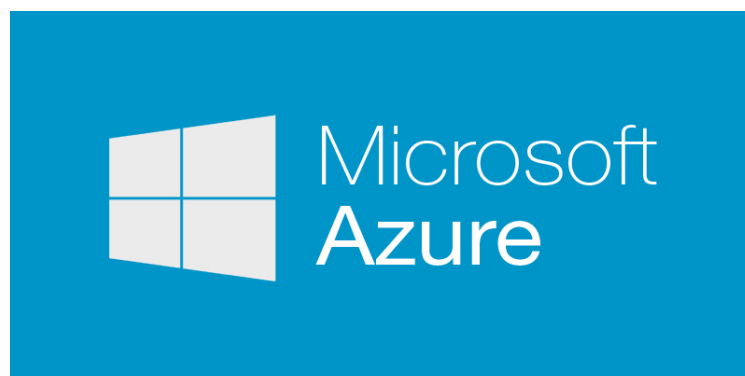


Obrázek 3 - Úrovně abstrakce aplikací [3]

3.2 Microsoft Azure

Microsoft Azure je cloud computing platforma společnosti Microsoft pro vývoj, správu a provoz vysoce škálovatelných aplikací. Základem platformy jsou obrovská datová centra rozmístěná po celém světě a vzájemně propojená vysokorychlostním spojením [3].

Microsoft Azure byl dříve znám pod pojmem Windows Azure, dne 25.3.2014 byl přejmenován na Microsoft Azure. Logo Microsoft Azure je ukázáno na Obr. 4 [22].



Obrázek 4 - Logo Microsoft Azure

3.2.1 Management Portal

Hlavním bodem služeb Microsoft Azure je Management Portal - administrační rozhraní pro všechny služby platformy Microsoft Azure. Na Obr. 5 vidíme Management Portal, na kterém je zobrazen seznam všech využívaných služeb.

The screenshot shows the Microsoft Azure Management Portal interface. At the top, there's a header with 'Microsoft Azure', a 'CREDIT STATUS' indicator, and a user profile for 'matasmartin@gmail.com'. On the left, a navigation pane lists various service categories: WEB APPS (2), VIRTUAL MACHINES (1), MOBILE SERVICES (1), CLOUD SERVICES (1), SQL DATABASES (1), STORAGE (3), HDINSIGHT (0), MEDIA SERVICES (0), SERVICE BUS (2), VISUAL STUDIO ONLINE (1), CACHE (0), BIZTALK SERVICES (0), and RECOVERY SERVICES. The main area displays a table titled 'all items' with columns for NAME, TYPE, STATUS, SUBSCRIPTION, and LOCATION. The table lists 15 items, including Storage Accounts, Cloud services, Web apps, Mobile Services, Virtual machines, Service Bus Namespaces, and SQL Databases, each with a status indicator (Online, Running, Ready, Active) and a location (West Europe, North Europe, North Central US).

NAME	TYPE	STATUS	SUBSCRIPTION	LOCATION
portalvhds4db20kn5x28rp	Storage Account	Online	Windows Azure MSDN ...	West Europe
portalvhdsq5124xbvsrmlc	Storage Account	Online	Windows Azure MSDN ...	North Europe
tomtomb	Storage Account	Online	Windows Azure MSDN ...	North Europe
tomtomb1	Cloud service	Running	Windows Azure MSDN ...	North Europe
pocasicz	Web app	Running	Windows Azure MSDN ...	West Europe
testkml	Web app	Running	Windows Azure MSDN ...	West Europe
pocasicz	Mobile Service	Ready	Windows Azure MSDN ...	North Europe
tomtomb	Virtual machine	Running	Windows Azure MSDN ...	North Europe
pocasiczMsg-ns	Service Bus Namesp...	Active	Windows Azure MSDN ...	North Europe
pocasiczHub-ns	Service Bus Namesp...	Active	Windows Azure MSDN ...	North Europe
pocasicz_db	SQL Database	Online	Windows Azure MSDN ...	North Europe
matysek	Visual Studio Online	Active	Windows Azure MSDN ...	North Central US

Obrázek 5 - Management Portal

Seznam služeb Microsoft Azure (služby pro tvorbu vícevrstevných aplikací jsou popsány podrobněji než ostatní služby):

- **Web Apps**
 - Hostování webových aplikací. Umožňuje vývojářům vytvářet stránky pomocí technologií ASP.NET, PHP, Node.js nebo Python, nebo nasadit některý ze známých redakčních systémů – WordPress, Umbraco, Joomla, nebo Drupal. Aplikace mohou být nasazeny za pomoci technologií jako FTP, Git, Mercurial nebo Team Foundation Server [22, 23].
 - Tato služba byla ohlášena v předběžné formě v červnu roku 2012 na události *Meet Microsoft Azure* [22].
 - Tato služba je zástupcem *Platform as a Service (PaaS)*.
- **Virtual Machines**
 - Umožňuje vývojářům migrovat aplikace a infrastrukturu beze změny existujícího kódu. Na Virtual Machines může běžet Windows Server 2008, Windows Server 2012 a několik distribucí Linuxu [22, 23].
 - Tato služba byla ohlášena v předběžné formě v červnu roku 2012 na události *Meet Microsoft Azure* [22].
 - Zákazníci si mohou vytvořit virtuální stroj, nad kterým mají naprostou kontrolu, ale přitom běží na datových centrech Microsoftu.
 - Tato služba je zástupcem *Infrastructure as a Service (IaaS)*.

- **Mobile Services**
 - Webové služby na bázi REST, ke kterým se lze připojit z libovolného mobilního zařízení. Mobile Services podporují platformy Windows, Android a iOS [23].
 - Podporují tzv. *push notifikace* - způsob jak oznámit mobilnímu zařízení (i velkému množství zařízení) že na serveru nastala určitá událost. Princip push notifikací tkví v tom, že je neustále nechané otevřené spojení mezi serverem a klientem.
 - Součástí služby mohou být další služby Microsoft Azure, například SQL databáze nebo Scheduler na pravidelné spouštění jobů.
- **Cloud Services**
 - Hostování aplikací a různých API. Podporuje technologie Java, Node.js, PHP, Python, .NET a Ruby [23].
 - Automatická instalace patchů OS a aplikací.
 - Tato služba je zástupcem *Platform as a Service (PaaS)*.
- **SQL Databases**
 - Přednastavená SQL databáze založená na technologii Microsoft SQL Server, běžící v cloudu. Dokáže se integrovat s Active Directory, Microsoft System Center, nebo Hadoop [22].
- **Storage**
 - Úložiště pro objekty typu BLOB (velký binární objekt), soubory, tabulky, nebo dokonce fronty [22].
- **HDInsight**
 - Nasazení Apache Hadoop – frameworku určeného k práci s velkým množstvím výpočetních a datových uzlů. HDInsight také poskytuje analýzu nasazeného Apache Hadoop [22, 23].
- **Media Services**
 - Poskytování podpory pro multimédia, obzvlášť pro online přehrávače videí.
 - Služba zahrnuje podporu různých formátů kódování multimédií, streaming (datové proudy), ochranu obsahu a analytické nástroje. Podporuje například technologie a platformy Flash, iOS, Android, HTML5 a Xbox [22, 23].
 - Tato služba je zástupcem *Platform as a Service (PaaS)*.
- **Service Bus**
 - Zasílání zpráv napříč službami.
- **Visual Studio Online**
 - Repozitář pro zdrojový kód a další služby.
- **Cache**
 - Nerelační úložiště dat typu klíč-hodnota.

- BizTalk Services
 - Integrace technologií.
- Recovery Services
 - Automatická ochrana proti pádům virtuálních strojů. Funguje na principu replikace virtuálních strojů [23].
- CDN
 - Z angl. content delivery network. Poskytování infrastruktury serverů ke stahování velkých dat koncovými uživateli.
- Automation
 - Nastavení automatického nasazení.
- **Scheduler**
 - Vytváření jobů, kterým lze nastavit konkrétní plánování, v jakou dobu se mají spouštět.
 - Joby hlásí postup do přehledného logu přístupného z Management Portalu.
- API Management
 - Sada nástrojů k nasazeným API. Zahrnuje analytické nástroje, dokumentační nástroje a další [23].
- Machine Learning
 - Analytické nástroje.
- Stream Analytics
 - Analytické nástroje.
- Networks
 - Virtuální síť, VPN (tj. virtuální privátní síť), nastavení vlastních DNS serverů, vlastních IP adres.
- Traffic Manager
 - Vlastní nastavení load balancingu (česky vyvažování zátěže).
- RemoteApp
 - Hostování aplikací přístupných z mobilních zařízení. Aplikace běží na Microsoft Azure a výsledek je streamován do prohlížeče mobilního zařízení [23].
- Management Service
 - API pro přístup k většině funkcí, které jsou dostupné uživateli přes Management Portal.
- Active Directory
 - Nástroje pro sdílení identity uživatele napříč aplikacemi.
- Marketplace
 - Analytické nástroje.

- StorSimple
 - Hostování hybridního cloudu a automatický přenos dat mezi úložištěm dat v cloudu a lokálním úložištěm.

3.3 Azure a vícevrstvé aplikace

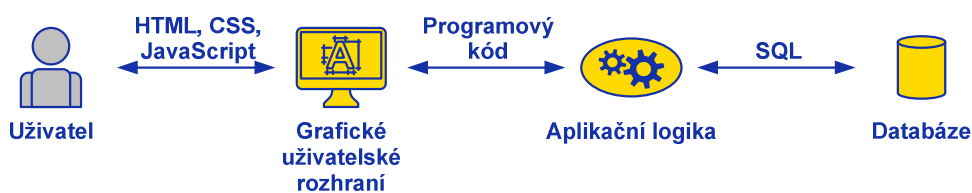
V této kapitole jsou zkoumány možnosti Microsoft Azure pro vytváření vícevrstevých aplikací.

3.3.1 Pojem vícevrstvá aplikace

Vícevrstvá aplikace je aplikace, jejíž funkčnost netvoří jeden celistvý program, ale více vzájemně spolupracujících vrstev, které běží zpravidla na různé výpočetní infrastruktuře. Příkladem může být internetový obchod, jehož klientská část běží v prohlížeči uživatele, aplikační logika na webovém či aplikačním serveru a data jsou uložena v databázovém serveru [24].

Nejznámějším případem vícevrstvé architektury je třívrstvá architektura (viz Obr. 6). Sestává z vrstev: [24]

- Prezentační vrstva
 - Zobrazuje informace pro uživatele, většinou formou grafického uživatelského rozhraní [24]
- Aplikační vrstva (angl. Business Logic)
 - Jádro aplikace, její logika a funkce, výpočty a zpracování dat [24]
- Datová vrstva
 - Nejčastěji databáze, která data uchovává, zpřístupňuje a zaručuje jejich konzistenci [24]



Obrázek 6 - Příklad třívrstvé aplikace

3.3.2 Vícevrstvé aplikace na Microsoft Azure

Podpora pro vytváření vícevrstevých aplikací je na Microsoft Azure výborná. Vývojář, který se rozhodne použít Microsoft Azure není nijak znevýhodněn oproti vývojáři, který použije jiné technologie, co se do otázky vícevrstvé architektury týče.

Nejslabším článkem nabízených služeb je rozhodně databáze, která má už ze své podstaty určité výhody, pokud je hostovaná lokálně. Viz tabulku 1.

Databáze v cloudu	Lokální databáze
Má pomalejší odezvu	Je rychlejší
Pouze Azure SQL založená na Microsoft SQL Server	Podpora libovolného SQL serveru
Analytické nástroje sledující vytížení a další statistiky	Pouhá instalace neposkytuje analytické nástroje
Automatická záloha, duplikování serverů a jejich synchronizace	Zálohování a ochranu proti pádu je potřeba obstarat ručně
Snadná škálovatelnost	Nesnadná škálovatelnost

Tabulka 1 - Srovnání databáze v cloudu a lokální databáze

Microsoft Azure je koncipován tak, aby bylo možno všechny vrstvy hostovat na cloudu. Ale i v případě, že se vývojář rozhodne jinak, je možné mít některé vrstvy aplikace lokálně. Nejvýznamnějším případem tohoto přístupu je možnost připojit aplikaci v cloudu k lokální databázi.

3.3.3 Třívrstvá aplikace na Microsoft Azure

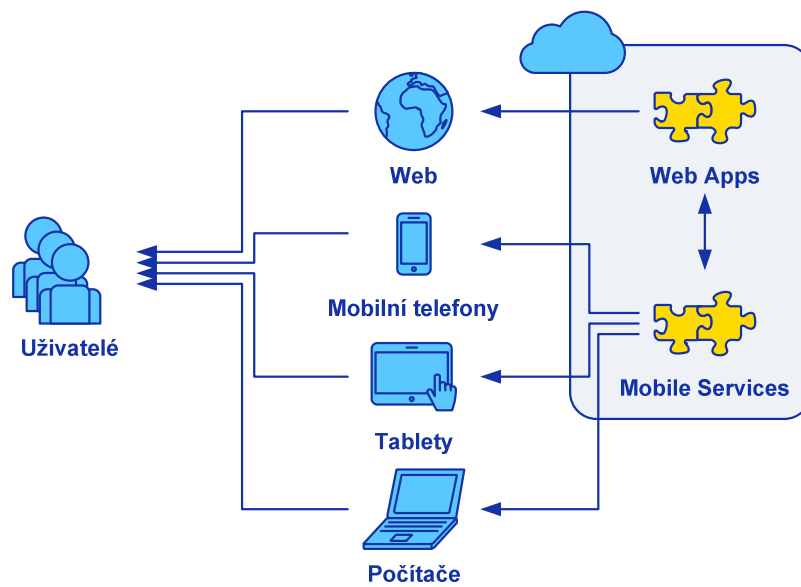
Microsoft Azure poskytuje celou škálu služeb, které se dají použít jednu nebo více vrstev třívrstvé aplikace. Níže jsou vyjmenovány jednotlivé služby, které se hodí pro implementaci jednotlivých vrstev. Pro popis služeb viz kapitolu 3.2.1.

- Prezentační vrstva
 - Webb Apps, Media Services
- Aplikační vrstva
 - Web Apps, Mobile Services, Cloud Services, Scheduler
- Datová vrstva
 - SQL Databases, Storage, Cache

3.4 Podpora multiplatformnosti na Microsoft Azure

Podpora multiplatformnosti na platformě Microsoft Azure je velmi dobrá, především díky možnosti pospojovat více služeb dohromady. Je tedy možné spojit například službu Web Apps se službou Mobile Services. Výsledné spojení umožňuje nasadit jednu aplikaci na web, ale zároveň poskytnout webové služby na bázi REST pro libovolné mobilní aplikace, které budou přistupovat ke stejné aplikaci. Viz Obr. 7.

Služba Media Services zase poskytuje širokou podporu různých platform, na kterých lze přehrávat multimédia.



Obrázek 7 - Multiplatformní aplikace na Microsoft Azure

4. KML a zobrazení geografických dat

4.1 Standard KML

KML je zkratka z angl. Keyhole Markup Language a je to formát XML souborů, který je zároveň mezinárodním standardem. Používá se k uložení geografických dat a jejich vizualizace v dvojdimenzionálních nebo trojdimenzionálních mapách. Důsledkem je, že tyto soubory mohou rychle růst na velikosti [14, 15, 19].

KML byl vyvinut firmou Keyhole, Inc. jako API pro její virtuální glóbus *Earth Viewer*. V roce 2004 firmu koupila společnost Google, Inc. a produkt přejmenovala na Google Earth. Ke dni 16.4.2008 se stal KML ve verzi 2.2 mezinárodním standardem Open Geospatial Consortium [14, 15].

Google Earth byla první aplikace, která dokázala zobrazit a graficky editovat KML soubory. Další projekty jako například Marble také začali podporovat formát KML [14]. Tento formát souborů býval hojně užíván ve webových aplikacích, ale od té doby je na úpadku, jelikož se dostaly do popředí jiné, více účinné formáty, jako například GeoJSON [19].

KML soubory specifikují sadu objektů, které je možno zobrazit v Bing Maps, Google Maps, Google Earth, popř. v jiných geoprostorových aplikacích podporujících KML [14]. Mezi tyto objekty patří například:

- Bod
- Polygon
- Křivka sestavená z úseček (tzv. LineString)
- Obrázek překrývající určitý prostor (tzv. PhotoOverlay)
- 3D modely
- Textové popisy

Příklad KML souboru je ukázán na Obr. 8.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<Placemark>
  <name>New York City</name>
  <description>New York City</description>
  <Point>
    <coordinates>-74.006393,40.714172,0</coordinates>
  </Point>
</Placemark>
</Document>
</kml>
```

Obrázek 8 - Příklad KML souboru

4.2 Soubor KMZ

KML soubory jsou často distribuovány v KMZ souborech, což jsou komprimované soubory KML s příponou .kmz. Obsahem KMZ souboru je jediný kořenový KML soubor s názvem doc.kml (v kořenovém adresáři) a případné odkazované KML soubory, obrázky, ikony, atd. v podadresářích [14].

4.3 Ustupování podpory KML

Jak již bylo zmíněno v kapitole 4.1, v r. 2008 se stal formát KML standardem a od té doby se rozšiřovala jeho podpora na různých platformách, včetně desktopu (především Google Earth), webu (viz kapitolu 4.4.1, která srovnává webové pluginy) i mobilních zařízení. Ještě v červnu r. 2013, kdy byla tato práce zadána, byl formát KML podporován natolik, že z níže popsaného seznamu neexistovala jediná položka. Za pouhé 2 roky je však patrné, že Microsoft i Google od podpory KML ustupují.

4.3.1 Seznam skutečností snižování podpory KML

- Webový plugin Google Earth – tedy nejlepší webový plugin co se zpracování KML souborů týče – je od 12.12.2014 považován za zastaralý a od 12.12.2015 končí jeho podpora úplně (viz kapitolu 4.4.1) [16]
- Zatímco starší verze webového pluginu Bing Maps – AJAX verze 6.3 – ještě KML soubor podporovala, AJAX verze 7 (aktuálně nejnovější verze pluginu) již soubor vůbec nedokáže načíst (viz kapitolu 4.4.1)
- Aplikace Google Maps pro Android přestala podporovat načtení KML souboru [18]

	Google Maps	Google Earth	Bing Maps (AJAX verze 7)	Bing Maps (AJAX verze 6.3)
Bod	Ano	Ano	Ne	Ano
Polygon	Ano	Ano	Ne	Ano
LineString	Ano	Ano	Ne	Ano
PhotoOverlay	Ano	Ano	Ne	Ne
3D zobrazení	Ne	Ano	Ne	Ne
Podpora zobrazení v čase	Ne	Ano	Ne	Ne

Tabulka 2 - Srovnání webových pluginů pro zobrazení KML

4.4 Zobrazení KML na jiných platformách

Tato kapitola se věnuje možnosti zobrazení KML souborů na jiných platformách než Windows Phone.

4.4.1 Srovnání webových pluginů podporujících KML

Některé aplikace nepodporují všechny objekty ani všechny funkce KML specifikace. Následující test tří nejrozšířenějších webových pluginů pro zobrazení geografických dat znázorňuje tuto skutečnost. Výsledek testu je znázorněn v tabulce 2.

Z testu vyšel jednoznačně nejlépe webový plugin Google Earth, který jako jediný splňuje specifikaci KML souborů, která umožňuje zobrazení 3D souborů. Nejhorší je na tom plugin Bing Maps.

4.4.2 Zobrazení na Androidu

Na OS Android lze KML soubor zobrazit snadněji než na OS Windows Phone. Soubor lze zobrazit pouhým otevřením v aplikaci. Nejvhodnější aplikace pro toto jsou: [17, 18]

- Locus Map Free
- Google Earth

Srovnání aplikací na Androidu pro zobrazení KML souboru je znázorněno v tabulce 3.

	Google Maps	Google Earth	Locus Map Free
Bod	Ne	Ano	Ano
Polygon	Ne	Ano	Ano
LineString	Ne	Ano	Ano
PhotoOverlay	Ne	Ne	Ano
3D zobrazení	Ne	Ano	Ne
Podpora zobrazení v čase	Ne	Ne	Ne

Tabulka 3 - Srovnání aplikací pro zobrazení KML na Androidu

Nutno podotknout, že dřívější verze aplikace Google Maps pro OS Android uměly načíst KML soubor. Od jisté doby ale aplikace tuto schopnost zcela ztratila [18].

4.5 Zobrazení KML na Windows Phone

Tato kapitola se věnuje možnostem zobrazení KML souboru na platformě Windows Phone.

4.5.1 Zobrazení v aplikaci

- Bing Maps a Here Maps
 - Výchozí aplikace pro zobrazení mapy, přímo od Microsoftu
 - Nepodporují zobrazení KML souboru
- Vectorial Map Lite
 - Freewareová aplikace od společnosti teq-IT, jejíž popis tvrdí, že dokáže načítat KML soubory. Byla však testována na několika validních KML souborech a žádný z nich nedokázala zobrazit.
- Run The Map
 - Freewareová aplikace od stejnojmenné společnosti. Tato aplikace je určena k zaznamenávání a zobrazování běžeckých tras. Dokáže načíst běžecskou trasu (tedy lomenou čáru sestávající ze série bodů, tzn. LineString) i z KML souboru. Žádné jiné objekty KML standardu ale načíst neumí, čímž se stává pro zobrazení geografických dat nedostatečnou. Viz tabulku 4.

	Run The Map
Bod	Ne
Polygon	Ne
LineString	Ano
PhotoOverlay	Ne
3D zobrazení	Ne
Podpora zobrazení v čase	Ne

Tabulka 4 - Srovnání aplikací pro zobrazení KML na Windows Phone

4.5.2 Zobrazení ve webovém prohlížeči

Webový plugin AJAX verze 6.3 podporuje načtení KML souboru, ale nepodporuje žádný mobilní prohlížeč. Naproti tomu novější verze pluginu – verze 7.0 – již nepodporuje načtení KML souboru, zato podporuje následující mobilní prohlížeče [20]: Internet Explorer Mobile Browser, Apple iPhone Browser, Google Android Browser, Research in Motion (RIM) BlackBerry Browser. Ve webovém prohlížeči tedy *nelze* na Windows Phone 8.1 zobrazit KML soubor.

4.5.3 Windows Runtime API (verze Windows Phone 8.1)

Windows Runtime API je základním API pro vývoj aplikací na platformě Windows Phone 8.1. Dává programátorovi rozsáhlou škálu možností práce s mapou, od jednoduchého zobrazení mapy po úpravu mapy a vkládání interaktivních grafických prvků. Avšak zobrazit KML soubor žádným jednoduchým způsobem neumí.

Jediný způsob zobrazení KML souboru by bylo vytvoření aplikace, která by naparsovala KML soubor a objekt po objektu by naparsovaná data vykreslovala do mapy.

4.5.4 Bing Maps API

Bing Maps API se dělí na několik dílčích API: [19]

- Windows Store Apps
 - V rámci knihy [19] byla vytvořena knihovna Microsoft Maps Spatial Toolbox, díky které je možné načíst KML soubory a zobrazit je v mapě. Tato knihovna je dostupná pouze pro Windows Store Apps pro desktop.
 - Cílové platformy: Desktop (Windows 8.1 a Windows 8)
- AJAX
 - Zobrazení ve webovém pluginu. Viz *Zobrazení ve webovém prohlížeči* výše
- REST Services
 - Neumožňují zobrazit KML soubor
 - Cílové platformy: Web, Mobilní zařízení, Desktop (Windows 8.1 nebo dřívější)
- Spatial Data Services
 - Neumožňují zobrazit KML soubor
 - Cílové platformy: Web, Mobilní zařízení, Desktop (Windows 8.1 nebo dřívější)

4.5.5 Závěr

Jediným způsobem zobrazení KML souboru na platformě Windows Phone 8.1 je vytvoření aplikace, která naparsoje KML soubor a objekt po objektu vykreslí naparsovaná data do mapy.

Žádný jednoduchý způsob zobrazení KML souboru na platformě Windows Phone 8 či 8.1 neexistuje.

4.6 Zobrazení geografických dat na Windows Phone

K zobrazení mapy na platformě Windows Phone 8.1 slouží komponenta `MapControl` z knihovny Windows Runtime API (verze Windows Phone 8.1) [21, 13]. Kód komponenty `MapControl` je zobrazen na Obr. 9.

```

<Maps:MapControl
  x:Name="MapControl1"
  MapServiceToken="<token>"
  Height="560"
  Margin="0,0,0,0" />

```

Obrázek 9 - Kód komponenty MapControl

Příklad práce s komponentou MapControl - vycentrování mapy okolo zadaných souřadnic – je znázorněn na Obr. 10 [21].

```

MapControl1.Center =
    new Geopoint(new BasicGeoposition() { Latitude = 47.604,
        Longitude = -122.329 });
MapControl1.ZoomLevel = 12;
MapControl1.LandmarksVisible = true;

```

Obrázek 10 - Příklad práce s komponentou MapControl

Do mapy lze snadno vložit libovolné geografické objekty:

- Body
- Vlastní ikony či obrázky
- Textové popisy
- Libovolné geometrické objekty: úsečky, elipsy, polygony, ...
- Obrázky překrývající určitý prostor

Veškeré výše zmíněné objekty lze přichytit na konkrétní GPS souřadnici v mapě.

V komponentě MapControl dokážeme zobrazit jakékoli objekty, které můžeme najít i ve standardu KML. Jen je potřeba je do komponenty vložit objekt po objektu a nastavit jejich vzhled.

Příklad vložení bodu do mapy je znázorněn na Obr. 11 [21].

```

MapIcon MapIcon1 = new MapIcon();
MapIcon1.Location = new Geopoint(new BasicGeoposition() {
    Latitude = 47.620,
    Longitude = -122.349 });
MapIcon1.NormalizedAnchorPoint = new Point(0.5, 1.0);
MapIcon1.Title = "Space Needle";
MapControl1.MapElements.Add(MapIcon1);

```

Obrázek 11 - Příklad vložení bodu do mapy

4.7 Možnosti uživatelského generování KML

V této kapitole jsou popsány aplikace, které umožňují uživateli zadat geografická data (jakýmkoliv způsobem) a z nich vygenerovat KML soubor.

4.7.1 Google Maps – Moje uložená místa

Online verze Google Maps obsahuje funkci Moje uložená místa. Tato funkce je poněkud skrytá, je snadné ji přehlédnout. Při její aktivaci ale umožňují Google Maps velice pohodlným a jednoduchým způsobem vytvářet KML soubor z bodů, lomených čar i polygonů s nastavitelným vzhledem.

Jedná se o webovou aplikaci Google Maps (maps.google.com), nikoliv o webový plugin Google Maps [26].

4.7.2 Tool for Google Maps

Velice jednoduchá webová stránka, která používá webový plugin Google Maps verze 3. Poskytuje veškeré potřebné možnosti ke generování KML souborů kromě možnosti textových popisů [27].

4.7.3 Free Map Tools

Velice jednoduchá webová stránka, umožňuje vygenerování KML souboru pouze s jediným bodem [28].

4.7.4 Earth Point

Z uživatelského hlediska velmi nepřívětivá webová aplikace. Převod na KML soubor se děje jedinečně z textového souboru či excelu, nelze ho najít v mapě, jako je tomu u ostatních aplikací. Podporované formáty jsou: xls,xlsx, xlsx, xlsb, txt a csv [29].

4.7.5 KML Tools

Z uživatelského hlediska také velmi nepřívětivá webová aplikace, protože opět nelze objekty kreslit do mapy. Navíc v každém módu podporuje jen jeden specifický formát. Obsahuje tři módy [30]:

1. Vygenerování jednoho bodu z GPS souřadnic
2. Vygenerování více bodů z CSV souboru
3. Vygenerování objektu LineString z DXF souboru

4.7.6 Mapmash

Příjemná webová aplikace postavená nad webovým pluginem Google Maps. Nevýhoda je nemožnost vytvoření více typů objektů do jednoho souboru [31].

4.7.7 Spreadsheet Mapper 3

Předvytvořený template Google Spreadsheets (obdoba excelu na webu). Již v sobě obsahuje skripty na tvorbu KML, jeho vystavení online, případně jeho průběžné obnovování. Podporuje pouze vytváření bodů [32].

4.7.8 Google Earth

Desktopová aplikace pro Windows, Mac, nebo Linux. Suverénně nejlepší ze všech aplikací.

Ovládání je intuitivní, mapu zobrazuje ve 3D, podporuje časový rozměr. Navíc umožňuje například do popisu objektů v mapě vkládat videa, která lze přehrát, nebo fotogalerie, které lze prohlížet [33].

4.7.9 Srovnání aplikací

Na tabulkách 5 a 6 jsou znázorněny jednotlivé funkčnosti, které aplikace mají nebo nemají.

	Google Maps	Tool for Google Maps	Free Map Tools	Earth Point
Bod	Ano	Ano	Jen jeden	Ano
Polygon	Ano	Ano	Ne	Ne
LineString	Ano	Ano	Ne	Ano
Překrytí mapy obrázkem	Ne	Ne	Ne	Ne
Textové popisy	Ano	Ne	Ano	Ano
Všechny druhy objektů najednou	Ano	Ano	N/A	Ano
Nastavitelný vzhled objektů	Ano	Ano	Ne	Ano
Podpora času	Ne	Ne	Ne	Ano

Tabulka 5 - Srovnání aplikací umožňujících vygenerovat KML soubor (1. část)

4.8 Knihovny pro generování KML

Podpora knihoven pro generování KML souboru je velmi dobrá. K dispozici je hned několik freewarových knihoven, které *podporují kompletní standard KML 2.2*, tedy dokáží vygenerovat veškeré objekty, které do tohoto standardu spadají. Stačí si vybrat programovací jazyk a příslušnou knihovnu.

	KML Tools	Mapmash	Spreasheet Mapper 3	Google Earth
Bod	Ano	Ano	Ano	Ano
Polygon	Ne	Ano	Ne	Ano
LineString	Ano	Ano	Ne	Ano
Překrytí mapy obrázkem	Ne	Ne	Ne	Ano
Textové popisy	Ano	Ano	Ano	Ano
Všechny druhy objektů najednou	Ne	Ne	N/A	Ano
Nastavitelný vzhled objektů	Ne	Ne	Ano	Ano
Podpora času	Ne	Ne	Ne	Ano

Tabulka 6 - Srovnání aplikací umožňujících vygenerovat KML soubor (2. část)

4.8.1 libkml

libkml je knihovna napsaná firmou Google v jazyce C++. Lze ji stáhnout z:

<https://code.google.com/p/libkml/>

Příklad použití knihovny je na Obr. 12.

```
KmlFactory* factory = KmlFactory::GetFactory();
CoordinatesPtr coordinates = factory->CreateCoordinates();
coordinates->add_latlng(37.42052549, -122.0816695);
PointPtr point = factory->CreatePoint();
point->set_coordinates(coordinates);
PlacemarkPtr placemark = factory->CreatePlacemark();
placemark->set_name("Cool Statue");
placemark->set_geometry(point);
KmlPtr kml = factory->CreateKml();
kml->set_feature(placemark);
std::string xml = kml->SerializePretty(kml);
```

Obrázek 12 - Příklad použití knihovny libkml

4.8.2 pyKML

pyKML je open-source knihovna napsaná v jazyce Python. Lze ji stáhnout z:

<https://code.google.com/p/pykml/>

Příklad použití knihovny je na Obr. 13.

```
pm1 = KML.Placemark(  
    KML.name("Hello World!"),  
    KML.Point(  
        KML.coordinates("-64.5253,18.4607")  
    )  
)  
from lxml import etree  
print etree.tostring(pm1, pretty_print=True)
```

Obrázek 13 - Příklad použití knihovny pyKML

4.8.3 SharpKml

SharpKml je knihovna napsaná v jazyce C# a dostupná ve formě DLL souboru. Je určena pro: .NET Framework 4 nebo vyšší, Silverlight 5, Windows Phone 8, Windows Store apps (Windows 8) nebo vyšší. Lze ji stáhnout z:

<https://sharpkml.codeplex.com/>

Příklad použití knihovny je na Obr. 14.

```
Point point = new Point();  
point.Coordinate = new Vector(-13.163959, -72.545992);  
Placemark placemark = new Placemark();  
placemark.Geometry = point;  
placemark.Name = "Machu Picchu";  
KmlFile kml = KmlFile.Create(placemark, false);  
using (FileStream stream = File.OpenWrite("my placemark.kml"))  
{  
    kml.Save(stream);  
}
```

Obrázek 14 - Příklad použití knihovny SharpKml

4.8.4 JAK (Java API for KML)

JAK je knihovna napsaná v jazyce Java. Lze ji stáhnout z:

<http://labs.micromata.de/projects/jak.html>

Příklad použití knihovny je na Obr. 15.

```
final Kml kml = new Kml();  
kml.createAndSetPlacemark()  
    .withName("London, UK").withOpen(Boolean.TRUE)  
    .createAndSetPoint().addToCoordinates(-0.126236, 51.500152);  
kml.marshall(new File("HelloKml.kml"));
```

Obrázek 15 - Příklad použití knihovny JAK

5. Analýza celého systému

Celá aplikace byla pojata spíše jako technologické demo. Jejím účelem bylo dokázat, že vybrané technologie lze pospojovat dohromady a zobrazit v mapě smysluplné informace. Nebylo tedy účelem vytvořit aplikaci, která bude nabízet něco nového a lepšího oproti konkurenčním aplikacím.

Zároveň ale bylo žádoucí, aby aplikace dávala smysl. Bylo tedy zvoleno zobrazení počasí v mapě a aplikaci byl dán název *PočasíCZ*.

PočasíCZ sestává z webového robota, který parsuje webové stránky předpovědi počasí a parsovaná data ukládá do databáze. Data jsou přes webové služby poslány klientské aplikaci a ta zobrazí teplotu a typ počasí (oblačno, slunečno, ...) ve formě malého obrázku, a to přímo v mapě, „vznášející“ se nad konkrétním městem, ke kterému se předpověď vztahuje. Popojetím s posuvníkem se mění čas, pro který je předpověď zobrazena.

Aby aplikace nebyla příliš jednoduchá, byla přidána stránka se seznamem měst a stránka, kde jsou zobrazeny teploty v určitém městě. Tyto stránky také slouží pro demonstraci toho, jak je aplikace rychlá, pokud je vyžádána menší porce dat.

5.1 Specifikace požadavků

Zadáním je vytvořit aplikaci, která bude do jisté míry multiplatformní v tom smyslu, že serverová část bude univerzální a již nebude nutné ji měnit, pokud se rozhodneme přidat klientskou aplikaci na další platformě. Aplikace musí umět zobrazit geografická data v mapě a obě části musí spolu vzájemně komunikovat prostřednictvím webových služeb. Klientská aplikace musí být pro Windows Phone 8.1, serverová aplikace pak musí být nasazena na Microsoft Azure.

5.2 Výběr technologií

Zadáním práce je vytvoření dvou částí multiplatformní aplikace: Serverové části a mobilní aplikace pro Windows Phone 8. Nyní všechny telefony s OS Windows Phone 8 byly automaticky upgradovány na Windows Phone 8.1, takže mobilní aplikace bude ve Windows Phone 8.1.

Je potřeba učinit několik technologických rozhodnutí. Výsledná kombinace technologií bude tvořit výslednou aplikaci (tzn. obě její části). Technologická rozhodnutí jsou následující:

1. V jaké ze služeb Microsoft Azure a v jakém jazyce bude naimplementována serverová část aplikace.
2. Jaká forma webových služeb bude použita pro komunikaci mezi oběma částmi.
3. Jaký formát dat bude použit k přenosu geografických dat.
4. V jakém jazyce bude naimplementována mobilní aplikace.
5. Jaká technologie se použije pro vykreslení geografických dat do mapy.

5.2.1 Služba Microsoft Azure pro serverovou část aplikace

Je zapotřebí vybrat jednu ze služeb Microsoft Azure pro hostování PaaS:

- Webb Apps
- Mobile Services
- Cloud Services
- Media Services
- Scheduler

Z těchto služeb jednoznačně vyčnívá služba Mobile Services, protože zahrnuje jak nasazení standardní aplikace (podobně jako Cloud Services), tak řešení webových služeb mezi mobilním zařízením a serverem. Vybráním této služby se tedy výrazně zjednoduší výběr webových služeb (viz kapitolu 5.2.2).

Služba Mobile Services má i další výhody:

- Již v sobě obsahuje službu Scheduler, která umožňuje nasazení jobu, který se bude pravidelně spouštět v určenou dobu. Takový job je nutností pro typ aplikace, která byla vybrána.
- Její součástí je Entity Framework – dobrá knihovna pro Objektově relační zobrazování (z angl. Object-relational mapping, ORM) – která usnadňuje práci s databázemi.
- Umožňuje připojení k libovolné SQL databázi, i k jiným způsobům uložení dat.

Proto byla vybrána služba Mobile Services.

Jako databáze byla vybrána výchozí databáze – Microsoft Azure SQL Database. Její výhodou je snazší propojení se službou Mobile Services (velké množství tutoriálů i ukázkové projekty).

Jako jazyk implementace byl vybrán C# pouze z důvodu toho, že jsem s ním více seznámen.

5.2.2 Výběr webových služeb

Jediná ze služeb Microsoft Azure, která v sobě již zahrnuje webové služby dělané na míru pro mobilní aplikace, je služba Mobile Services. Ta byla vybrána k implementaci serverové části aplikace.

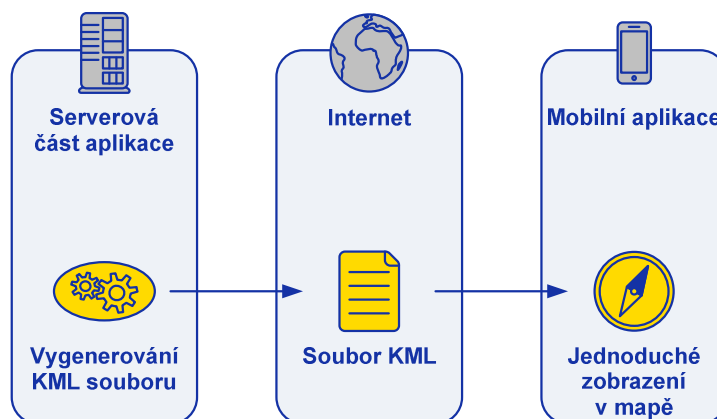
Webové služby Microsoft Azure Mobile Services (dále jen MAMS) jsou založeny na bázi REST (viz kapitolu 2.4).

Podpora platform na straně klienta je velká – Windows Store Apps, Windows Phone, Android, iOS a další [23]. Stačí jen stáhnout příslušnou knihovnu pro komunikaci s MAMS.

5.2.3 Přenos geografických dat

Při zadávání této diplomové práce bylo předpokládáno, že bude použit formát KML. Na serveru by se vygeneroval KML soubor, webovými službami by se přenesl do mobilní

aplikace, a ta by ho poté (jednoduchým příkazem) vykreslila do mapy. Tento přístup je znázorněn na Obr. 16.

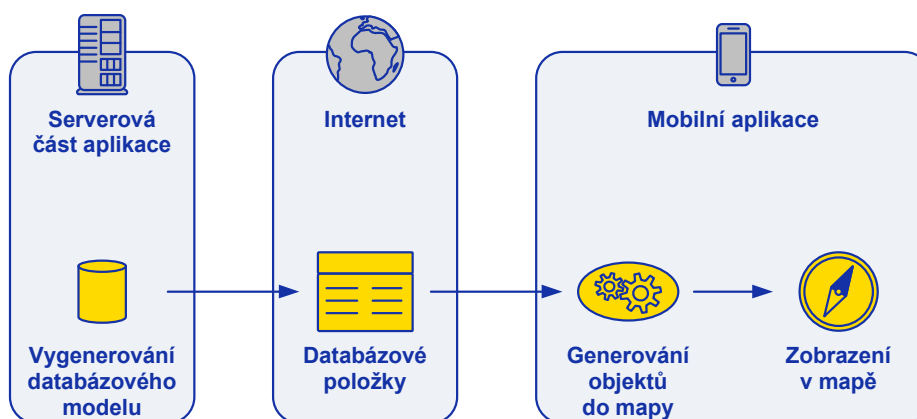


Obrázek 16 - Původně uvažovaný způsob přenosu geografických dat

Od podpory KML souborů je ale v poslední době ustupováno (podrobněji v kapitole 4.3). Navíc žádný jednoduchý způsob zobrazení KML souboru na Windows Phone 8.1 není. Jediný možný způsob by zahrnoval parsování KML souboru a vykreslování objektu po objektu (podrobněji v kapitole 4.5). S těmito novými poznatky je tedy zbytečné generovat KML soubor jen proto, aby byl na klientské aplikaci opět rozparsován.

Byl tedy zvolen způsob vytvoření vlastního databázového modelu. Veškerá geografická data budou uložena v databázi na serveru, odkud se jednotlivé databázové položky budou stahovat do klientské aplikace. Ty budou poté objekt po objektu vykreslovány do mapy.

Při použití služby MAMS (viz kapitolu 5.2.1) se implementace tohoto způsobu stává jednodušší, jelikož použití služby je jednoduché a dobře dokumentované. Zvolený způsob přenosu geografických dat je znázorněn na Obr. 17.



Obrázek 17 - Zvolený způsob přenosu geografických dat

5.2.4 Způsob implementace mobilní aplikace

Mobilní aplikace je naimplementována ve WPF (jazyk XAML) a v jazyku C#. Byla zapotřebí knihovna pro práci s MAMS. Ta je rozdělena na dvě knihovny

reprezentované dvěma balíky – `Microsoft.WindowsAzure.Mobile` a `Microsoft.WindowsAzure.Mobile.Ext`.

5.2.5 Technologie pro vykreslení geografických dat do mapy

Pro vykreslení geografických dat do mapy byla použita knihovna Windows Runtime API (verze Windows Phone 8.1), což je knihovna, která je přítomna v každé aplikaci na OS Windows Phone 8.1. Tato knihovna v sobě obsahuje komponentu `MapControl`, ve které se zobrazuje mapa. Pro použití komponenty `MapControl`, viz kapitulu 4.6.

Na operačním systému Windows Phone 8 se k zobrazení mapy používala komponenta `Microsoft.Phone.Maps.Controls.Map` [19].

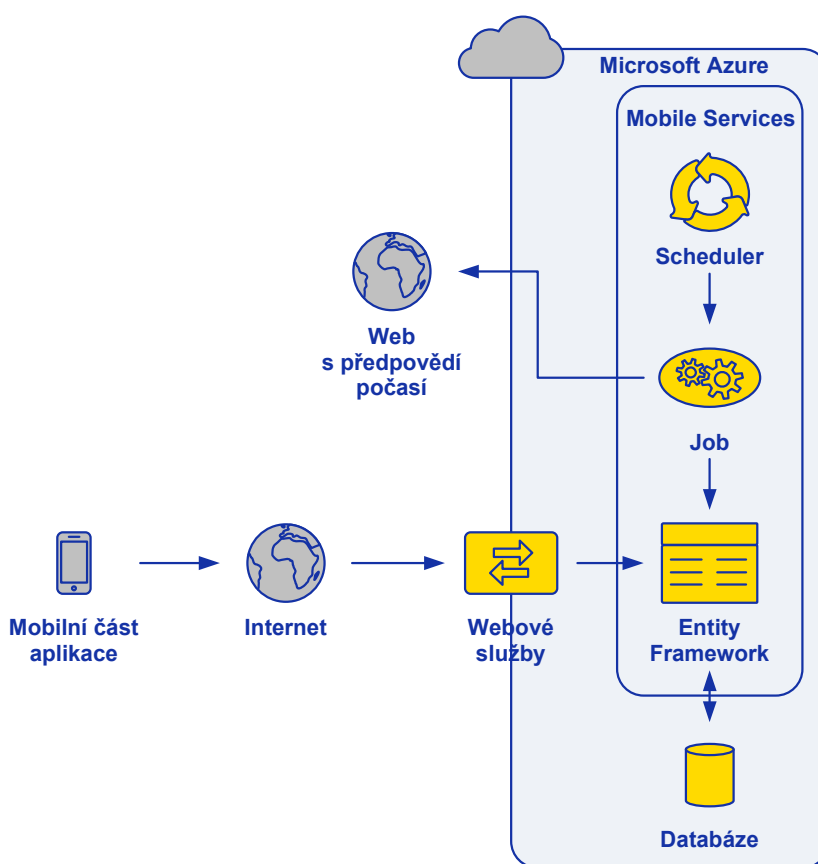
5.2.6 Publikování na Windows Store

Publikování vytvořené aplikace na Windows Store bylo přidáno jako rozšíření – nebylo to v zadání diplomové práce. Pomohlo to však při testování aplikace – bylo možné ověřit, jak rychlá bude aplikace ve skutečnosti (bez debug módu). Navíc je díky tomu možné ověřit, jak si aplikace povede oproti konkurenčním aplikacím.

6. Návrh serverové části aplikace

Celá serverová aplikace je nasazena na cloudové službě Microsoft Azure. Sestává ze služby Mobile Services a databáze Microsoft Azure SQL Database založené na MS SQL Server.

V rámci Mobile Services pak budou využity další služby – ScheduledJobs, ve které bude vytvořen job, který je hlavní částí serverové aplikace, Scheduler, který umožňuje automatické opakované spuštění jobu po stanovených intervalech, vše je konfigurovatelné z Management Portalu, a Entity Framework, který umožňuje mapování objektů na databázové položky. Služba Mobile Services také zajišťuje webové služby ke komunikaci s klientskou aplikací. Na Obr. 18 jsou zachyceny hlavní funkce serverové aplikace.



Obrázek 18 - Návrh serverové části aplikace

6.1 Automatické spuštění jobu

Web s počasím, odkud se stahují data (což je v současné konfiguraci web <http://yr.no>), je aktualizován každou hodinu, vždy cca 15 minut po celé hodině (tedy 13:15, 14:15, atd.). K tomuto času byla přičtena bezpečnostní rezerva a služba Scheduler je nakonfigurována tak, aby každou hodinu, vždy 22 minut po celé hodině (tedy 13:22, 14:22, atd.) spouštěla job.

6.2 Načtení konfiguračního souboru

Když je job spuštěn, první, co udělá, je načtení konfiguračního souboru ve formátu CSV. V něm je seznam měst a u každého města jsou následující položky:

- Název města
- URL adresa webové stránky, ze které jsou staženy údaje o počasí v konkrétním městě
- Zeměpisná délka (angl. longitude)
- Zeměpisná šířka (angl. latitude)

6.3 Parsování

Další činnost, kterou job provede, je načtení webových stránek a jejich naparsování.

Pro každé město z konfiguračního souboru je načtena webová stránka, jejíž vzhled je znázorněn na Obr. 19. Zdrojový kód (HTML) této stránky je naparsován. Jsou parsovány hlavičky jednotlivých tabulek i jejich řádky.

Today and tonight, 08 May 2015					Tomorrow, 09 May 2015				
Time	Forecast	Temp.	Precip.	Wind	Time	Forecast	Temp.	Precip.	Wind
Friday 12:00		19°	0 mm	Gentle breeze, 5 m/s from southeast	Saturday 06:00		12°	0 – 0.1 mm	Light air, 1 m/s from east-northeast
Friday 13:00		20°	0 mm	Gentle breeze, 4 m/s from southeast	Saturday 07:00		13°	0 mm	Light air, 1 m/s from north-northwest
Friday 14:00		20°	0 mm	Gentle breeze, 4 m/s from southeast	Saturday 08:00		13°	0 mm	Calm, 0 m/s
Friday 15:00		21°	0 mm	Gentle breeze, 4 m/s from southeast	Saturday 09:00		15°	0 mm	Light air, 2 m/s from west-southwest
Friday 16:00		21°	0 mm	Gentle breeze, 4 m/s from southeast	Saturday 10:00		17°	0 mm	Light air, 2 m/s from west
Friday 17:00		21°	0 mm	Gentle breeze, 4 m/s from east-southeast	Saturday 11:00		20°	0 mm	Light air, 1 m/s from west
Friday 18:00		21°	0 mm	Gentle breeze, 4 m/s from east-southeast	Saturday 12:00		22°	0 – 0.1 mm	Gentle breeze, 4 m/s from west
Friday 19:00		21°	0 mm	Gentle breeze, 4 m/s from east-southeast	Saturday 13:00		22°	0 – 0.1 mm	Gentle breeze, 5 m/s from west
Friday 20:00		20°	0 mm	Gentle breeze, 4 m/s from east-southeast	Saturday 14:00		22°	0 mm	Gentle breeze, 5 m/s from west-southwest
Friday 21:00		19°	0 mm	Gentle breeze, 4 m/s from east-southeast	Saturday 15:00		22°	0 mm	Moderate breeze, 8 m/s from west-southwest

Obrázek 19 - Příklad formátu webové stránky s počasím

6.3.1 Parsování hlavičky

Nejprve jsou naparsovány údaje z hlavičky jednotlivých tabulek. Obvykle jsou přítomny tabulky dvě:

Today and tonight

Tomorrow

V některých denních dobách jsou přítomny tabulky tři:

Tonight
Tomorrow
Day after tomorrow

Příklad zdrojového kódu hlavičky tabulky je na Obr. 20 (některé části byly nahrazeny třemi tečkami pro zjednodušení).

```
<table class="...">
  <caption>
    <strong>Today and tonight,</strong> 08 May 2015
  </caption>
  ...
</table>
```

Obrázek 20 - Příklad zdrojového kódu hlavičky tabulky

Žlutě zvýrazněné části jsou z textu vyparsovány a uloženy. Tedy 1. den v textové formě a 2. den ve formátu, který je možno převést na datum.

6.3.2 Parsování jednotlivých řádek

Pro každou řádku v předpovědi je vytvořena datová položka. Příklad zdrojového kódu jedné řádky předpovědi je na Obr. 21 (některé části byly nahrazeny třemi tečkami pro zjednodušení).

```
<tr>
  <td scope="row">Friday <strong>20:00</strong></td>
  <td title="Cloudy. For the period: 20:00-21:00">
    
  </td>
  <td class="temperature plus" title="...">20°</td>
  <td class="precipitation" title="...">0 mm</td>
  <td class="txt-left" title="...">
    <img ... />Gentle breeze, 4 m/s from east-southeast
  </td>
</tr>
```

Obrázek 21 - Zdrojový kód jednotlivých řádek tabulky

Žlutě zvýrazněné části jsou z textu vyparsovány a uloženy. Tedy 1. textová hodnota času, 2. typ počasí a 3. teplota.

6.4 Uložení změn do databáze

Všechny položky v databázi jsou smazány a jsou nahrazeny novými položkami:

- Do tabulky s městy (třída `Town`) jsou uloženy některé údaje z konfiguračního souboru: jméno města a jeho zeměpisná délka a šířka
- Do tabulky s počasím (třída `WeatherItem`) jsou uložena následující data
 - Pro každé město a pro každou hodinu uvedenou na webu jedna položka, která obsahuje tyto údaje (viz kapitolu 6.3):
 - Den v textové formě (naparsováno z hlavičky tabulky)

- Hodinu v textové formě (naparsováno z řádky)
- Datum v datovém typu `DateTime` (naparsováno z hlavičky tabulky)
- Typ počasí (naparsováno z řádky)
- Teplotu (naparsováno z řádky)
- Název města, ke kterému se předpověď vztahuje (načteno z konfiguračního souboru)

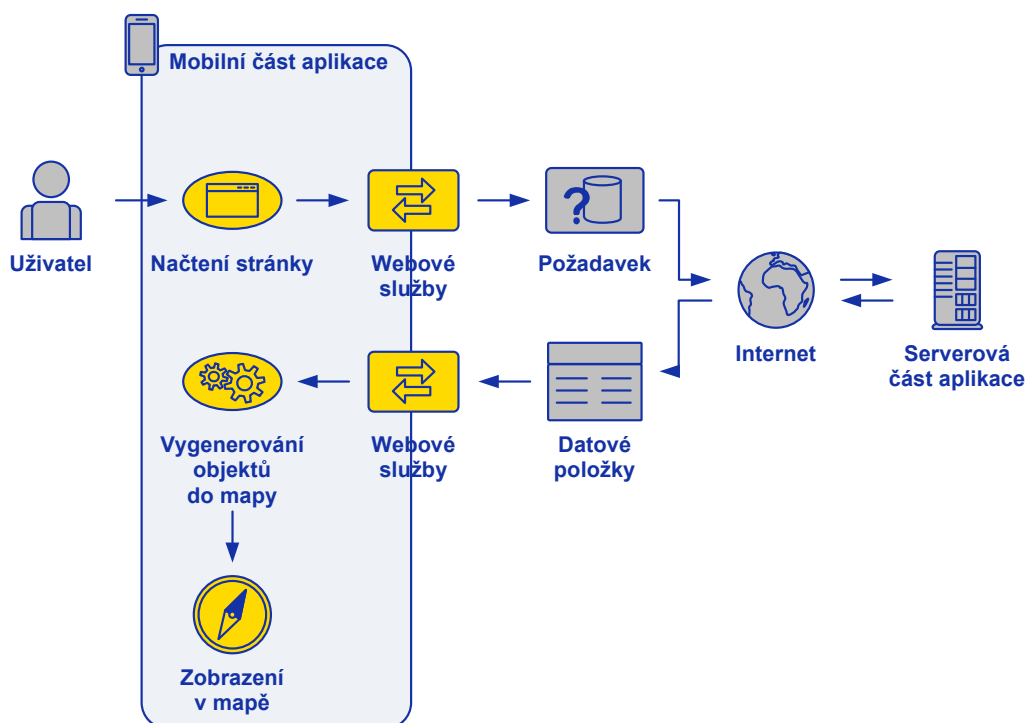
Všechny tyto kroky se provedou v jedné databázové transakci, takže se nemůže stát, že by v určitém okamžiku byly z databáze odstraněny staré položky, ale ještě by nebyly přidány nové.

7. Návrh mobilní aplikace

Když je uživatelem načtena stránka, zobrazí se prázdná a spustí se asynchronní požadavek webových služeb pro získání dat ze serveru (viz kapitolu 7.1). V požadavku lze specifikovat omezující podmínku, kterou musí vrácená data splňovat. Díky tomu lze omezit velikost vrácené odpovědi.

Požadavek je zpracován serverem a ten zašle zpět odpověď obsahující datové entity. Data jsou poté zpracována a buď zobrazena (na 2 stránkách, které neobsahují mapu), nebo vygenerována do mapy objekt po objektu (na zbývajících stránce, kde se informace zobrazují v mapě).

Celý tento proces je znázorněn na Obr. 22.



Obrázek 22 - Návrh mobilní aplikace

7.1 Asynchronní požadavek

Načítání dat do stránky se děje prostřednictvím asynchronního volání. Když je uživatelem načtena stránka, zobrazí se prázdná a spustí se asynchronní požadavek webových služeb pro získání dat ze serveru. Mezitím co aplikace čeká na získání dat, je uživateli graficky oznamováno, že se na něco čeká.

Díky tomu, že je požadavek asynchronní, je celá aplikace více responzivní. Mezitím, co se čeká na odpověď ze serveru, je vlákno, které obhospodařuje GUI, volné a uživatel má možnost interakce s GUI. Tento přístup je již dnes u všech mobilních aplikací standardní.

7.2 Entity vrácené serverem

Když požadavek dosáhne serveru, je zpracován a Entity Framework na serveru získá data z SQL databáze. Právě přes Entity Framework je (za využití webových služeb) poslána odpověď zpět klientské aplikaci, takže odpověď neobsahuje databázové položky, ale datové entity, tzn. objekty objektového jazyka.

7.3 Navigace

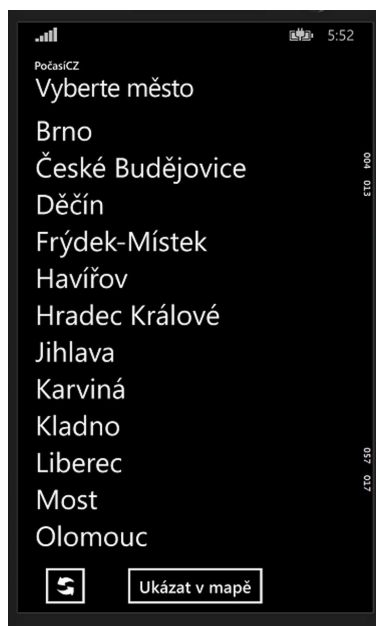
Aplikace obsahuje celkem tři stránky, mezi kterými lze navigovat hardwarovým tlačítkem zpět na telefonu.

Ťuknutím zpět na třetí stránce je zobrazena druhá stránka. Ťuknutím zpět na druhé stránce je zobrazena první stránka. Ťuknutí zpět na první stránce aplikace nezpracovává, toto ťuknutí způsobí pozastavení aplikace (tzn. uživatel „vyjede“ z aplikace ven).

Pokud je uživatel vrácen na stránku ťuknutím na tlačítko zpět, tato stránka není znovu načtena. Její obsah zůstává v paměti z posledního načtení. Pokud si uživatel přeje stránku znovu načíst, musí ťuknout na tlačítko obnovení.

7.4 První stránka – Vybrání města

Když uživatel prvně spustí aplikaci, je mu zobrazena stránka s nadpisem *Vyberte město* a seznamem měst, které bylo načteno ze serveru (tzn. města z konfiguračního souboru, viz kapitolu 6.2). V zápatí stránky jsou zobrazena dvě tlačítka. Jedno pro obnovení obsahu stránky (znovu načte obsah stránky ze serveru) a jedno pro zobrazení třetí stránky – viz kapitolu 7.6. Toto tlačítko nese název *Ukázat v mapě*. Uživatel může ťuknout na některý z názvů měst a je přenesen na druhou stránku, která stáhne a zobrazí informace specifická pro město, které bylo vybráno na první stránce. Vzhled první stránky je zobrazen na Obr. 23.



Obrázek 23 - Vzhled první stránky

7.5 Druhá stránka – Počasí ve městě

Tato stránka je vždy zobrazena pro jedno konkrétní město. Ze serveru jsou staženy všechny položky `WeatherItem`, které se týkají tohoto města. Teplota těchto položek je zobrazena v přehledném seznamu. V zápatí stránky jsou tři prvky:

- Zatřítko (z angl. checkbox) *Vždy zobrazovat toto město*, viz kapitolu 7.5.1
- Tlačítko pro obnovení obsahu stránky (znovu načte obsah stránky ze serveru)
- Tlačítko *Ukázat v mapě*, které zobrazí třetí stránku, viz kapitolu 7.6

Vzhled druhé stránky je zobrazen na Obr. 24.



Obrázek 24 - Vzhled druhé stránky

7.5.1 Funkce zatřítko ,Vždy zobrazovat toto město‘

Kdykoli uživatel zatrhne toto tlačítko, je do trvalých konfiguračních souborů aplikace uložena informace, že se má tato stránka zobrazovat při spuštění aplikace. Při příštím spuštění aplikace se tedy uživateli nezobrazí první stránka, ale rovnou druhá stránka, do které se načtou informace právě o tom městě, které bylo načteno naposledy, když uživatel zatřhl zatřítko.

Poznámka: I když je uživateli zobrazena druhá stránka při spuštění, je první stránka načtena na pozadí. To proto, aby při stisknutí tlačítka zpět byl přechod na první stránku rychlejší.

Při načtení druhé stránky je vždy porovnáno aktuálně zobrazované město s městem uloženým v trvalých konfiguračních souborech aplikace. Pokud jsou shodné, je hned při načtení stránky zatřítko zatřžené. *Příklad:* Uživatel má zobrazeno město Praha a zatrhne zatřítko. Z této stránky může vyjet a zobrazovat jiná města. Jakmile si ale opět zobrazí město Praha, je zatřítko zatřžené hned při načtení stránky.

Pokud uživatel zatržítka odzatrhne, je tato informace opět propagována do trvalých konfiguračních souborů aplikace a při příštím spuštění aplikace je zobrazena první stránka.

7.6 Třetí stránka – mapa

Při navigaci na třetí stránku jsou ze serveru staženy všechny položky WeatherItem i všechny položky Town.

Přiblížení a vycentrování mapy záleží na tom, odkud bylo na tuto stránku navigováno. Při navigaci z první stránky je přiblížení menší, tak aby byla zobrazena celá Česká republika (při zobrazení na výšku). Pokud bylo na tuto stránku navigováno ze druhé stránky, je mapa vycentrována kolem konkrétního města, které bylo zobrazeno na druhé stránce. Přiblížení je větší, takže je zobrazeno jen několik málo sousedních měst.

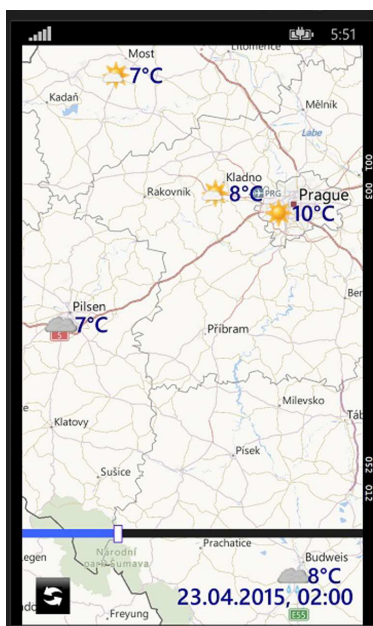
Mapa je dynamická a lze s ní provádět všechny standardní operace, tedy: Posouvání, přibližování, oddalování. Při přibližování a oddalování se automaticky ukazují či ztrácejí další detaily mapy v závislosti na úrovni přiblížení.

U každého z načtených měst se v mapě zobrazují informace o počasí. Konkrétně je to ikona typu počasí (oblačno, slunečno, přehánky, ...), stejná jako na stránce <http://yr.no>, ze které se data stahují, a teplota ve stupních Celsia.

V zápatí stránky jsou další tři prvky:

- Posuvník, kterým uživatel nastavuje čas, pro který se zobrazuje počasí
- Tlačítko pro obnovení obsahu stránky (znovu načte obsah stránky ze serveru)
- Datum a čas zobrazovaného počasí

Vzhled třetí stránky je zobrazen na Obr. 25.



Obrázek 25 - Vzhled třetí stránky

7.6.1 Zobrazení počasí pro chybějící data

Stránka <http://yr.no>, odkud jsou data čerpána, poskytuje různou úroveň detailu pro různá města. Pro města na západě České republiky poskytuje údaje o počasí pro každou hodinu. Avšak pro města na východě České republiky poskytuje pouze údaje pro každou třetí hodinu.

Při zobrazování položek v mapě pak tedy nastává problém, pokud uživatel vybere zobrazení takového času, pro který jsou k dispozici data pro některá města, ale pro jiná města data k dispozici nejsou.

Příklad:

Stránka <http://yr.no> zobrazuje údaje pro město Praha pro časy 14:00, 15:00, 16:00, 17:00, atd. Avšak pro město Brno zobrazuje údaje pouze pro časy 14:00, 17:00, atd.

Pokud tedy uživatel chce zobrazit počasí například pro čas 15:00 nebo 16:00, je situace řešena následovně:













Pro města, která nemají k dispozici údaje pro tyto časy, je zobrazena nejbližší *předchozí* položka počasí.

Může se stát, že k dispozici není nejbližší předchozí položka počasí. Příklad:

Aktuální čas je 14:25. Pro město Praha jsou k dispozici nejbližší údaje pro časy: 15:00, 16:00, 17:00, atd. Pro město Brno však jsou k dispozici nejbližší údaje pro časy: 17:00, 20:00, atd. To znamená, že pro položky 16:00 a 17:00 pro Brno neexistují nejbližší *předchozí* položky. Proto jsou pro tyto časy zobrazeny v mapě nejbližší *následující* položky.

7.6.2 Tabulka typů počasí

V tabulce 7 je zobrazen seznam typů počasí, které lze zobrazit v mapě. Typy počasí jsou v tabulce ve trojici. Nejprve ikona, která se zobrazuje v mapě, poté český význam typu počasí a poté anglický název typu počasí, napařovaný z webové stránky.

	Slunečno	Clear sky		Silný déšť	Heavy rain
	Polojasno	Fair		Sníh	Snow
	Oblačno	Partly cloudy		Lehké přeháňky	Light rain showers
	Zataženo	Cloudy		Silné přeháňky	Heavy rain showers
	Přeháňky	Rains showers		Lehký déšť	Light rain
	Déšť	Rain		Lehký sníh	Light snow

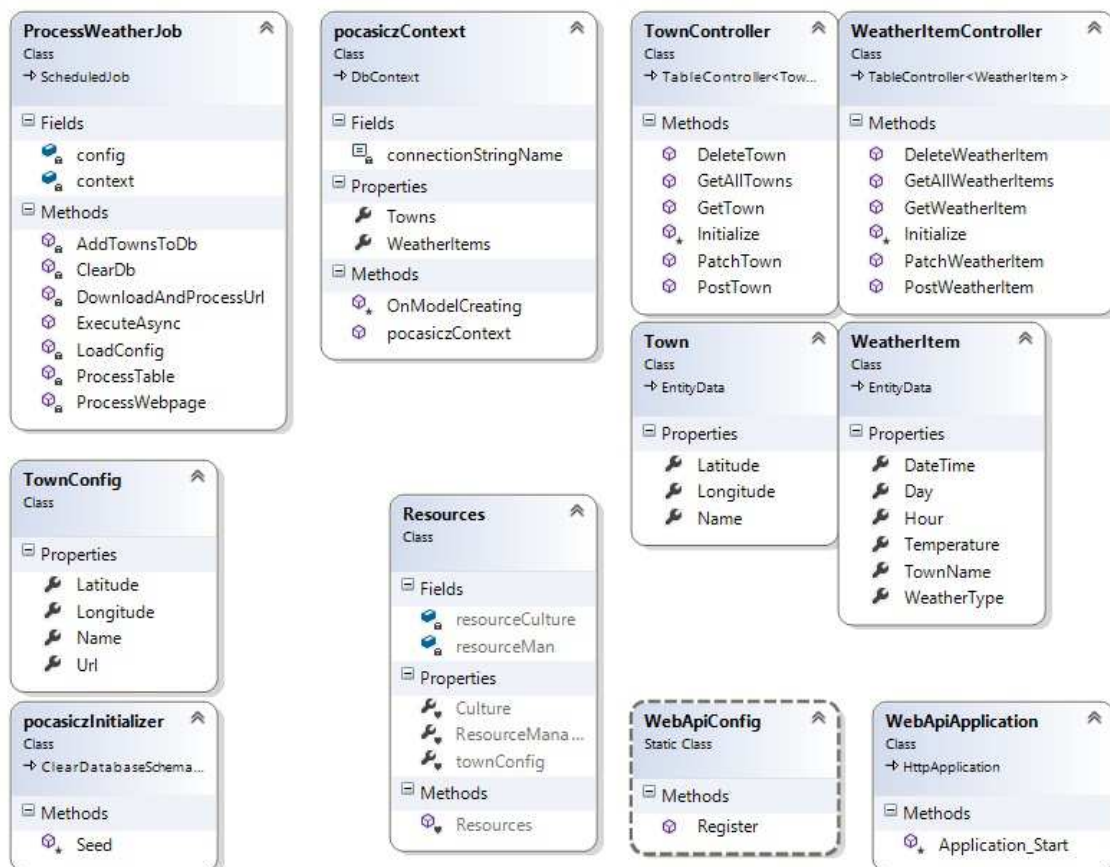
Tabulka 7 - Seznam typů počasí

8. Implementace serverové části aplikace

V této kapitole je popsán způsob implementace serverové části aplikace.

8.1 Obecný model

V této kapitole je zobrazen diagram tříd celé serverové části aplikace (na Obr. 26) a jsou popsány nejdůležitější třídy implementace.



Obrázek 26 - Diagram tříd

8.1.1 ProcessWeatherJob

Tato třída obsahuje celý job, který je spuštěn plánovačem (služba Scheduler) Microsoft Azure. Vstupním bodem této třídy je metoda `ExecuteAsync`. Job načte konfigurační soubor, stáhne všechny zdrojové kódy stránek v konfiguračním souboru a poté naparsuje data ze zdrojových kódů stránek a uloží je do databáze.

8.1.2 TownConfig

Obsahuje konfigurační data pro jedno město načtená z konfiguračního souboru. To je čtveřice dat: název města, URL stránky, kde se vyskytuje počasí vztažené k tomuto město, zeměpisná délka a zeměpisná šířka.

8.1.3 Town

Entitní třída, která obsahuje stejné položky, jako příslušná databázová tabulka, kterou vytvoří Entity Framework.

8.1.4 WeatherItem

Entitní třída, která obsahuje stejné položky, jako příslušná databázová tabulka, kterou vytvoří Entity Framework.

8.1.5 TownController

Třída, která definuje operace, které lze provádět s entitní třídou Town. To jsou operace jako získání položky, úprava položky, smazání položky apod.

8.1.6 WeatherItemController

Třída, která definuje operace, které lze provádět s entitní třídou WeatherItem. To jsou operace jako získání položky, úprava položky, smazání položky apod.

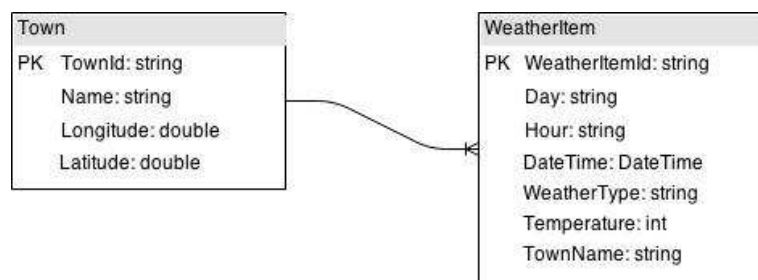
8.1.7 pocaszContext

Kontext Entity Frameworku, který definuje přístup k databázi, seznamy instancí entitních tříd, nastavení toho, kdy se má vytvářet schéma databáze (tzn. struktura tabulek, jednotlivé sloupce, apod.), a další.

8.2 Datové struktury

Databázové tabulky i jejich data jsou do databáze Microsoft Azure SQL Database automaticky vygenerovány technologií Entity Framework. Třídy Town a WeatherItem jsou odděděny od EntityData, čímž je Entity Frameworku dán signál, že právě tyto dvě třídy se budou ukládat do databáze.

Databázový model je znázorněn na Obr. 27.



Obrázek 27 - Databázový model

8.3 Použité technologie

Pro implementaci serverové části bylo použito několik různých technologií. Především MAMS, Scheduler a Entity Framework (více popsány v kapitole 5.2.1).

Dále je to knihovna HtmlAgilityPack pro práci se zdrojovým kódem HTML stránky a k usnadnění parsování informací z HTML stránky. Největší výhodou této knihovny je

možnost práce s HTML stránkami, které nejsou validní z hlediska standardu XML, což je dnes velké množství HTML stránek, například kvůli vloženému javascriptu.

Byla také použita databáze Microsoft Azure SQL Database, více popsaná v kapitole 3.2.1.

8.4 Poznámky k realizaci

V této kapitole jsou zmíněny všechny důležité zvláštnosti implementace a nasazení této serverové části aplikace.

8.4.1 Automatická konverze času

Při prvním nasazení hotové aplikace na Microsoft Azure byl objeven zajímavý problém, který se při testování na lokálním stroji neprojevil. Při použití webových služeb byl mezi serverem a mobilní aplikací datový typ `DateTime` automaticky posouván o určitý počet hodin v závislosti na rozdílu časových zón serveru a klienta.

Toto bylo velice neočekávané chování, jelikož datový typ `DateTime` v sobě neobsahuje úplné informace o tom, v jaké časové zóně je vyjádřen. Umožňuje však omezenou informaci o časové zóně – `DateTime` obsahuje vlastnost `Kind`, která může být nastavena na `Local`, `Unspecified`, nebo `Utc`.

První řešení tedy spočívalo v nastavení vlastnosti `Kind` na `Utc`, to znamená specifikováním, že tato konkrétní instance datového typu `DateTime` je v čase UTC. Toto řešení ale nepomohlo – webové služby i nadále konvertovaly tento datový typ stejným způsobem.

Druhé řešení spočívalo v použití metody `TimeZoneInfo.ConvertTime` a použití rozdílu mezi časovou zónou českého času a časovou zónou, ve které byl server. Toto řešení již zafungovalo.

MOBILE SERVICES: SCHEDULER

Create new job

JOB NAME

 ?

SCHEDULE

Every

On demand

You can add a schedule later.

Obrázek 28 - Vytvoření nového jobu na Microsoft Azure

8.4.2 Nastavení nového jobu na Microsoft Azure

Přes Management Portal (viz kapitolu 3.2.1) lze snadno vytvořit job, kterému lze nastavit rozvrh na pravidelné spouštění. Stačí vyplnit název třídy, která dědí od třídy Scheduled Job, oříznutý o slovo Job. V našem případě tedy máme třídu:

```
public class ProcessWeatherJob : ScheduledJob
```

A do Management Portal vyplníme text `ProcessWeather`. Microsoft Azure již sám vyhledá správnou třídu a vytvoří z ní job. Proces vytváření jobu je zachycen na Obr. 28.

Nastavení rozvrhu daného jobu se dá udělat i složitějším způsobem, který umožní jeho podrobnější nastavení.

Každý job může vypisovat údaje do logu, který je taktéž přehledně zobrazen přes Management Portal. Výpis jobu `ProcessWeather` je zobrazen na Obr. 29.

MESSAGE	SOURCE	TIME STAMP
Changes saved. Job ended at 4/20/2015 8:34:40 AM	ExecuteAsync	Mon Apr 20 2015, 10:34:40
URLs processed at 4/20/2015 8:34:33 AM	ExecuteAsync	Mon Apr 20 2015, 10:34:33
Towns added to DB at 4/20/2015 8:34:23 AM	ExecuteAsync	Mon Apr 20 2015, 10:34:23
Config loaded at 4/20/2015 8:34:23 AM	ExecuteAsync	Mon Apr 20 2015, 10:34:23
DB cleared at 4/20/2015 8:34:23 AM	ExecuteAsync	Mon Apr 20 2015, 10:34:23
Job started at 4/20/2015 8:34:21 AM	ExecuteAsync	Mon Apr 20 2015, 10:34:21

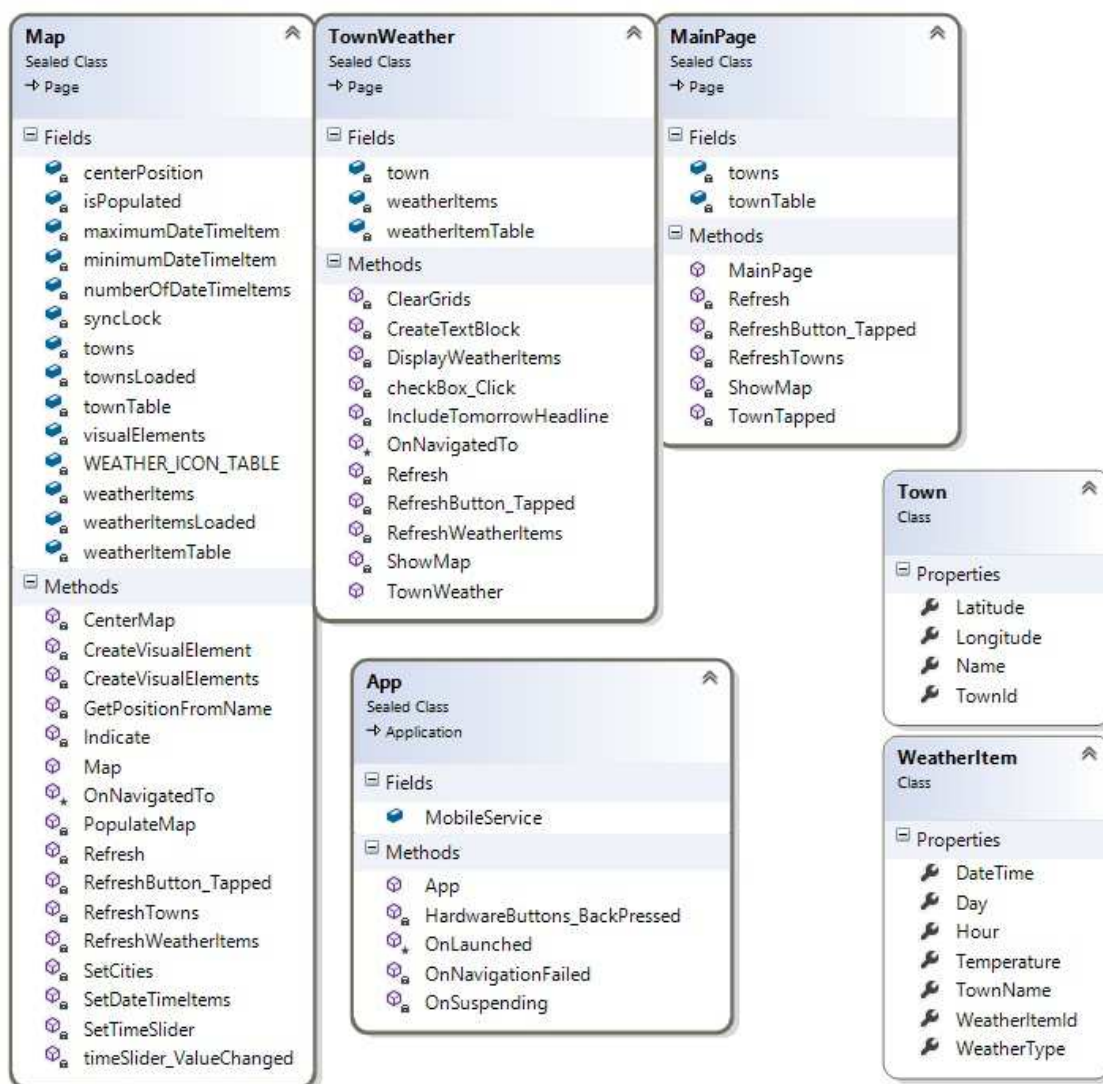
Obrázek 29 - Log jobu `ProcessWeatherJob`

9. Implementace mobilní aplikace

V této kapitole je popsán způsob implementace mobilní aplikace ve Windows Phone 8.1.

9.1 Obecný model

V této kapitole je zobrazen diagram tříd celé mobilní aplikace (na Obr. 30) a jsou popsány nejdůležitější třídy implementace.



Obrázek 30 - Diagram tříd

9.1.1 App

Definuje metody a atributy platné pro celou aplikaci. Vytváří Frame – oblast displeje, do které se načítají jednotlivé stránky. Zpracovává událost **HardwareButtons_BackPressed**, díky čemuž je možné navigovat mezi stránkami aplikace pomocí hardwarového tlačítka zpět.

9.1.2 MainPage

První stránka, na které je zobrazen seznam měst. Uživatel si z nich může jedno vybrat a tím je přesměrován na druhou stránku.

9.1.3 TownWeather

Druhá stránka, která zobrazuje teplotní údaje v jednom městě. Na této stránce je možné ťuknout na tlačítko *Ukázat v mapě*, čímž je uživatel přesměrován na třetí stránku.

9.1.4 Map

Třetí stránka, která zobrazuje počasí v mapě. Objekty `Town` a `WeatherItem` načítá najednou – spustí dva asynchronní procesy – jeden pro získání seznamu položek `Town` a jeden pro získání seznamu položek `WeatherItem`. Počká, až se dokončí oba procesy, a poté vytvoří všechny grafické komponenty.

9.1.5 Town

Entitní třída, která musí mít stejný formát jako entitní třída serverové aplikace. Používá direktivy knihovny `Json`.

9.1.6 WeatherItem

Entitní třída, která musí mít stejný formát jako entitní třída serverové aplikace. Používá direktivy knihovny `Json`.

9.2 Datové struktury

Viz kapitolu 8.2, kde je popsán datový model serverové části aplikace. Datový model je stejný pro serverovou i klientskou část aplikace.

9.3 Použité technologie

K implementaci prezenční vrstvy mobilní aplikace byla použita technologie WPF (Windows Presentation Foundation) a jazyk XAML (eXtensible Application Markup Language), který vychází ze standardu XML.

Zbytek implementace mobilní aplikace je v jazyce C# na platformě .NET pro mobilní aplikace pro Windows Phone 8.1.

Pro práci s webovými službami MAMS bylo nutné použít tyto tři knihovny:

```
Microsoft.WindowsAzure.Mobile  
Microsoft.WindowsAzure.Mobile.Ext  
Newtonsoft.Json
```

Přičemž první dvě jsou knihovny přímo nutné pro práci s MAMS, zatímco třetí knihovna umožňuje přidávat anotace k entitním třídám a pomáhá tak zjednodušit způsob získávání entitních dat ze serveru.

9.4 Poznámky k realizaci

V této kapitole jsou zmíněny všechny důležité zvláštnosti implementace a nasazení této mobilní aplikace.

9.4.1 Práce s komponentou MapControl

Základní práce s komponentou MapControl je popsána v kapitole 4.6. V aplikaci PočasíCZ byl použit následující přístup:

Byla vytvořena komponenta `StackPanel`, s horizontální orientací (tzn. obsah bude skládat vedle sebe, každá další komponenta bude vpravo od předchozí). Do této komponenty byly vloženy komponenty: `Image` (obrázek s načtenou ikonou) a `TextBox` (text zobrazující teplotu, například „12°C“).

Komponenta `StackPanel` byla poté „připnuta“ do mapy ke geografické pozici určitého města.

Z důvodu zpříjemnění uživatelské zkušenosti s aplikací díky zrychlení interakce uživatele s mapou byl použit následující postup. Již při načtení stránky jsou vytvořeny všechny výše zmíněné komponenty `StackPanel` i jejich obsah. Ale všechny komponenty, které se nevztahují k právě zobrazovanému času mají nastaven příznak `Visibility` na `false`, tzn. nejsou viditelné. V momentě, kdy se změní zobrazovaný čas, se určitým komponentám přiřadí příznak `Visibility` na `true` a tudíž se v mapě „objeví“.

9.4.2 Nasazení na Windows Store

Pro možnost nasadit aplikaci na Windows Store bylo nutné koupit si vývojářský účet pro jednotlivce. Tento účet stojí jednorázově 427Kč.

Samotné nasazení aplikace spočívá ve vytvoření instalačního package – tento package lze vytvořit ve Visual Studiu. Poté je nutné tento package ověřit nástrojem ověřování kvality softwaru od společnosti Microsoft. Pokud nástroj vyhodnotí aplikaci jako ověřenou, je možné instalační package nahrát do vývojářského portálu.

Po nahrání package je nutné vyplnit různé údaje o aplikaci a především nahrát ikony aplikace v konkrétním rozlišení a příponou `png`.

Po odsouhlasení a uložení změn prochází package ještě dalším kolem odsouhlasování na straně Microsoftu. Pokud je aplikace vyhodnocena jako neškodlivá a funkční, je aplikace publikována na Windows Store, kde si ji může každý uživatel stáhnout (pokud jsme toto nastavili při nahrávání package).

10. Ověření funkčnosti

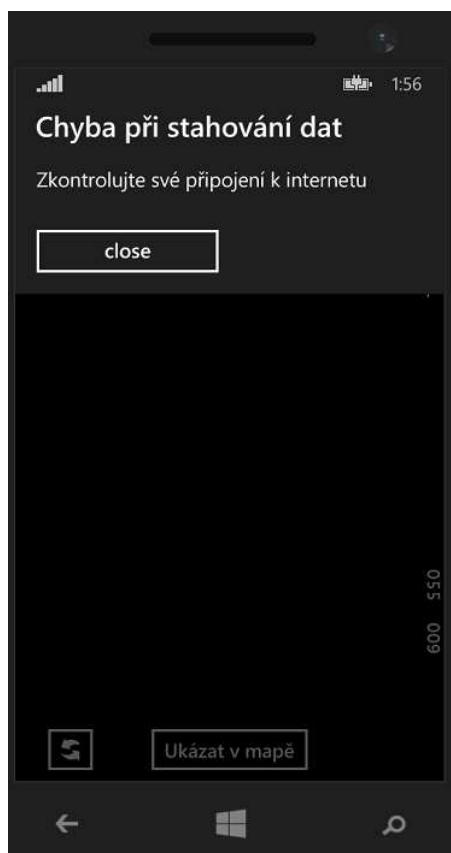
V této kapitole je popsán proces testování aplikace a měření důležitých parametrů aplikace, především časová odezva.

10.1 Testování

Aplikace byla důkladně testována. To zahrnovalo následující testy:

1. Na emulátoru oproti MAMS na lokálním stroji – na počítači Lenovo G780, dvoujádrový procesor Intel Core i5-3210M, 4GB RAM (dále jako „lokální stroj“)
2. Na emulátoru oproti MAMS nasazeným na Microsoft Azure
3. Na telefonu Nokia Lumia 630 s OS Windows Phone 8.1 (dále jako „reálné zařízení“) oproti MAMS nasazeným na Microsoft Azure
4. Na reálném zařízení oproti MAMS nasazeným na Microsoft Azure, při nainstalování aplikace z Windows Store, tedy bez možnosti debugování

Aplikace byla testována i při pohybu ve městě, kdy se mění rychlost a kvalita internetového připojení, případně internetové spojení zcela vypadne. V případě selhání internetového spojení se zobrazí chybová hláška na Obr. 31.



Obrázek 31 - Chybová hláška při špatném internetovém spojení

Aplikace je nyní stabilní. Důkladným testováním poslední její verze nebylo dosaženo žádného pádu a není známa žádná chyba aplikace.

Jak již bylo zmíněno, aplikace byla testována na telefonu Nokia Lumia 630 s 512 MB paměti RAM, který v kategorii telefonů pro Windows Phone patří k do těch nejslabších. Z toho vyplývá, že by aplikace měla plynule běžet na jakémkoliv telefonu s Windows Phone 8.1.

10.2 Měření důležitých parametrů

V této kapitole jsou naměřené časy jednotlivých časově náročných operací aplikace, jak v klientské aplikaci, tak na straně serveru.

10.2.1 Klientská část

V tabulce 8 jsou naměřené časy jednotlivých časově náročných operací aplikace. Každá hodnota byla měřena nejméně třikrát, v tabulce je vždy *uvedena průměrná hodnota*. Časy byly měřeny na stopkách. Jednotlivá čísla označují způsob testování, viz kapitolu 10.1. V případě testování 4. byly vyzkoušeny nízkorychlostní i vysokorychlostní mobilní sítě.

	Spuštění aplikace	Načtení stránky se seznamem měst	Načtení stránky s výpisem teplot ve městě	Načtení stránky s počasím v mapě
Testování 1.	6,9s	1,2s	0,9s	3,6s
Testování 2.	6,7s	1,6s	0,9s	4,4s
Testování 3.	5,3s	1,0s	0,5s	8,3s
Testování 4.	2,0s	0,7s	0,5s	8,1s

Tabulka 8 - Měření odezvy aplikace

Naměřené hodnoty nejsou nijak překvapivé. Výsledná rychlost načtení stránky je závislá na více faktorech, nejvýznamnějším faktorem je však množství dat, které je nutno stáhnout ze serveru. U načtení prvních dvou stránek je aplikace velice plynulá a z uživatelského hlediska příjemná. Při načítání třetí stránky je uživatel nucen chvíli čekat.

10.2.2 Serverová část

V tabulce 9 jsou naměřené časy doby běhu jobu. Job zapisuje do logu časy svého spuštění a svého ukončení. Z těchto časů byla odečtena naměřená hodnota. Každá hodnota byla měřena nejméně třikrát, v tabulce je vždy *uvedena průměrná hodnota*.

	Nasazený na MAMS	Job na lokálním stroji
Doba běhu jobu	16s	28s

Tabulka 9 - Měření doby běhu jobu

10.3 Srovnání s dalšími aplikacemi, které umí zobrazit počasí v mapě

10.3.1 AccuWeather

Zobrazuje radar – množství oblačnosti a srážek.

Radar je možno animovat v čase, aplikace ale neukazuje časové údaje, takže je animace téměř bezcenná. Jediný spolehlivý údaj je radarový snímek přítomnosti – ten se zobrazí jako první, před stisknutím tlačítka animace.

10.3.2 Meteoservis Lite

Zobrazuje radar – množství oblačnosti a srážek.

Zobrazená data jsou ve formě obrázku, který jde přibližovat pohybovat se v něm. Nejedná se však o skutečnou mapu.

10.3.3 Počasí (MSN Počasí)

Zobrazuje radar, ale i teplotu a ikonu podobným způsobem jako PočasíCZ.

Zobrazuje však tyto údaje jako jednu hodnotu pro celou Českou republiku. Mapu nelze nijak přiblížit.

10.3.4 Weather (The Weather Channel)

Zobrazuje radar – pouze srážky.

Animace v čase je provedena stejným způsobem jako v PočasíCZ.

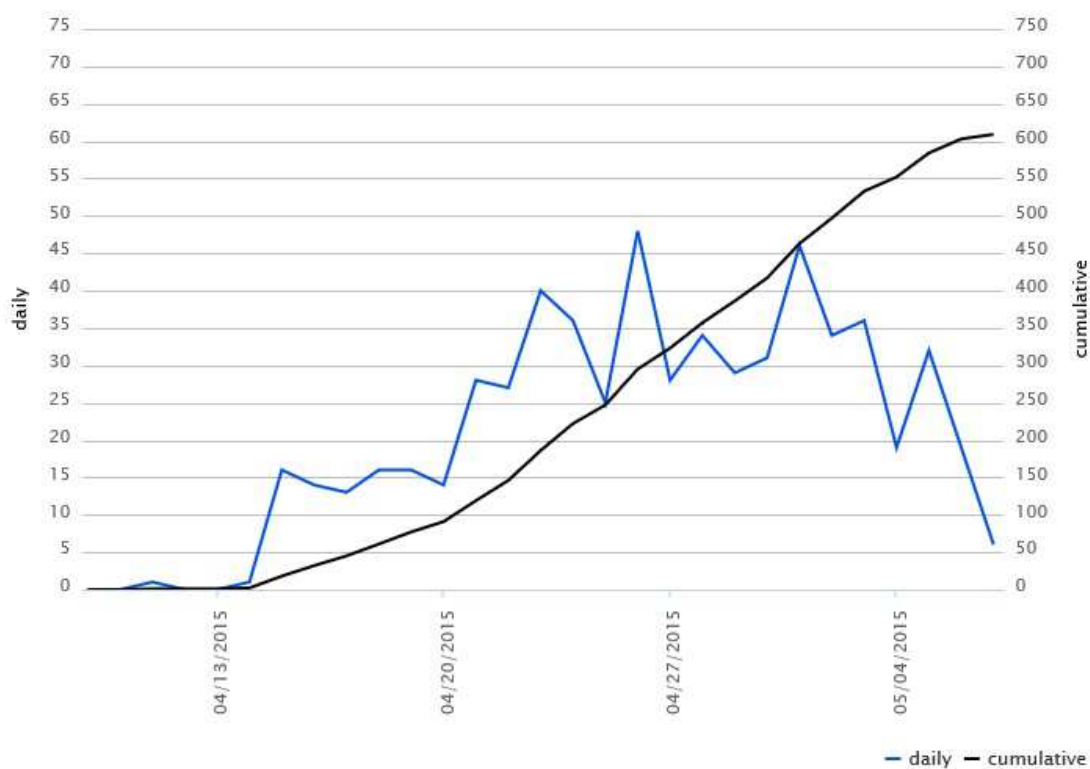
10.3.5 Srovnání

Je vidět, že žádná aplikace zobrazující stejná data přímo v mapě neexistuje. Aplikace je svým způsobem jedinečná a přináší na trh něco nového. Avšak není tak všestranná jako většina oblíbených aplikací zobrazujících počasí.

10.4 Oblíbenost aplikace na Windows Store

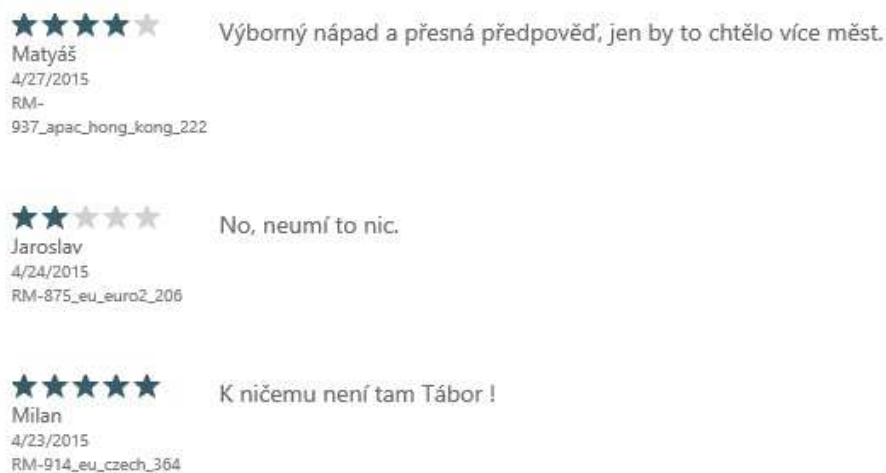
Aplikace byla necelé dva měsíce volně dostupná na Windows Store. Za tu dobu si ji stáhlo 609 uživatelů. Viz graf poskytovaný vývojářům přímo ve vývojářském portálu Windows Store, zobrazen na Obr. 32.

App downloads



Obrázek 32 - Počet stažení aplikace PočasíCZ

Za dobu kdy bylo PočasíCZ volně dostupné také aplikace obdržela tři hodnocení (viz Obr. 33).



Obrázek 33 - Hodnocení aplikace PočasíCZ

10.5 Hodnocení multiplatformnosti

Jedním ze základních požadavků na aplikaci bylo, aby byla multiplatformní, resp. aby se snadno daly přidávat další platformy. Tento požadavek byl splněn. Byla vytvořena

serverová část a jedna klientská část. Nyní již by se mohly přidávat další klientské části na dalších platformách, které by se připojovaly ke stejným webovým službám. Ty by již nebylo potřeba nijak upravovat.

Dala by se snadno přidat také platforma web – například přidáním služby Microsoft Azure Web Apps (popsána v kap. 3.2.1), která se dá snadno připojit k MAMS (ve které je napsána serverová část aplikace).

11. Technická omezení a systémové požadavky

Aplikace běží na jakémkoli telefonu s OS Windows Phone 8.1.

Podporované displeje jsou:

- WVGA (různé varianty rozlišení od 360 x 640 po 480 x 854)
- WXGA (různé varianty rozlišení od 768 x 1152 přes 720 x 1280 až po 768 x 1366)
- 720p (rozlišení 720 x 1280)

K nainstalování aplikace je potřeba stáhnout instalační package, který má 916 KB. Nainstalovaná aplikace zabírá 2,77 MB.

12. Závěr

Byla naprogramována serverová část aplikace za použití služby Microsoft Azure Mobile Services. Ta se připojí na stránku s předpovědí počasí, odkud naparsuje data předpovědi a ty uloží do databáze.

Dále byla naprogramována klientská část aplikace ve Windows Phone 8.1. Ta se připojí k serverové části pomocí webových služeb založených na REST, stáhne potřebná data a zobrazí je v mapě. Uživatel tedy může vidět teplotu a typ počasí (slunečno, zataženo, ...) „vznášející“ se v mapě přímo nad městem.

Původní představa o práci předpokládala, že formát KML (tj. formát souboru, který uchovává geografické informace) půjde snadno zobrazit v mapě, např. otevřením v jiné aplikaci nebo použitím určité knihovny. Zobrazení KML souboru se však na platformě Windows Phone 8.1 ukázalo jako celkem složitá věc. Navíc trend velkých softwarových společností posledních let je upouštět od podpory KML. Výsledná aplikace tedy formát KML nepoužívá. Místo toho generuje grafiku do mapy objekt po objektu.

Všechny cíle práce byly splněny a k nim ještě jeden navíc, který nebyl zadán: Aplikace byla publikována na Windows Store. I přesto, že byla výsledná aplikace původně uvažována spíše jako technologické demo, nakonec se na trhu Windows Store do určité míry prosadila. Je vidět, že možnost zobrazit počasí přímo v mapě přilákala několik uživatelů. Získala za necelé dva měsíce 609 stažení a 3 hodnocení s průměrným hodnocením 3,67 z 5. Samozřejmě se nemůže rovnat s neznámějšími aplikacemi na předpověď počasí, ale na poměry omezených časových možností diplomové práce to lze vidět jako úspěch.

Přehled zkratek a pojmů

.NET	Platforma pro vývoj a běh aplikací.
API	Application programming interface. Označuje rozhraní pro programování aplikací. Jde o sbírku procedur, funkcí, tříd či protokolů nějaké knihovny (ale třeba i jiného programu nebo jádra operačního systému), které může programátor využívat.
C++	Programovací jazyk.
C#	Programovací jazyk.
Code-behind	Vývojový model používaný v .NETu, u kterého každá vizuální třída (tzn. třída definující GUI) má přiřazenu svoji třídu s kódem, která zpracovává události vyvolané v GUI a další logiku aplikace. Tyto přiřazené třídy se taktéž označují jako code-behind.
CSS	Cascading Style Sheets. Kaskádové styly. Jazyk pro popis způsobu zobrazení elementů na stránkách napsaných v HTML.
CSV	Comma separated values. Typ datového souboru, kde na každé řádce je jeden záznam a jednotlivé hodnoty jednoho záznamu jsou odděleny čárkou.
Debugování	Proces vývoje softwaru při kterém může vývojář krokovat jednotlivé příkazy kódu a sledovat jak probíhají.
DXF	Formát souboru používaný nástroji na návrh designu, jako například SketchUp nebo AutoCAD.
GPS	Global Positioning System. Globální polohovací systém je systém, s jehož pomocí je možno určit polohu a přesný čas kdekoliv na Zemi nebo nad Zemí s přesností do deseti metrů.
GPS souřadnice	Zeměpisná šířka a zeměpisná délka (a někdy i výška) určitého bodu na povrchu Země (nebo v jeho blízkosti).
GUI	Graphical User Interface. Grafické uživatelské rozhraní.
HTML	HyperText Markup Language. Značkovací jazyk používaný pro tvorbu webových stránek.
HTML5	Novější verze HTML. Viz pojem HTML.
IaaS	Infrastructure as a Service. Infrastruktura jako služba znamená používání vlastní virtuální infrastruktury na fyzické infrastruktuře poskytovatele.
IDE	Integrated Development Environment. Vývojové prostředí.
Javascript	Multiplatformní programovací jazyk. Nejčastěji se používá pro programování interaktivity webových stránek.
KML	Keyhole Markup Language. Typ souboru pro zobrazení geografických dat.
LineString	Termín ze specifikace KML. Jedná se o lomenou čáru definovanou sadou bodů.

Load balancing	Technika vyvažování zátěže přesměrováním požadavků na různé servery, které jsou schopny vyřídit požadavek uživatele.
MAMS	Microsoft Azure Mobile Services. Jedna ze služeb nabízená v Microsoft Azure. Slouží pro komunikaci s mobilními zařízeními.
Management Portal	Uživatelské rozhraní ke službám Microsoft Azure.
Open-source	Software s otevřeným zdrojovým kódem. Uživatelé mohou zdrojový kód prohlížet a za určitých podmínek i upravovat.
ORM	Object-relational mapping. Objektově relační zobrazení. Technika která zajišťuje automatickou konverzi dat mezi relační databází (SQL) a objektově orientovaným programovacím jazykem.
OS	Operační systém.
PaaS	Platform as a Service. Platforma jako služba. Jedná se o outsourcing služeb potřebných pro provoz aplikace, ale ne samotné aplikace.
Repozitář	Soubory a složky zdrojového kódu uložené ve stromové struktuře.
Responzivní	Responzivní aplikace je taková aplikace, která se nezasekává, ale na každou událost, kterou uživatel vyvolá, je jasně a rychle zobrazena reakce.
Ruby	Programovací jazyk.
SaaS	Software as a Service. Software jako služba. Představuje pronájem celé aplikace zákazníkem.
SOA	Servisně orientovaná architektura. Architektura založená na využívání webových služeb.
Soubor DXF	Formát souboru používaný nástroji na návrh designu, jako například SketchUp nebo AutoCAD.
SQL	Structured Query Language. Jazyk určený k práci s relační databází.
UNIX	UNIX je ochranná známka operačního systému vytvořeného v Bellových laboratořích americké firmy AT&T v roce 1969. Existují různé systémy, které jsou UNIXem inspirovány. Takové systémy se souhrnně označují jako unixové systémy (angl. unix-like).
Virtualizace	Provoz 1 až n virtuálních serverů na jednom nebo více fyzických serverech.
Visual Studio	Nejrozšířenější IDE pro vývoj aplikací pro Windows a Windows Phone
XML	Extensible Markup Language. Známý značkovací jazyk. Příklad: <dokument><tělo /></dokument>

Literatura

- [1] ZIKMUND, Martin. Co je to Cloud computing a proč se o něm mluví. In: *Business Vize* [online]. 2010 [cit. 2015-04-24]. Dostupné z: <http://www.businessvize.cz/software/co-je-to-cloud-computing-a-proc-se-o-nem-mluvi>
- [2] MICROSOFT. *Azure Technical Documentation Library* [online]. 2015, 2015-03-18 [cit. 2015-04-24]. Dostupné z: <https://msdn.microsoft.com/library/azure/>
- [3] MATYS, Milan. *Cloud computing na platformě Windows Azure*. Pardubice, 2012. Diplomová práce. Univerzita Pardubice.
- [4] Platform. 2015. *FOLDOC - Computing Dictionary* [online]. [cit. 2015-05-12]. Dostupné z: <http://foldoc.org/platform>
- [5] Cross platform Definition. 2015. *PC Magazine Encyclopedia* [online]. [cit. 2015-05-12]. Dostupné z: <http://www.pcmag.com/encyclopedia/term/40495/cross-platform#fbid=aHfb3ldkqPq>
- [6] BALDWIN, Chelsea. Cross-Platform Mobile Development Options. In: *Build Blog by ThinkApps* [online]. 2015 [cit. 2015-04-28]. Dostupné z: <http://thinkapps.com/blog/development/develop-for-ios-v-android-cross-platform-tools/>
- [7] Multiplatformní .NET aplikace Inventoroid vytvořená pomocí Xamarinu. In: *Czech MSDN Blog* [online]. 2014 [cit. 2015-04-28]. Dostupné z: <http://blogs.msdn.com/b/vyvojari/archive/2014/06/09/multiplatformni-aplikace-inventoroid-pomoci-xamarinu.aspx>
- [8] IOS / Android cross platform development. In: *Stack Overflow* [online]. 2015 [cit. 2015-04-28]. Dostupné z: <http://stackoverflow.com/questions/6171128/ios-android-cross-platform-development>
- [9] Web Services Glossary. 2015. *World Wide Web Consortium (W3C)* [online]. [cit. 2015-05-12]. Dostupné z: <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>
- [10] RICHARDSON, Leonard a Sam RUBY. 2007. *RESTful web services*. 1st ed. Sebastopol: O'Reilly, xxiv, 419 s. ISBN 978-0-596-52926-0.
- [11] Web Services Activity. 2015. *World Wide Web Consortium (W3C)* [online]. [cit. 2015-05-12]. Dostupné z: <http://www.w3.org/2002/ws/>
- [12] MUELLER, John. Understanding SOAP and REST Basics. In: *Software Quality Matters Blog* [online]. 2013-01-08 [cit. 2015-04-30]. Dostupné z: <http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/>
- [13] LACKO, Ľuboslav. *Vývoj aplikací pro Windows 8.1 a Windows Phone*. 1. vyd. Brno: Computer Press, 2014, 328 s. ISBN 978-80-251-3822-9.
- [14] KML | OGC. 2015. *Open Geospatial Consortium* [online]. [cit. 2015-05-12]. Dostupné z: <http://www.opengeospatial.org/standards/kml/>

- [15] GOOGLE, INC. 2015. *Keyhole Markup Language* [online]. [cit. 2015-05-12]. Dostupné z: <https://developers.google.com/kml/documentation/?csw=1>
- [16] GOOGLE, Inc. *Google Earth API (Deprecated)* [online]. 2014, 2014-12-08 [cit. 2015-04-30]. Dostupné z: <https://developers.google.com/earth/>
- [17] ONWUMERE, Vincent Ojinnaka. How to View KMZ or KML File on Mobile Devices. In: *Environment in View* [online]. 2013-09-18 [cit. 2015-04-30]. Dostupné z: <http://environmentinview.blogspot.cz/2013/09/how-to-view-kmz-or-kml-file-on-mobile.html>
- [18] How to view web based KML files on your mobile device. In: MARKIE, Aaron. *MTB-TV - Mountain Bike Trail Videos* [online]. 2015 [cit. 2015-04-30]. Dostupné z: <http://markie.com.au/mtb/blog.asp?BlogID=36&BlogName=How-to-view-web-based-KML-files-on-your-mobile-device>
- [19] BRUNDRITT, Ricky. *Location Intelligence for Windows Store Apps* [online]. 2014 [cit. 2015-05-01]. ISBN 978-1-291-76109-2. Dostupné z: <https://rbrundritt.wordpress.com/my-book/>
- [20] Bing Maps AJAX Control 7.0 Supported Browsers. In: *MSDN Library* [online]. 2015 [cit. 2015-05-02]. Dostupné z: <https://msdn.microsoft.com/en-us/library/gg427618.aspx>
- [21] How to display maps in the Map control. In: *MSDN Library* [online]. 2015 [cit. 2015-05-02]. Dostupné z: <https://msdn.microsoft.com/en-us/library/windows/apps/xaml/dn642089.aspx>
- [22] Review: Microsoft Azure beats Amazon and Google for mobile development. 2015. *InfoWorld* [online]. [cit. 2015-05-11]. Dostupné z: <http://www.infoworld.com/article/2890167/application-development/review-microsoft-azure-beats-amazon-and-google-for-mobile-development.html>
- [23] MICROSOFT. *Microsoft Azure: Cloud Computing Platform & Services* [online]. 2015 [cit. 2015-05-04]. Dostupné z: <http://azure.microsoft.com/en-us/>
- [24] Deployment Patterns. 2015. *MSDN - Microsoft Developer Network* [online]. [cit. 2015-05-11]. Dostupné z: <https://msdn.microsoft.com/en-us/library/ms998478.aspx?f=255&MSPPErr=-2147217396>
- [25] TEJASWI REDKAR, Tony Guidici. *Windows Azure platform*. 2nd ed. Berkeley, CA: Apress, 2011. ISBN 978-143-0235-644.
- [26] *Google Maps* [online]. 2015. [cit. 2015-05-12]. Dostupné z: <https://www.google.cz/maps>
- [27] *Tool for Google Maps v3 (version 3)* [online]. 2014. [cit. 2015-05-12]. Dostupné z: <http://www.birdtheme.org/useful/v3tool.html>
- [28] *Free Map Tools* [online]. 2015. [cit. 2015-05-12]. Dostupné z: <http://www.freemaptools.com/kml-file-creator.htm>
- [29] *Earth Point* [online]. 2015. [cit. 2015-05-12]. Dostupné z: <http://www.earthpoint.us/ExcelToKml.aspx>
- [30] *KML Tools* [online]. 2014. [cit. 2015-05-12]. Dostupné z: <http://kmltools.appspot.com/>

- [31] *Mapmash / KML Creator* [online]. 2015. [cit. 2015-05-12]. Dostupné z: <http://www.mapmash.in/kmlpoint.html>
- [32] *Spreadsheet Mapper 3* [online]. 2015. [cit. 2015-05-12]. Dostupné z: <http://www.google.cz/earth/outreach/tutorials/spreadsheet3.html>
- [33] *Google Earth Outreach* [online]. 2015. [cit. 2015-05-12]. Dostupné z: <http://www.google.com/earth/>

Příloha A – Uživatelská příručka

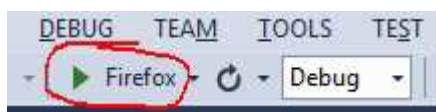
Postup spuštění aplikace na lokálním stroji

Aplikaci PočasíCZ si může každý vývojář spustit na svém stroji. K tomu je zapotřebí:

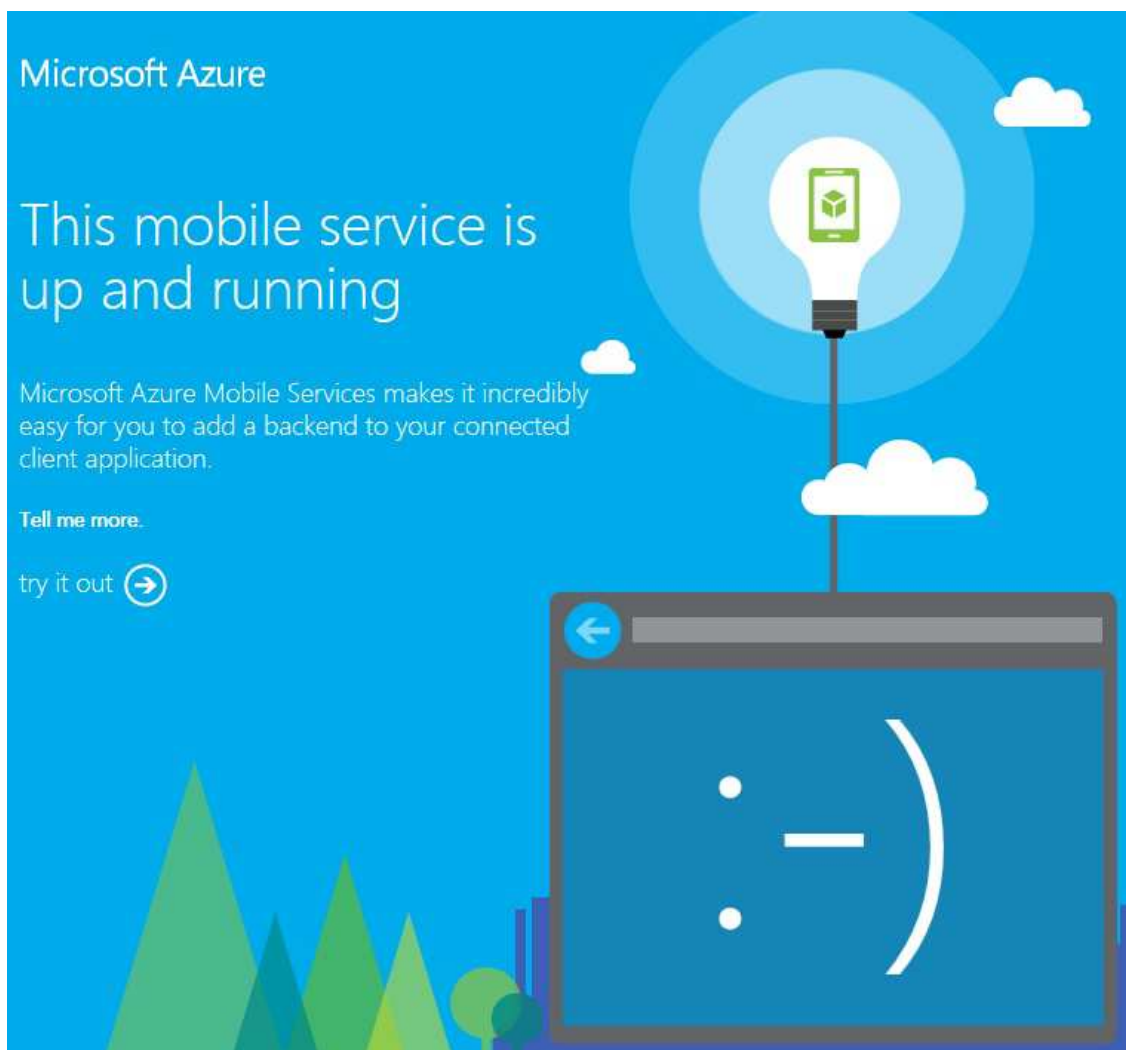
- Operační systém Windows 8.1
- Visual Studio 2013

Postup spuštění aplikace:

1. Vložte instalační DVD do vašeho počítače.



Obrázek 34 - Spuštění projektu počasiczService



Obrázek 35 - Spuštění webových služeb

2. Na instalačním DVD naleznete dvě složky: `src` a `doc`. Ve složce `src` je složka `pocasicz`, kterou zkopírujete na disk vašeho počítače.
3. Spusťte Visual Studio 2013.
4. Otevřete soubor `pocasicz.sln`, který naleznete ve složce `pocasicz`, kterou jste v kroku 2. zkopírovali do vašeho počítače.

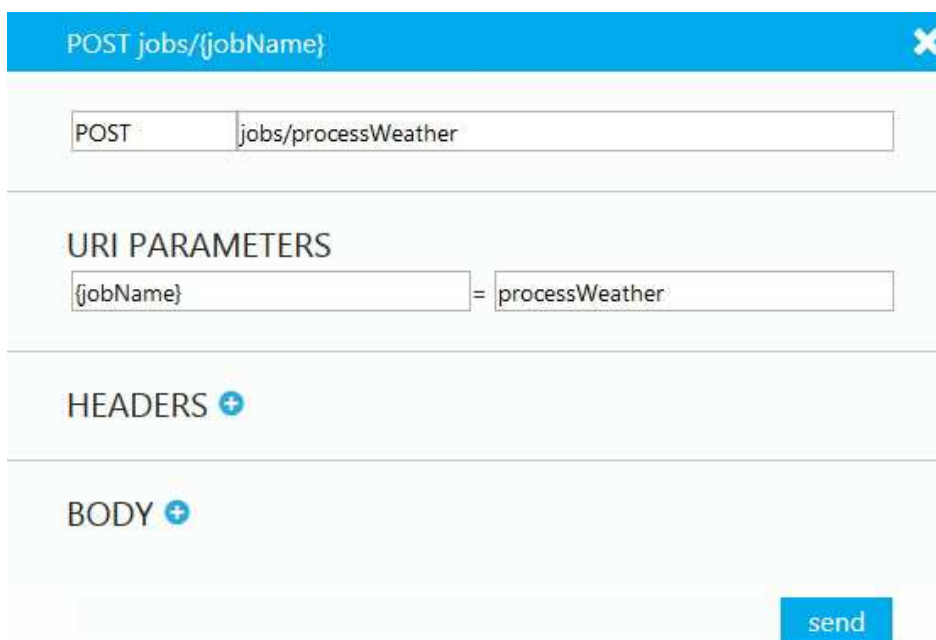
Endpoint	Method
Jobs	POST jobs/{jobName}
Town	GET tables/Town
	GET tables/Town/{id}
	PATCH tables/Town/{id}
	POST tables/Town
	DELETE tables/Town/{id}
WeatherItem	GET tables/WeatherItem
	GET tables/WeatherItem/{id}
	PATCH tables/WeatherItem/{id}
	POST tables/WeatherItem
	DELETE tables/WeatherItem/{id}

Obrázek 36 - API Documentation

Endpoint	Method	Action
POST jobs/{jobName}		try this out
Request Information		
jobName	No documentation available.	Define this parameter in the request URI.

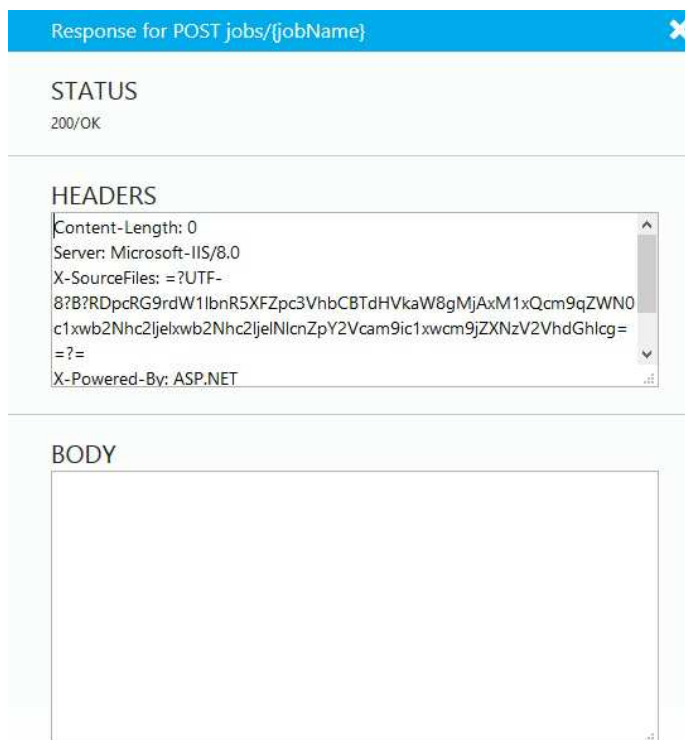
Obrázek 37 - POST jobs

5. Visual Studio otevře solution sestávající ze dvou projektů: pocasicz.WindowsPhone a pocasiczService.
6. Pravým tlačítkem na projekt pocasiczService -> Set as StartUp Project.
7. Spustíte projekt v jednom z vašich internetových prohlížečů, viz Obr. 34.
8. Otevře se vám stránka, viz Obr. 35. Klikněte na šipku *try it out*.
9. Otevře se vám stránka, viz Obr. 36. Klikněte na POST jobs/{jobName}.
10. Otevře se vám stránka, viz Obr. 37. Klikněte na *try this out*.
11. V otevřeném okně vyplňte text processWeather do pole za znakem ‚=‘, tak jak je to vidět na Obr. 38. Klikněte na tlačítko send.
12. Zobrazí se okno potvrzující úspěšné spuštění jobu, viz Obr. 39.
13. Počkejte 30 sekund na dobehnutí jobu.
14. Ve Visual Studiu zastavte běh aktuálního projektu.
15. Pravým tlačítkem na projekt pocasicz.WindowsPhone -> Set as StartUp Project.
16. Spustíte projekt v emulátoru, viz Obr. 40.

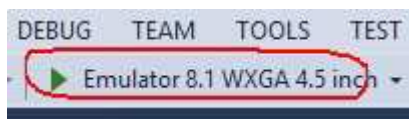


The image shows a REST client interface with a blue header bar containing the text "POST jobs/{jobName}" and a close button (X). Below the header, there is a text input field containing "POST" and another containing "jobs/processWeather". Underneath, there is a section titled "URI PARAMETERS" with a table-like structure: a box containing "{jobName}" followed by an equals sign and a box containing "processWeather". Below this are sections for "HEADERS" and "BODY", each with a plus sign icon. At the bottom right, there is a blue button labeled "send".

Obrázek 38 - Spuštění jobu processWeather



Obrázek 39 - Oznamení o úspěšném proběhnutí jobu



Obrázek 40 - Spuštění emulátoru

Instalace aplikace z Windows Store

Uživatelé mobilních zařízení si mohou zdarma nainstalovat aplikaci PočasíCZ. Mobilní zařízení musí mít následující parametry:

- Operační systém Windows Phone 8.1
- Jeden z následujících podporovaných displejů:
 - WVGA (různé varianty rozlišení od 360 x 640 po 480 x 854)
 - WXGA (různé varianty rozlišení od 768 x 1152 přes 720 x 1280 až po 768 x 1366)
 - 720p (rozlišení 720 x 1280)
- 3 MB volné paměti

Postup instalace:

1. Spustíte aplikaci Store.
2. Dejte vyhledat „PočasíCZ“.
3. Vyhledanou aplikaci nainstalujte.

Uživatelská příručka pro aplikaci PočasíCZ

Toto je uživatelská příručka pro uživatele, kteří chtějí používat PočasíCZ – aplikaci, která umožňuje zobrazovat počasí v České republice přímo v mapě. Přesná předpověď až na 48 hodin dopředu.

Aplikace funguje jen při zapnutém internetovém spojení. Pokud nastane výpadek vašeho internetového spojení, aplikace vás na tuto skutečnost upozorní zobrazením okna, viz Obr. 41.



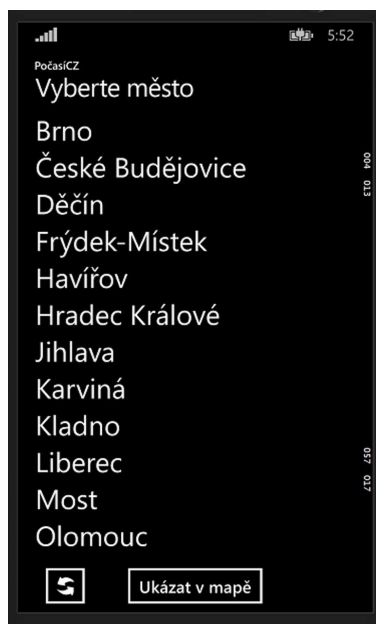
Obrázek 41 - Chyba při stahování dat

Při spuštění aplikace se vám zobrazí první stránka, na které je možné vybrat si město, viz Obr. 42. Pro zobrazení počasí ve vybraném městě ťukněte na vybrané město. Pro obnovení obsahu stránky ťukněte na ikonu obnovení v levém dolním rohu stránky. Pro zobrazení počasí celé České republiky v mapě klikněte na tlačítko *Ukázat v mapě*.

Pokud jste na první stránce ťukli na některé z měst, zobrazí se vám druhá stránka, na které jsou zobrazeny teploty ve městě na příštích až 48 hodin, viz Obr. 43. Pokud si přejete při příštím spuštění aplikace přejít rovnou na tuto stránku, zatrhněte možnost ‚Vždy zobrazovat toto město‘. Pro obnovení obsahu stránky ťukněte na ikonu obnovení v levém dolním rohu stránky. Pro zobrazení počasí v oblasti tohoto města v mapě klikněte na tlačítko *Ukázat v mapě*.

Pokud jste na první nebo druhé stránce ťukli na tlačítko *Ukázat v mapě*, zobrazí se vám třetí stránka, na které se zobrazuje počasí přímo v mapě, viz Obr. 44. V pravém dolním rohu se zobrazuje čas, pro který je zobrazeno počasí. Posuvníkem v dolní části obrazovky můžete měnit tento čas, počasí se bude měnit v závislosti na něm. Pro obnovení obsahu stránky ťukněte na ikonu obnovení v levém dolním rohu stránky. Mapu můžete

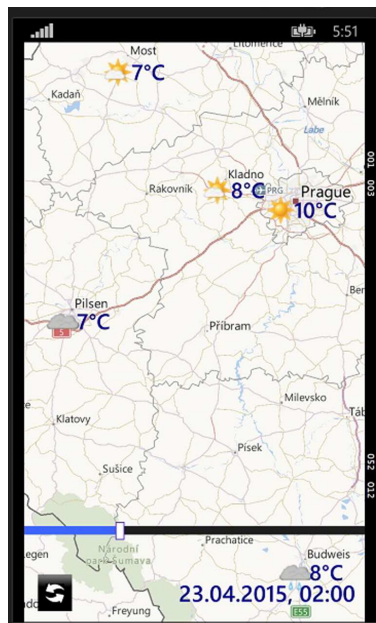
posouvat, zmenšovat a zvětšovat podle libosti. Počasí je zobrazeno pro 20 největších měst České republiky.



Obrázek 42 - Stránka s výběrem měst



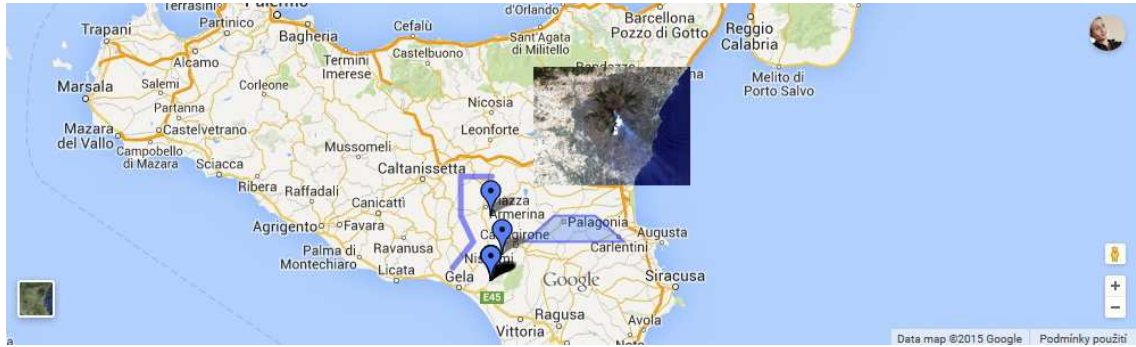
Obrázek 43 - Počasí ve městě



Obrázek 44 - Počasí v mapě

Příloha B – Zobrazení KML ve webovém prohlížeči

Tato příloha ukazuje zobrazení jednoho ukázkového souboru v různých webových pluginech.



Obrázek 45 - Ukázkový KML soubor v pluginu Google Maps



Obrázek 46 - Ukázkový KML soubor v pluginu Google Earth



Obrázek 47 - Ukázkový KML soubor v pluginu Bing Maps, AJAX verze 6.3

Příloha C - Konfigurační soubor načítaný jobem

Zde je aktuální obsah konfiguračního souboru, který obsahuje seznam měst, pro který se mají načíst data z webu, a jejich GPS souřadnice.

Name,Url,Longitude,Latitude

Pra-

ha,http://www.yr.no/place/Czech_Republic/Prague/Prague/hour_by_hour.html,14.4211,50.0872

Br-

no,http://www.yr.no/place/Czech_Republic/South_Moravia/Brno/hour_by_hour.html,16.616667,49.2

Ostrava,http://www.yr.no/place/Czech_Republic/Moravia-Silesia/Ostrava/hour_by_hour.html,18.292778,49.835556

Pl-

zeň,http://www.yr.no/place/Czech_Republic/Plze%C5%88/Plze%C5%88/hour_by_hour.html,13.3775,49.746944

Libe-

rec,http://www.yr.no/place/Czech_Republic/Liberec/Liberec/hour_by_hour.html,15.04693,50.76365

Olo-

mouc,http://www.yr.no/place/Czech_Republic/Olomouc/Olomouc/hour_by_hour.html,17.250833,49.593889

Ústí nad

Labem,http://www.yr.no/place/Czech_Republic/%C3%9Ast%C3%AD_nad_Labem/%C3%9Ast%C3%AD_nad_Labem/hour_by_hour.html,14.041667,50.658333

České

Budějovice,http://www.yr.no/place/Czech_Republic/South_Bohemia/Budweis/hour_by_hour.html,14.474722,48.974722

Hradec

Králové,http://www.yr.no/place/Czech_Republic/Hradec_Kr%C3%A1lov%C3%A9/Hradec_Kr%C3%A1lov%C3%A9/hour_by_hour.html,15.8375,50.211111

Pardubice,http://www.yr.no/place/Czech_Republic/Pardubice/Pardubice/hour_by_hour.html,15.77916,50.03861

Havířov,http://www.yr.no/place/Czech_Republic/Moravia-

Silesia/Hav%C3%AD%C5%99ov/hour_by_hour.html,18.433333,49.8

Zlín,http://www.yr.no/place/Czech_Republic/Zl%C3%ADn/Zl%C3%ADn/hour_by_hour.html,17.666944,49.233056

Kladno,http://www.yr.no/place/Czech_Republic/Central_Bohemia/Kladno/hour_by_hour.html,14.102778,50.147222

Most,http://www.yr.no/place/Czech_Republic/%C3%9Ast%C3%AD_nad_Labem/Most/hour_by_hour.html,13.636667,50.503056

Opava,http://www.yr.no/place/Czech_Republic/Moravia-

Silesia/Opava/hour_by_hour.html,17.9,49.933333

Frýdek-Místek,http://www.yr.no/place/Czech_Republic/Moravia-

Silesia/Fr%C3%BDdek-M%C3%ADstek/hour_by_hour.html,18.347778,49.688056

Karviná,http://www.yr.no/place/Czech_Republic/Moravia-

Silesia/Karvin%C3%A1/hour_by_hour.html,18.533333,49.85

Jihlava,http://www.yr.no/place/Czech_Republic/Vyso%C4%8Dina/Jihlava/hour_by_hour.html,15.590278,49.396389

Děčín,http://www.yr.no/place/Czech_Republic/%C3%9Ast%C3%AD_nad_Labem/D%C4%9B%C4%8D%C3%ADn/hour_by_hour.html,14.196111,50.773611

Teplice, http://www.yr.no/place/Czech_Republic/%C3%9Ast%C3%AD_nad_Labem/Teplice/hour_by_hour.html, 13.816667, 50.633333