

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

---

# **Komunitní geografický portál v Adobe Air**

## **Geographic community portal using Adobe Air**

2010

Bc. Michal Prílepok

Súhlasím so zverejnením tejto diplomovej práce podľa požiadavky čl. 26, odst. 9 *Studijného a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava*.

V Ostrave 7. mája 2010

.....

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave 7. mája 2010

.....

Ďakujem vedúcemu diplomovej práce Ing. Janu Platošovi za poskytnutú podporu, cenné informácie a trpezlivosť pri mojich otázkach. Vďaka patrí aj tým, ktorí ma podporovali pri tejto práci ale aj počas celého môjho vysokoškolského štúdia.

## **Abstrakt**

Diplomová práca prináša popis technológie Adobe AIR, popis základných API funkcií pre tvorbu užívateľského rozhrania, prístup k lokálnemu súborovému systému, komunikáciu po sieti, prácu s médiami. V práci sa nachádza aj popis nástrojov na ladenie a zostavenie inštalačného balíčka. Možnosti technológie Adobe AIR sú ukázané na jednoduchom geografickom portáli. Na tomto portáli sa nachádzajú body záujmu, GPS trasy a fotografie usporiadané v kategóriách.

**Kľúčové slová:** Adobe AIR, framework, HTML, CSS, JavaScript, Ajax

## **Abstract**

The thesis provides a description of the Adobe AIR technology, a description of the basic API functions for creating user interface, accessing the local file system, network communication and working with the media. The work also includes a description of the tools for debugging and building the installation package. Adobe AIR technology options are shown/ demonstrated on the simple geographic portal. This portal contains points of interest, GPS tracks and photos organized into categories.

**Keywords:** Adobe AIR, framework, HTML, CSS, JavaScript, Ajax

## Zoznam použitých skratiek a symbolov

AIR	– Adobe Integrated Runtime
DRM	– Digital rights management - správa digitálnych práv
ELS	– Encrypted local store - šifrované uložisko
GPS	– Globálny Pozičný Systém
HTML	– Hyper Text Markup Language
POI	– Point of interest - bod záujmu
RIA	– Rich Internet applications
SWF	– Small Web Format - formát súboru pre Adobe Flash

## Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Adobe AIR</b>	<b>6</b>
2.1	Porovnanie technológií . . . . .	6
2.2	Výhody Adobe AIR . . . . .	7
2.3	Vlastnosti AIR aplikácie . . . . .	8
<b>3</b>	<b>API Adobe AIR</b>	<b>10</b>
3.1	Natívne okná a menu aplikácie a panel úloh . . . . .	10
3.2	Súbory a dáta . . . . .	14
3.3	Média s bohatým obsahom . . . . .	19
3.4	Interakcia s operačným systémom . . . . .	25
3.5	Siete a komunikácia . . . . .	26
3.6	Distribúcia a aktualizácia aplikácie . . . . .	33
3.7	Lokalizácia názvu, popisu aplikácie a inštalátora . . . . .	35
<b>4</b>	<b>Vývojové nástroje</b>	<b>37</b>
4.1	AIR Debug Launcher (ADL) . . . . .	37
4.2	AIR Developer Tool (ADT) . . . . .	38
4.3	Vytvorenie a podpísanie inštalačného balíčka . . . . .	38
4.4	Ladenie aplikácie pomocou AIR HTML Introspector . . . . .	39
4.5	Vytvorenie HTML založenej aplikácie . . . . .	41
4.6	Vytvorenie certifikátu . . . . .	43
4.7	Vytvorenie inštalačného súboru . . . . .	44
<b>5</b>	<b>Popis portálu a klientskej aplikácie</b>	<b>45</b>
5.1	Klientska aplikácia . . . . .	45
5.2	Serverová časť . . . . .	53
5.3	Administrátorské rozhranie . . . . .	59
<b>6</b>	<b>Zaver</b>	<b>61</b>
<b>7</b>	<b>Literatúra</b>	<b>62</b>
	<b>Prílohy</b>	<b>62</b>
<b>A</b>	<b>DVD-ROM</b>	<b>63</b>

---

## Zoznam tabuliek

1	Vlastnosti AIR aplikácie . . . . .	9
2	Vzhľad natívneho okna v rôznych operačných systémoch . . . . .	11
3	Vlastnosti okna . . . . .	11
4	Triedy Adobe AIR menu . . . . .	13
5	Triedy určené pre prácu s lokálnym súborovým systémom . . . . .	14
6	Konštanty objektu pdfCapability . . . . .	20
7	Triedy pre prácu so zvukom . . . . .	22
8	Udalosti triedy NetStream . . . . .	23
9	Metódy triedy NetStream . . . . .	24
10	Metódy objektu NativeApplication . . . . .	25
11	Vlastnosti triedy URLRequest - bezpečné prostredie . . . . .	29
12	Vlastnosti triedy URLRequest . . . . .	29
13	ADL návratové hodnoty . . . . .	38
14	Vstupné parametre ladiacich metód . . . . .	40
15	Ikony ladiacich metód . . . . .	40
16	Menu aplikácie . . . . .	47
17	Databázové tabuľky . . . . .	55

---

## Zoznam obrázkov

1	Neštandardný vzhľad AIR aplikácie . . . . .	10
2	AIR HTML Introspector . . . . .	39
3	poiAir klientska aplikácia . . . . .	46
4	Zoznam kategórií POI . . . . .	48
5	Zoznam POI v kategórií . . . . .	48
6	Detail POI . . . . .	49
7	Formulár pre pridanie POI . . . . .	50
8	Kategorie GPS trás . . . . .	51
9	Zobrazenie fotoalbumu . . . . .	52
10	Zobrazenie profilu používateľa . . . . .	53
11	Zobrazenie úvodnej stránky portálu . . . . .	54



## Zoznam výpisov zdrojového kódu

1	Vytvorenie nového natívneho okna . . . . .	12
2	Nastavenie poslucháča na udalosť okna . . . . .	12
3	Vytvorenie nového databázového súboru . . . . .	16
4	Vytvorenie databázy v synchrónnom režime . . . . .	17
5	Posielanie SQL príkazu . . . . .	17
6	Ošetrovanie výniekov SQL . . . . .	18
7	Uloženie objektu do ELS . . . . .	18
8	Prečítanie objektu z ELS . . . . .	19
9	Zmazanie objektu z ELS . . . . .	19
10	Zmazanie ELS . . . . .	19
11	Zistenie podpory pre PDF obsahu . . . . .	20
12	Zobrazenie PDF obsah . . . . .	20
13	Zistenie obmedzení aplikácie . . . . .	25
14	Sledovanie prítomnosti užívateľa . . . . .	26
15	Sledovanie zmien v sieti . . . . .	27
16	Zistenie dostupnosti http zdroja . . . . .	27
17	Nastavenie prihlasovacích údajov . . . . .	30
18	Poslanie XML dát na server . . . . .	31
19	Získanie dát z externého zdroja . . . . .	31
20	Vytvorenie viacerých premenných objektu URLVariables . . . . .	32
21	Poslanie http požiadavky s parametrami . . . . .	32
22	Inštalácia AIR aplikácie z internetového prehliadača . . . . .	35
23	Spustenie AIR aplikácie z internetového prehliadača . . . . .	35
24	Lokalizácia názvu aplikácie . . . . .	36
25	Lokalizácia dátumu . . . . .	36
26	Popisový súbor AIR aplikácie . . . . .	41
27	Popisový súbor AIR aplikácie HelloWorld . . . . .	42
28	Štruktúra HTML dokumentu . . . . .	42
29	HelloWorld.html . . . . .	42
30	Zoznam kategórií POI . . . . .	55
31	Výpis zoznamu POI pre zvolenú kategóriu . . . . .	56
32	Detail POI . . . . .	56
33	Výpis zoznamu kategórií GPS trás . . . . .	56
34	Výpis zoznamu GPS trás v kategórií . . . . .	57
35	Zoznam fotoalbumov . . . . .	58

## 1 Úvod

Cieľom tejto diplomovej práce tejto práce je popísať technológiu Adobe AIR, jej možnosti a vlastnosti, vytvoriť ukážkovú aplikáciu pre geografický informačný portál. Ukážková aplikácia sa bude skladať z dvoch častí. Prvá časť je umiestnená na serveri.

Serverová časť je vytvorená pomocou technológie ASP .Net v programovacom jazyku C#. Dáta sú uložené v databáze. Serverová časť bude slúžiť na sprostredkovanie dát medzi databázou a klientskou aplikáciou, spravovať obsah portálu pomocou administratívneho rozhrania. Výstup serverovej časti je textovo formátovaný.

Klientska aplikácia je vytvorená technológiami určenými na tvorbu internetových stránok HTML, CSS a JavaScript (AJAX). Bude spúšťaná na frameworku Adobe AIR ako samostatná desktopová aplikácia. So serverovou časťou bude komunikovať zasielaním HTTP požiadaviek POST. Aplikácia prijaté dáta vhodne naformátuje a zobrazí. Obsah je zobrazovaný pomocou technológie JavaScript, ktorá vkladá do existujúceho HTML dokumentu nové tagy.

Na portáli budú ukladané body záujmu (tzv. POI), GPS trasy a fotografie. Body záujmu a GPS trasy budú usporiadané v kategóriách. Kategórie bodov záujmu budú obohatené o ikonu. Pridávané fotografie sa budú zaraďovať do fotoalbumov.

Body záujmu bude možné vyexportovať z aplikácie vo formáte, ktorý umožňuje ich ďalšie spracovanie v ľubovoľnom softvéri.

## 2 Adobe AIR

Adobe AIR je runtime modul pre rôzne operačné systémy, ktorý vývojárom umožňuje používať známe webové technológie *HTML*, *Ajax*, *Adobe Flash* alebo *Adobe Flex*, k vytvoreniu internetových aplikácií určených pre stolovné počítače. Pomocou programu Adobe AIR môžu vývojári využívať svoje doterajšie schopnosti k vytváraniu pútavých, vizuálne bohatých aplikácií (RIA aplikácie), ktoré spájajú výkon miestnych prostriedkov s obsahom webu. Adobe AIR aplikácie je možné spúšťať na operačných systémoch Windows, Linux a Mac OS.<sup>1</sup>

**Rich Internet applications** (RIA) sú webové aplikácie, funkcionálne rovnaké ako klasické desktopové aplikácie známe pod názvom tučný klient. Základnými spoločnými znakmi týchto aplikácií je prenos a manipulácia dát na externom dátovom úložisku, napríklad na firemnom serveri a rovnako pohodlné užívateľské prostredie. V dnešnom svete internetu sa vývojárske spoločnosti snažia čo najviac uľahčiť prístup a používanie programov ich užívateľom. Najpohodlnejším riešením je sprístupniť program cez okno webového prehliadača. Základným rozdielom medzi tučným klientom a RIA je prístup a spustenie programu. Zatiaľ čo tučného klienta je potrebné inštalovať na užívateľský počítač, RIA je plne funkčne prístupná v akomkoľvek webovom prehliadači bez inštalácie, pričom užívateľské prostredie zostáva nezmenené.

RIA nepoužíva pre zmenu stavu programu obnovenie celej obrazovky, ako je to pri bežnej webovej stránke. Všetko sa deje tak ako pri klasickej desktopovej aplikácii. Hlavnou výhodou je plná flexibilita a nezávislosť na operačnom systéme.

Pre tvorbu RIA poznáme technológie:

- Adobe Flash
- Adobe Flex a Adobe AIR
- Java applet
- JavaScript / Ajax
- Microsoft Silverlight

### 2.1 Porovnanie technológií

#### 2.1.1 Adobe Flash

Flash bol formálne uvedený v roku 1996 spoločnosťou Macromedia. V súčasnosti technológiu Flash podporuje Adobe, ako platformu pre online interakciu s médiami - zdieľanie videa. Flash bol pôvodne vyvinutý pre médiá využívajúce vektorovo založené animácie, zatiaľ čo Shockwave bola platforma pre video a interaktivitu. Flash pohltil shockwave vo všetkých aspektoch používania, s výnimkou online 3D hier a interaktívnych školiacich

<sup>1</sup>Popis Adobe AIR je prevzatý z literatúry [1].

aplikácií, čo je primárne použitie shockwave. Flash je pomerne univerzálny, dokáže pracovať s videami, obrázkami, vektorovými animáciami, snímkovými animáciami pomocou skriptovacieho jazyka ActionScript a multiplatformnej podpore.

### 2.1.2 Adobe Flex

Flex je založený na platforme Flash, ale viac programátorský prístup k interaktívnemu návrhu webových aplikácií. Zatiaľ čo Flash je prispôbený k vývoju aplikácií založených na vektorových animáciách, ako bol pôvodne navrhnutý. Flex umožňuje programátorom rýchlo vyvíjať aplikácie a ich vzhľad pomocou *MXML* a *ActionScriptu* spolu s ďalšími technológiami pre prenos videa a zvuku. Flex je postavený na platforme Flash, a vytvára výstup vo formáte Flash. Flex aplikácie sú multiplatformové. Flex natívne používa komunikáciu server - klient pre získanie dát, ktorá umožňuje vytvoriť veľmi ľahko dynamické, interaktívne aplikácie, ako sú klienti pre internetové obchody, video rozhovory a mnoho ďalších. Flex má širokú ponuku vstavaných komponentov od zoznamov, textových polí a tlačidiel, funkciu drag and drop, kontrola formulárov a animačné efekty. Z užívateľského hľadiska je Flex aplikácia pekne navrhnutá Flash aplikácia, z hľadiska vývojára je Flex štandardnou metódou na výrobu Flash aplikácie, ktorá nie je založená na animáciách.

### 2.1.3 Microsoft Silverlight

Microsoft Silverlight je platforma na vývoj RIA aplikácií, ktorá je porovnateľná s Adobe Flex. Výhodou Silverlightu je kompatibilita s .Net platformou a možnosť použitia dynamických jazykov python, jscript, VB, a perl. Microsoft Silverlight má v súčasnej dobe obmedzenú podporu iných operačných systémov.

## 2.2 Výhody Adobe AIR

Adobe AIR umožňuje:

- použiť existujúci kód stránky a spoločne s *AJAX* technológiou ju preniesť na pracovnú plochu počítača
- použitie rovnakého kódu v prehliadači a na pracovnej ploche
- tvorba platformne nezávislých aplikácií
- využiť technológie *WebKit*, *SQLite*, a *ECMAScript*

Aplikácie založené na technológii Adobe AIR môžu byť vyvíjané pomocou technológií:

- Flash, Flex, ActionScript
- HTML, JavaScript, CSS, Ajax
- PDF môže byť v ľubovoľnej aplikácii

Výsledná Adobe AIR aplikácie môže byť:

- Založená na Flash alebo Flex, hlavný obsah je tvorený Flash/Flex (SWF)
- Založená na Flash alebo Flex s HTML alebo PDF, obsah je tvorený Flash/Flex (SWF) s vloženým HTML (HTML, JavaScript, CSS) alebo PDF obsahom
- Založená na HTML, obsah je tvorený pomocou (HTML, JavaScript, CSS)
- Založená na HTML s Flash/ Flex alebo PDF obsahom. Aplikácia je tvorená pomocou HTML s vkladným Flash/Flex (SWF) alebo PDF obsahom

HTML a Javascript sú spracovávajúce v Adobe AIR WebKit HTML/ JavaScript jadrom. Užívatelia pracujú rovnako s AIR aplikáciou ako s natívnou aplikáciou. Runtime sa inštaluje iba raz na užívateľov počítač a aplikácie sa spúšťajú rovnakým spôsobom ako natívne aplikácie. [1]

### **2.3 Vlastnosti AIR aplikácie**

AIR aplikácia môže byť spustená v internetovom prehliadači alebo ako samostatná aplikácia. Každý typ aplikácie iné vlastnosti. Vlastnosti AIR aplikácie spustenej v internetovom prehliadači a na pracovnej ploche ako plnohodnotná aplikácia je zhrnuté v nasledujúcej tabuľke č. 1. [5]

<b>Vlastnosť</b>	<b>Aplikácia v internetovom prehliadači</b>	<b>Aplikácia na pracovnej ploche</b>
<b>Šírenie aplikácie</b>	Aplikácie môžu byť šírené, používané a preskúmané. Aplikácia nevyžaduje inštaláciu.	Inštalované aplikácie majú väčšiu životnosť, funkcionality a väčší výkon. Aplikáciu je možné nainštalovať do internetového prehliadača alebo ako tradičnú desktopovú aplikáciu.
<b>Inštalácia aplikácie</b>	Aplikácia je aktualizovaná nahraním nového obsahu na stránku.	AIR umožňuje aktualizovať aplikácie pomocou API po nahraní nového obsahu na stránku.
<b>Aktualizácia aplikácie</b>	Aplikácia môže byť spustená na viacerých operačných systémoch a internetových prehliadačoch.	AIR aplikácia je platformne nezávislá, môže byť inštalovaná a spúšťaná na viacerých operačných systémoch.
<b>Programovacie jazyky</b>	JavaScript je vykonávaný internetovým prehliadačom a ActionScript Adobe Flash prehrávačom.	Integrované JavaScript a ActionScript virtuálne stroje sú kompatibilné s internetovým prehliadačom.
<b>Schopnosť spustenia aplikácie na pozadí (Background capability)</b>	Aplikácia môže byť spustená len vo viditeľnom okne internetového prehliadača	Aplikácia môže byť spustená na pozadí a využívať oznámenia ako tradičná desktopová aplikácia.
<b>Životnosť</b>	Činnosť aplikácie je obmedzená na životnosť relácie (session). Keď sa vypne internetový prehliadač, informácie sa stratia.	RIA aplikácia inštalovaná môže ukladať informácie a môže pracovať offline.
<b>Integrácia s pracovnou plochou</b>	Aplikácia je spustená v bezpečnom prostredí, integrácia s pracovnou plochou je obmedzená.	Aplikácia môže pristupovať k súborovému systému, schránke (clipboard), využívať funkciu drag and drop, systémové oznámenia, oznamovací listu a atď.
<b>Užívateľské rozhranie, ovládanie</b>	Aplikácia je spustená v internetovom prehliadači, ktorý má vlastné ovládacie prvky, vzhľad a integráciu s pracovnou plochou.	Aplikácia má prispôbitelné užívateľské rozhranie a integráciu s pracovnou plochou.
<b>Dátové úložisko</b>	Aplikácia má obmedzené dátové úložisko, ktoré môže internetový prehliadač zmazať.	Aplikácia má neobmedzený prístup k lokálnemu súborovému systému, lokálnej databáze a šifrovanému úložisku.

Tabuľka 1: Vlastnosti AIR aplikácie

### 3 API Adobe AIR

Adobe AIR odstraňuje uzavretosť aplikácií voči jednému operačnému, na ktorom je aplikácia spúšťaná. AIR aplikácia môže pristupovať k súborovému systému, využívať štandardné prvky užívateľského rozhrania operačného systému, komunikovať po sieti, dáta ukladať do lokálnej SQL databázy, pracovať s obsahom systémovej schránky, asociovať si typy súborov, funkciu Drag and Drop, ukladať dáta v zašifrovanej forme v uložisku aplikácie, pracovať so zvukom a DRM.<sup>2</sup>

Pri tvorbe vzhľadu AIR aplikácie je možné použiť vzhľad komponentov operačného systému, na ktorom je aplikácia spúšťaná alebo si vytvoriť úplne vlastný vzhľad podľa potreby. Tiež je možné použiť jednotný vzhľad pre všetky operačné systémy pomocou technológie Adobe Flex.



Obr. 1: Neštandardný vzhľad AIR aplikácie

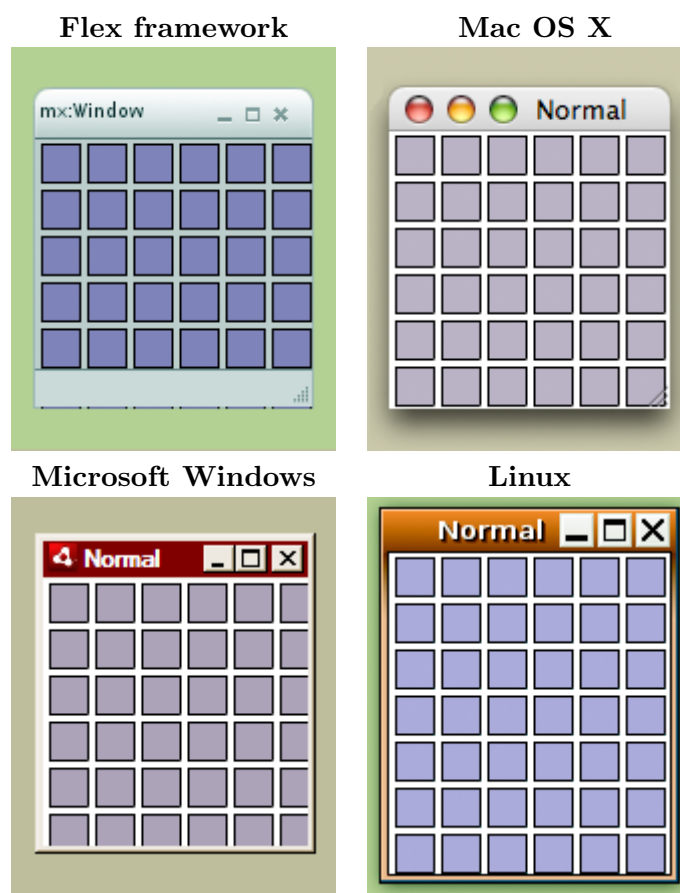
#### 3.1 Natívne okná a menu aplikácie a panel úloh

##### 3.1.1 Natívne okná

AIR umožňuje použiť multiplatformné API pre tvorbu okien aplikácie pomocou Flash, Flex a HTML programovania. AIR umožňuje ľahké prispôbenie vzhľadu aplikácie. Okná aplikácie môžu mať vzhľad ako štandardné desktopové aplikácie operačného systému, na ktorom je aplikácia spustená, jednotný vzhľad Flex frameworku alebo vlastný vzhľad. Vzhľad aplikácie sa dá úplne upraviť podľa potreby s využitím priesvitnosti a alpha blending-u. Okno aplikácie nemusí mať štandardný obdĺžnikový tvar. Vzhľad štandardné okna na rôznych operačných systémoch je ukázaný v tabuľke č. 2.

Pre každé okno je možné nastaviť nasledujúce vlastnosti podľa tabuľky č. 3. Nové

<sup>2</sup>Popis Adobe AIR API je prevzatý z literatúry [1, 2].



Tabuľka 2: Vzhľad natívneho okna v rôznych operačných systémoch

Vlastnosť	Prednastavená hodnota	Popis
systemChrome	standard	Hlavička okna s ovládacími prvkami (standard, none)
type	normal	Typ okna (Normal, Utility, Lightweight)
transparent	false	Priehľadnosť okna (true, false)
maximizable	true	Schopnosť maximalizácie
minimizable	true	Schopnosť minimalizácie
resizable	true	Zmena rozmerov okna

Tabuľka 3: Vlastnosti okna



natívne okno môžeme vytvoriť pomocou objektu *air.NativeWindow*. Vlastnosti okna môžeme upraviť objektom *air.NativeWindowInitOptions*, ktorý predáme ako vstupný parameter objektu *air.NativeWindow*.

---

```
function createNativeWindow() {
    //nastavenie vlastností okna
    var options = new air.NativeWindowInitOptions();
    options.transparent = false;
    options.systemChrome = air.NativeWindowSystemChrome.STANDARD;
    options.type = air.NativeWindowType.NORMAL;
    //vytvorenie okna
    var newWindow = new air.NativeWindow(options);
    newWindow.title = "A_title";
    newWindow.width = 600;
    newWindow.height = 400;
    //aktivácia a zobrazenie okna
    newWindow.activate();
}
```

---

Výpis 1: Vytvorenie nového natívneho okna

Okno aplikácie môže obsahovať HTML alebo SWF obsah.

### Udalosti okna

Na zachytenie udalostí vyvolaných oknom je potrebný zaregistrovaný poslucháč v inštancii okna. Nasledujúci príklad ukazuje zaregistrovanie poslucháča na udalosť zatvorenie okna.

---

```
window.nativeWindow.addEventListener(air.Event.CLOSING, onCloseEvent);
```

---

Výpis 2: Nastavenie poslucháča na udalosť okna

Môžeme zachytávať nasledujúce udalosti okna:

- **activate** - Nastane, ak sa okno stane aktívnym (get focus).
- **deactivate** - Nastane, ak sa okno stane neaktívnym (lost focus).
- **closing** - Odosiela sa, ak sa okno zatvára. Nastane, ak sa užívateľ pokúsi okno zatvoriť tlačidlom na zavretie okna v záhlaví okna.
- **close** - Nastane, keď sa okno zatvorí.

### 3.1.2 Natívne menu aplikácie

Triedy pre natívne menu umožňujú vytvoriť menu aplikácie vo vzhľade operačného systému, na ktorom je aplikácia spustená. *NativeMenu* objekty môžu byť použité pre menu aplikácie (k dispozícii na Mac OS X), menu okna (k dispozícii na platformách Windows a Linux), kontextové menu a pop-up menu.

AIR podporuje nasledujúce typy menu:

**Application menu** je globálne menu, ktoré platia pre celú aplikáciu. Toto menu je podporované na Mac OS X, ale nie na operačnom systéme Windows alebo Linux. Na Mac

OS X operačnom systéme sa automaticky vytvorí menu aplikácie. Pomocou AIR API je možné pridať položku a ponuku do štandardného menu. K položkám je možné pridať poslucháčov pre manipuláciu s existujúcou ponukou príkazov, alebo odstrániť existujúce položky.

**Window menu** je menu spojené s jedným oknom, a zobrazí sa pod hlavičkou. Ponuky môžu byť pridané do okna tým, že sa vytvorí objekt `NativeMenu` a ten sa pridá do vlastnosti menu `NativeWindowObject`. Window menu sú podporované na operačných systémoch Windows a Linux, ale nie na Mac OS X. Natívne menu môže byť použité iba s oknom, ktoré má nastavený štandardný vzhľad (`systemChrome`).

**Kontextové menu** je menu, ktoré sa zobrazuje po kliknutí pravým tlačidlom myši na interaktívny objekt v SWF obsahu alebo dokument element v HTML obsahu. Kontextové menu môžeme pomocou triedy `NativeMenu` alebo `ContextMenu`. V HTML obsahu je možné použiť Webkit HTML a JavaScript API na pridať kontextového menu k HTML elementu.

**Dock a menu v systémovej lište** Tieto ikonové menu sú podobné kontextovému menu a sú zaradené medzi ikony aplikácií v Mac OS X dock alebo do oznamovacej oblasti na hlavnom paneli vo Windows a Linuxe. Dock a menu v systémovej lište používa triedy `NativeMenu`. Na Mac OS X sa do menu pridávajú štandardné položky operačného systému. Na operačných systémoch Windows alebo Linux neexistuje štandardné menu.

**Pop-up menu** je ako kontextové menu, ale nie je nevyhnutne spojené s konkrétnym objektom aplikácie alebo jeho komponentov. Pop-up menu môže byť zobrazené kdekoľvek v okne aplikácie zavolaním metódy `display()` objektu `NativeMenu`.

**Vlastné menu** Natívne menu je zostavené úplne podľa operačného systému a existuje mimo Flash a HTML interpretačných modelov. Vlastné menu je možné vytvoriť pomocou MXML, ActionScript, alebo JavaScriptu.

Balíček	Triedy
flash.display	NativeMenu,NativeMenuItem
flash.ui	ContextMenu, ContextMenuItem
flash.events	Event

Tabuľka 4: Triedy Adobe AIR menu

### 3.1.3 Ikona v paneli úloh

Mnohé operačné systémy majú panel úloh, ako je v Mac OS X dock, ktorý môže obsahovať ikony predstavujúce aplikácie. Adobe AIR poskytuje rozhranie pre interakciu s ikonou na paneli úloh prostredníctvom vlastnosti `NativeApplication.nativeApplication.icon`.

AIR vytvorí objekt `NativeApplication.nativeApplication.icon` automaticky. Typ objektu je **DockIcon** alebo **SystemTrayIcon** v závislosti na operačnom systéme. Môžeme určiť, ktorú z podtried `InteractiveIcon` podporuje súčasný operačný systém pomocou `NativeApplication.supportsDockIcon` a `NativeApplication.supportsSystemTrayIcon` vlastnosti. Základná `InteractiveIcon` trieda poskytuje tieto vlastnosti šírka, výška, a bitmapy,

ktoré môžeme použiť k zmene obrázku ikony. Prístup nedostupným vlastnostiam DockIcon alebo SystemTrayIcon na zlom operačnom systéme vygeneruje chybu.

## 3.2 Súbory a dáta

V AIR aplikáciách je možné pracovať s lokálnym súborovým systémom, vyžívať funkciu *Drag and drop*, prácu so *systémovou schránkou*, lokálnou *SQL databázou*, ukladať zašifrované údaje, overovať XML podpis, pracovať s bajtovým poľami.

### 3.2.1 Práca s lokálnym súborovým systémom

Adobe AIR obsahuje triedy, ktoré je možné použiť na prístup, vytváranie a spravovanie súborov a zložiek. Tieto triedy sú obsiahnuté v balíčku *flash.filesystem*.

Trieda	Popis
File	Objekt File predstavuje cestu k súboru alebo zložke. Pomocou objektu file je možné vytvoriť ukazovateľ na súbor alebo zložku.
FileMode	Trieda FileMode definuje konštanty používané v parametri fileMode metód open() a openAsync() triedy FileStream. FileMode parameter týchto metód určuje schopnosti objektu FileStream akonáhle je súbor otvorený, medzi ktoré patrí písanie, čítanie, pripojenie a aktualizácia.
FileStream	Objekt FileStream slúži na otvorenie súboru pre čítanie a zápis. Po vytvorení objektu File, ktorý ukazuje na existujúci alebo nový súbor, je možné pristupovať k obsahu súboru a manipulovať s ním.

Tabuľka 5: Triedy určené pre prácu s lokálnym súborovým systémom

Niektoré metódy majú synchronnú a asynchronnú variantu:

- File.copyTo() a File.copyToAsync()
- File.deleteDirectory() a File.deleteDirectoryAsync()
- File.deleteFile() a File.deleteFileAsync()
- File.getDirectoryListing() a File.getDirectoryListingAsync()
- File.moveTo() a File.moveToAsync()
- File.moveToTrash() a File.moveToTrashAsync()

Asynchronne verzie funkcií umožňujú inicializovať procesy, ktoré bežia na pozadí a odosiľajú informácie o dokončení a chybách. Zatiaľ čo je funkcia pustená na pozadí, môže sa vykonávať iný kód. Ku každej asynchronnej funkcii musí byť priradený poslucháč pomocou metódy addEventListener(), ktorý volá funkciu.

Synchronne verzie funkcií umožňujú písanie jednoduchšieho kódu, ktorý nemusí byť priradený priradeného poslucháča na udalosť. Počas vykonávania tejto funkcie nemôže byť spustený iný kód, pokiaľ nie je funkcia dokončená.

Trieda File môže byť použitá na nasledujúce operácie:

- Získanie cesty k špeciálnym zložkám(dokumentom, ploche, zložke užívateľa), zložky, z ktorej bola aplikácia spustená.
- Kopírovať súbory a zložky
- Presúvať súbory a zložky
- Mazat' súbory a zložky, presúvať ich do koša
- Vypisovať zoznam súborov a zložiek v zložke
- Vytvárať dočasné súbory a zložky
- Získať informácie o súbore alebo zložke

Trieda FileStream umožňuje AIR aplikácii čítať a zapisovať do súborového systému, pracovať s XML dokumentmi.

### 3.2.2 Drag-and-drop

Pomocou tried obsiahnutých drag-and-drop API je možné obohatiť aplikáciu o gesta. Gesto je akcia zo strany užívateľa sprostredkovaná operačným systémom a aplikácia, vyjadrujúca zámer kopírovania, presúvania alebo odkazovania sa na informácie. *Drag-out gesto* nastane, keď užívateľ ťahá objekt von z komponenty alebo aplikácie. *Drag-in gesto* nastane, keď užívateľ ťahá objekt do komponenty alebo aplikácie. Podporované formáty sú:

- Rastrové obrázky (bitmapy)
- Súbory
- HTML formátovaný obsah
- Text
- URL

### 3.2.3 Prácu so systémovou schránkou

Pomocou tried v API pre prácu so systémovou schránkou je možné *kopírovať informácie do a zo schránky*. Prenášané môžu byť nasledujúce formáty:

- Rastrové obrázky
- Súbory
- Text
- HTML-formátovaný text

- Rich Text Format dáta
- URL
- Serializované objekty

### 3.2.4 Lokálna SQL databáza

Adobe AIR zahŕňa schopnosť vytvárať a pracovať s lokálnou SQL databázou. Runtime obsahuje SQL databázu s podporou so štandardnými SQL operáciami. Používa open source **SQLite databázový systém**. Lokálna SQL databáza môže byť použitá na ukladanie lokálnych, perzistentných dát. Napríklad môže byť použitá pre dáta aplikácie, nastavenie aplikácie užívateľom, ukladanie dokumentov, alebo akýkoľvek iný typ dát, ktoré majú byť aplikáciou uložené.

#### Vytvorenie databázy

Pred vytvorením databázového súboru je potrebné vytvoriť inštanciu objektu *SQLConnection*. Pripojenie k databázovému súboru je možné otvoriť v **synchronnom režime** pomocou metódy `open()` alebo **asynchronom režime** metódou `openAsync()`. Ak je zavolaná funkcia `open()` alebo `openAsync()` s inštanciou `File` na neexistujúci súbor, bude vytvorený nový databázový súbor, ktorý otvorí spojenie. Ak je zavolaná metóda pre otvorenie spojenia s nulovou referenciou na súbor, nový databázový súbor bude vytvorený v pamäti počítača.

Nasledujúci kód ukazuje proces vytvorenia databázového súboru (novej databázy) pomocou asynchrónneho režimu spustenia. V tomto prípade je súbor databázy uložený v uložisku aplikácie s názvom súboru „DBSample.db“.

---

```
// vloženie odkazu na AIRAliases.js
var conn = new air.SQLConnection();
conn.addEventListener(air.SQLEvent.OPEN, openHandler);
conn.addEventListener(air.SQLErrorEvent.ERROR, errorHandler);
// databázový súbor bude uložený v uložisku aplikácie
var folder = air.File.applicationStorageDirectory;
var dbFile = folder.resolvePath("DBSample.db");
conn.openAsync(dbFile);
function openHandler(event)
{
    air.trace("databáza bola úspešne vytvorená");
}
function errorHandler(event)
{
    air.trace("Chybová správa:", event.error.message);
    air.trace("Detaily:", event.error.details);
}
```

---

Výpis 3: Vytvorenie nového databázového súboru

---

```

// vloženie odkazu na AIRAliases.js
var conn = new air.SQLConnection();
// databázový súbor bude uložený v~uložisku aplikácie
var folder = air.File.applicationStorageDirectory;
var dbFile = folder.resolvePath("DBSample.db");
try
{
    conn.open(dbFile);
    air.trace("databáza_bola_úspešne_vytvorená_");
}
catch (error)
{
    air.trace("Chybová_správa:", error.message);
    air.trace("Detaily:", error.details);
}

```

---

Výpis 4: Vytvorenie databázy v synchrónnom režime

### Posielanie SQL príkazov

Pre poslanie SQL príkazu je potrebné mať otvorené spojenie s databázovým súborom. SQL príkaz sa zapisuje do inštančie objektu *SQLStatement*. Príkazy je možné zasielať v **synchrónnom** alebo **asynchrónnom režime**. Nasledujúci príklad ukazuje vytvorenie tabuľky s názvom „employees“ v existujúcom databázovom súbore, pomocou asynchrónneho režimu. Tento kód predpokladá existujúcu inštanciu *SQLConnection* s názvom *Conn*.

---

```

// vloženie odkazu na AIRAliases.js
// vytvorenie a otvorenie inštančie SQLConnection s~názvom conn
var createStmt = new air.SQLStatement();
createStmt.sqlConnection = conn;
var sql =
    "CREATE_TABLE_IF_NOT_EXISTS_employees_(\" +
    "empId_INTEGER_PRIMARY_KEY_AUTOINCREMENT_\" +
    "firstName_TEXT_\" +
    "lastName_TEXT_\" +
    "salary_NUMERIC_CHECK_(salary_>_0)\" +
    ")";
createStmt.text = sql;
createStmt.addEventListener(air.SQLEvent.RESULT, createResult);
createStmt.addEventListener(air.SQLErrorEvent.ERROR, createError);
createStmt.execute();
function createResult(event)
{
    air.trace("Tabuľka_bola_vytvorená");
}
function createError(event)
{
    air.trace("Chybová_správa:", event.error.message);
    air.trace("Detaily:", event.error.details);
}

```

---

Výpis 5: Posielanie SQL príkazu

Pri použití synchrónneho režimu je potrebné funkciu `createStmt.execute()` volať v **try bloku** a chybové hlášky zachytávať pomocou **catch bloku** výnimky.

```

try
{
    createStmt.execute();
    air.trace("Tabuľka_bola_vytvorená");
}
catch (error)
{
    air.trace("Chybová_správa:", error.message);
    air.trace("Detaily:", error.details);
}

```

Výpis 6: Ošetrenie výniek SQL

### 3.2.5 Ukladanie zašifrovaných dát

Pre každú aplikáciu poskytuje AIR runtime trvalé šifrované uložiisko (*encrypted local store (ELS)*) inštalované na počítači užívateľa. Toto umožní pracovať s dátami, ktoré sú uložené na lokálnom disku v zašifrovanej podobe, ktorá nemôže byť ľahko dešifrovaná inými užívateľmi. Každá aplikácia má vlastné uložiisko pre každého užívateľa samostatne. Dáta sú šifrované *AES-CBC 128-bit šifrovacím algoritmom*.

Lokálne šifrované uložiisko nie je chránené pred prístupom iných aplikácií. Dáta v ELS môžu byť stratené z rôznych dôvodov, pri odinštalovaní aplikácie alebo zmene ID vydavateľa aplikácie. S ELS by sa malo zaobchádzať ako s dočasnou pamäťou a neukladať trvalé dáta.

#### Uloženie dát

Pre uloženie dát do ELS slúži statická metóda `setItem()` triedy `EncryptedLocalStore`. Dáta sú uložené v hash tabuľke pomocou reťazca ako kľúča, s dátami uloženými v byte poli.

```

var str = "Bob";
var bytes = new air.ByteArray();
bytes.writeUTFBytes(str);
air.EncryptedLocalStore.setItem("firstName", bytes, false);

```

Výpis 7: Uloženie objektu do ELS

Tretí parameter *stronglyBound* metódy `setItem()` je voliteľný. Ak je tento parameter nastavený na **true**, ELS pripojí aplikáciu k ukladanej položke digitálny podpis aplikácie.

Položky uložené s vlastnosťou `stronglyBound` nastavenou na **true** (silno viazané položky), môžu byť načítané len v aplikácii, ktorá ich uložila. Po aktualizácii aplikácie nebude možné prečítať silno viazané položky prečítať v novej aplikácii. Ak je parameter `stronglyBound` nastavený na **false** (prednastavená hodnota), musí súhlasiť len identifikátor vydavateľa aplikácie.

## Načítanie dát

Na načítanie dát z ELS slúži metóda *getItem()*. Vo vstupnom parametri metódy sa zadáva identifikátor uloženej položky. Údaje sa vracajú ako objekt `ByteArray`.

---

```
var str = "Bob";
var bytes = new air.ByteArray();
bytes.writeUTFBytes(str);
air.EncryptedLocalStore.setItem("firstName", bytes, false);
```

---

Výpis 8: Prečítanie objektu z ELS

## Zmazanie uložených dát

Jednu položku z ELS je možné odstrániť metódou *EncryptedLocalStore.removeItem()*. Vstupným parametrom metódy je identifikátor uloženej položky.

---

```
air.EncryptedLocalStore.removeItem("firstName");
```

---

Výpis 9: Zmazanie objektu z ELS

Všetky uložené dáta z ELS je možné odstrániť zavolaním metódy *EncryptedLocalStore.reset()*.

---

```
air.EncryptedLocalStore.reset();
```

---

Výpis 10: Zmazanie ELS

## 3.3 Média s bohatým obsahom

AIR aplikácia môže pracovať aj iným obsahom ako je HTML alebo SWF obsah. V AIR aplikáciách je možné pracovať so zvukom, DRM (digital rights management) a PDF obsahom.

### 3.3.1 PDF obsah

AIR aplikácia môže zobrazit' PDF obsah pomocou triedy *HTMLLoader*, WebKit jadra a *Adobe Reader browser plug-inu*. PDF obsah môže byť zobrazený na celú výšku alebo šírku, prípadne ako časť rozhrania. Pre zobrazenie PDF obsahu je potrebný nainštalovaný Adobe Reader alebo Adobe Acrobat vo verzii 8.1 alebo novší.

### Zistenie podpory pre PDF

Podporu pre zobrazenie PDF obsahu je možné zistiť pomocou vlastnosti *pdfCapability* objektu *HTMLLoader*. Táto vlastnosť môže byť nastavená na jednu z nasledujúcich hodnôt uvedených v tabuľke č. 6.

Nasledujúci kód zisťuje, či je možné zobraziť obsah v PDF formáte a AIR aplikácii a či vlastnosť *HTMLPDFCapability* nevracia chybový kód.



Konštanta HTMLPDFCapability	Popis konštanty
STATUS_OK	Bola zistená postačujúca verzia Adobe Reader (8.1 alebo novšia) a PDF obsah môže byť zobrazený do HTMLLoader objektu.
ERROR_INSTALLED_READER_NOT_FOUND	Adobe Reader nebol nájdený. HTMLLoader nemôže zobrazit' PDF obsah.
ERROR_INSTALLED_READER_TOO_OLD	Adobe Reader bol nájdený, ale verzia je príliš stará. HTMLLoader nemôže zobrazit' PDF obsah.
ERROR_PREFERRED_READER_TOO_OLD	Bola zistená postačujúca verzia (8.1 alebo novšia) Adobe Reader, ale verzia Adobe Reader, ktorá je nastavená pre prácu s PDF obsahom je staršia ako Reader 8.1. HTMLLoader nemôže zobrazit' PDF obsah.

Tabuľka 6: Konštanty objektu pdfCapability

```

if (air.HTMLLoader.pdfCapability == air.HTMLPDFCapability.STATUS_OK)
{
    air.trace("PDF_môže_byt'_zobrazený.");
} else {
    air.trace("PDF_nemôže_byt'_zobrazené._Chybový_kód:", HTMLLoader.pdfCapability);
}

```

Výpis 11: Zistenie podpory pre PDF obsahu

### Zobrazenie PDF obsahu

PDF obsah je možné zobrazit' vytvorením inštancie *HTMLLoader*, nastavením rozmerov a cesty k PDF. PDF obsah môže byť zobrazený rovnakom ako v internetovom prehliadači. Nasledujúci príklad načíta PDF z externej stránky pomocou *iframe*.

```

<html>
  <body>
    <h1>PDF test</h1>
    <iframe id="pdfFrame" width="100%" height="100%"
      src="http://www.example.com/test.pdf"/>
  </body>
</html>

```

Výpis 12: Zobrazenie PDF obsah

### 3.3.2 Práca so zvukom

AIR aplikácie umožňujú nasledujúcu prácu zo zvukom:

- Vložiť externý zvukový súbor ako je MP3 do aplikácie.
- Vložiť zvukovú informáciu do SWF súboru, nahrat' ju pomocou `<script src="[swfFile].swf" type="application/x-shockwave-flash"/>` a prehrať.
- Získať audio vstup pomocou mikrofónu pripojeného k počítaču.
- Pristupovať k zvukovým dátam umiestneným na serveri.
- Dynamicky generovať zvuk.

Pri načítaní zvukových dát z externého zvukového súboru, je možné začať prehrávať začiatok zvukového súboru, zatiaľ čo sa stále načítava zvyšok zvukových dát. AIR podporuje zvukové súbory, ktoré sú uložené vo formáte *MP3*. Nie je možné priamo načítať alebo prehrávať zvukové súbory v iných formátoch, ako je *WAV* alebo *AIFF*. Pri práci so zvukom sa využívajú triedy z balíka *runtime.flash.media*. Trieda *Sound* umožňuje prístup k informáciám pri načítaní zvukového súboru alebo priradení funkcie na udalosť, ktorá spustí prehrávanie. Keď začnete hrať zvuk, AIR umožní prístup k objektu *SoundChannel*. Pri každom zvuku, ktorý prehráva, aplikácia používa vlastný *SoundChannel* objekt. Celkový výstup zo všetkých objektov *SoundChannel* je to, čo hrajú reproduktory. Pomocou *SoundChannel* objektu je možné upraviť vlastnosti zvuku, prípadne zastaviť prehrávanie. Ak je potrebné kontrolovať kombinovaný zvuk, k tomu slúži trieda *SoundMixer*, ktorá dokáže ovládať zmiešaný výstup.

Aplikácia môže pristupovať k nasledujúcim zdrojom zvukov:

- Externé zvukové súbory nahrané v priebehu programu,
- Zvukové zdroje vložené do SWF súboru.
- Zvukové dáta z mikrofónu pripojeného k systému užívateľa.
- Zvukové dáta prenášané zo vzdialeného média servera.
- Zvukové dáta dynamicky generované pomocou *sampleData* udalosti.

Zvuková AIR architektúra obsahuje nasledujúce triedy:

#### Správa digitálnych práv

Adobe Flash Media Rights Management Server (FMRMS) poskytuje vydavateľom médií možnosť distribuovať obsah, konkrétne *FLV* a *MP4* súborov. Vydavateľ môže distribuovať médiá ako šifrované *FLV* súbory, ktoré je možné stiahnuť a hrať v Adobe Media Player, alebo v Adobe AIR aplikácií, ktorá využíva pre správu digitálnych práv (DRM) API. S FMRMS môžu poskytovatelia obsahu používať identitne založené licencie k ochrane ich obsahu.

<b>Trieda</b>	<b>Popis triedy</b>
Sound	Táto trieda spracováva načítanie zvuku, riadi základné zvukové vlastnosti a prehráva zvuk.
SoundChannel	Keď aplikácia prehráva Sound objekt, je vytvorený nový objekt SoundChannel na ovládanie prehrávania. Tento objekt môže ovládať hlasitosť ľavého a pravého zvukového kanála. Každý prehrávaný zvukový súbor má vlastný SoundChannel objekt.
SoundLoaderContext	Trieda SoundLoaderContext určuje, koľko sekúnd bude načítaných do vyrovnávacej pamäti, a či runtime hľadá cross-doménovú politiku súboru na serveri pri načítaní súboru. Objekt SoundLoaderContext sa používa ako parameter Sound.load() metódy.
SoundMixer	Trieda SoundMixer ovláda prehrávanie a bezpečnostné vlastnosti, ktoré sa týkajú všetkých zvukov v aplikácií. V skutočnosti sú viac zvukových kanálov je zmiešavaných prostredníctvom spoločného objektu SoundMixer. Hodnoty vlastností objektu SoundMixer ovplyvňujú všetky objekty SoundChannel, ktoré hrajú zvuk.
SoundTransform	Trieda SoundTransform obsahuje hodnoty, ktoré riadia hlasitosť zvuku a šírenie mono signálu v stereo alebo multi-kanálové zvukové pole (pannig). Objekt SoundTransform môže byť aplikovaný na jednotlivé SoundChannel objekty, aby globálny SoundMixer objekt, alebo na objekt Microphone.
ID3Info	ID3Info objekt obsahuje vlastnosti, ktoré predstavujú ID3 metadáta informácie, ktoré sú často uložené v MP3 zvukových súboroch.
Microphon	Trieda Microphone predstavuje mikrofón alebo iné zvukové vstupné zariadenie pripojené k počítaču používateľa. Audio vstup z mikrofónu môže byť smerovaný na reproduktory alebo poslaný na vzdialený server. Microphone objekt ovládanie zisku, vzorkovaciu frekvenciu a ďalšie vlastnosti.

Tabuľka 7: Triedy pre prácu so zvukom

Udalosť	Popis
drmAuthenticate	Definované v triede DRMAuthenticateEvent. Táto udalosť sa odosiela, ak objekt NetStream sa snaží prehrať DRM šifrovaný obsah, ktorý vyžaduje, aby mal oprávnenie na jej prehranie. Medzi vlastnosti tejto udalosti patrí hlavička, usernamePrompt, passwordPrompt, a urlPromp, ktoré je možné použiť na získanie a nastavenie oprávnenia používateľa. Táto udalosť nastane opakovane, pokiaľ objekt NetStream dostane platné oprávnenie užívateľa.
drmError	Sú definované v DRMErrorEvent triede a odoslané, ak sa objekt NetStream snaží prehrať DRM zašifrovaný súbor a nastane DRM chyba. Napríklad, DRM chyba nastane, ak používateľské oprávnenie zlyhá. Tento prípad môže nastať, ak používateľ nemá zakúpené práva pre zobrazenie obsahu alebo preto, že poskytovateľ obsahu nepodporuje zobrazovaciu aplikáciu.
drmStatus	Definovaný v triede DRMStatusEvent. Táto udalosť sa odošle, keď sa začne DRM šifrovaný obsah prehrávať (ak je používateľ overený a má je oprávnenie na obsah). Objekt DRMStatusEvent obsahuje informácie vzťahujúce sa ku kupónu. Napríklad či obsah môže byť k dispozícii offline, alebo kedy končí platnosť kupónu a obsah už nemôže byť videný.
status	Sú definované v events.StatusEvent a odosielajú sa, ak aplikácia sa pokúsi prehrať obsah zašifrovaný pomocou Digital Rights Management (DRM) s odvolaním sa na NetStream.play() metódu. Hodnota tejto vlastnosti je stavový kód "DRM.encryptedFLV".

Tabuľka 8: Udalosti triedy NetStream

Keď sa užívateľ pokúsi prehrať DRM zašifrovaný súbor, aplikácia kontaktuje FMRMS, kontaktuje systém vydavateľa prostredníctvom svojho rozhrania pre poskytovanie služieb (SPI) na overenie užívateľa a získanie licencie, kupónu (*voucher*), ktorý určuje, či má používateľ umožnený prístup k obsahu a na ako dlho. Kupón tiež určuje, či užívateľ môže pristupovať k obsahu offline a na ako dlho.

Identitne založené licencie podporujú aj anonymný prístup. Napríklad, anonymný prístup umožňuje poskytovateľovi distribuovať reklamný obsah alebo umožniť voľný prístup k obsahu pre určený počet dní. Archívny materiál môže byť považovaný za prémiový obsah, ktorý musí byť zaplatený a vyžaduje užívateľské oprávnenie.

Trieda *NetStream* poskytuje jednosmerný prenos medzi Flash Player alebo aplikáciou AIR, a to buď Flash Media Server alebo lokálnym súborovým systémom. Trieda *NetStream* taktiež podporuje progresívny download. Objekt *NetStream* je kanál v *NetConnection* objekte. Trieda *NetStream* odoslania štyri DRM udalosti podľa tabuľky č 8 a obsahuje nasledujúce DRM metódy uvedené v tabuľke č. 9.

Metóda	Popis
resetDRMVouchers()	Odstráni všetky lokálne uložené DRM kupóny. Aplikácia musí znova stiahnuť kupóny pre prístup k šifrovanému obsahu. Napríklad nasledujúci kód odstráni všetky kupóny z archívu: <code>NetStream.resetDRMVouchers();</code>
setDRMAuthenticationCredentials()	Skupina overovacích údajov (užívateľské meno, heslo a typ autentifikácie) potrebných pre objekt <code>NetStream</code> pre autentifikáciu. Platné typy autentifikácie sú DRM a proxy. Pomocou <i>DRM</i> overovania je oprávnenie overené oproti FMRMS. Pomocou <i>proxy</i> overovania, oprávnenie sa overuje proti proxy serveru a musí odpovedať požadovaný proxy server. Napríklad proxy voľba umožňuje použitie autentifikácie oproti proxy serveru, ak podnik vyžaduje takýto krok predtým, než môže používateľ prístupit' k internetu. Ak je použitá anonymná autentifikácia, musí sa užívateľ aj naďalej overiť oproti FMRMS s cieľom získania kupónu a prehrania obsahu. Druhýkrát je možné použiť <code>setDRMAuthenticationCredentials()</code> s DRM voľbou na overovanie oproti FMRMS.
preloadEmbeddedMetadata()	Analyzuje vložené metaúdaje v miestnych médiá súboroch. Ak sa nájdu DRM metadáta, AIR volá <code>onDRMContentData()</code> funkciu.

Tabuľka 9: Metódy triedy `NetStream`

Metóda	Popis metódy
isSetAsDefaultApplication()	Vráti true, ak je AIR aplikácia v súčasnej dobre spojená so zadaným typom súboru.
setAsDefaultApplication()	Vtvorí asociáciu medzi aplikáciou a zadaným typom súboru.
removeAsDefaultApplication()	Odstráni asociáciu medzi aplikáciou a zadaným typom súboru.
getDefaultApplication()	Vráti cestu k aplikácií, ktorá je asociovaná so zadaným typom súboru.

Tabuľka 10: Metódy objektu NativeApplication

Objekt NetStream vyvolá onDRMContentData() a onPlayStatus() funkcie ako výsledok volania metódy preloadEmbeddedMetaData(). OnDRMContentData() funkcia je zavolaná, ak boli nájdené DRM metaúdaje v súbore. OnPlayStatus() funkcia sa zavolá po úplnom spracovaní súboru. OnDRMContentData() a onPlayStatus() funkcie, musia byť priradené inštancii objektu NetStream. Ak sa použije rovnaký NetStream objekt na načítanie kupónu a prehranie obsahu, musí sa počkať na metódu onPlayStatus() volanou preloadEmbeddedMetaData() pred začatím prehrávania.

### 3.4 Interakcia s operačným systémom

V tejto časti je popísaná časť API, ktorá umožňuje spravovať asociáciu súborov v operačnom systéme, sledovať aktivitu užívateľa a získať informácie o Adobe AIR runtime.

#### 3.4.1 Správa asociovaných súborov

Asociácia medzi aplikáciou a typom súboru musí byť uvedená v popisovom súbore aplikácie. Pri inštalácii aplikácie, inštalátor automaticky nastaví pre každý definovaný typ súboru jeho asociáciu, ak nie je inštalovaná aplikácia už nastavená. Na prepísanie existujúcich asociácií slúži metóda *NativeApplication.setAsDefaultApplication()*.

Pomocou metód triedy NativeApplication môžeme spravovať asociáciu súborov. Každá metóda má ako vstupný paramter príponu súboru.

#### 3.4.2 Zistenie AIR schopností

Schopnosti a obmedzenia aplikácia je možné zistiť pomocou nasledujúceho kódu, ktorý zistí v akom prostredí je aplikácia spustená.

---

```

if (window.runtime)
{
    if (air.Security.sandboxType == air.Security.APPLICATION)
    {
        alert("AIR_aplikácia_v~bezpečnom_prostredí.");
    }
    else

```

```

    {
        alert ("AIR_aplikácia_nie_je_bezpečnom_prostredí.")
    }
}
else
{
    alert ("Mimo_Adobe_AIR_runtime")
}

```

Výpis 13: Zistenie obmedzení aplikácie

### 3.4.3 Sledovanie prítomnosti užívateľa

Objekt `NativeApplication` odosiela dve udalosti, ktoré pomáhajú zistiť, kedy užívateľ aktívne používa počítač. Aktivita myši a klávesnice je kontrolovaná v intervale nastavenou vlastnosťou `NativeApplication.idleThreshold`. `NativeApplication` odošle **userIdle** udalosť, ak v nastavenom intervale používateľ nepoužil klávesnicu alebo myš. Ak sa niečo na vstupe klávesnice alebo myši zmení, objekt `NativeApplication` pošle udalosť **userPresent**. `IdleThreshold` interval sa meria v sekundách a východzia hodnota je 300 (5 minút). Pomocou vlastnosti `NativeApplication.nativeApplication.lastUserInput` je možné získať počet sekúnd od poslednej vstupnej informácie užívateľa.

Nasledujúci kód nastaví prahnečinnosti na 2 minútu a priradí poslucháčov pre udalosti `userIdle` a `userPresent`.

```

air . NativeApplication . nativeApplication . idleThreshold = 120;
air . NativeApplication . nativeApplication . addEventListener( air . Event . USER_IDLE, function(event) {
    air . trace("Používateľ_je_nedostupný.");
});
air . NativeApplication . nativeApplication . addEventListener( air . Event . USER_PRESENT, function(
    event) {
    air . trace("Používateľ_je_prítomný.");
});

```

Výpis 14: Sledovanie prítomnosti užívateľa

## 3.5 Sieť a komunikácia

V AIR aplikáciách je možné využiť viacero typov komunikácií na získanie externého obsahu alebo na spoluprácu dvoch aplikácií. Na získanie externého obsahu je možné použiť URL požiadavky tvorené triedou `URLLoader`, `URLVariables`, alebo `socket XML` a binárnu komunikáciu. Na medziaplikačnú komunikáciu slúži trieda `LocalConnection`.

### 3.5.1 Sledovanie pripojenia k sieti

Adobe AIR poskytuje prostriedky na kontrolu zmien v pripojení k sieti z počítača, na ktorom je nainštalovaná AIR aplikácia. Táto informácia je užitočná, pokiaľ aplikácia používa dáta získavané zo siete. Taktiež je možné overiť dostupnosť sieťových služieb.

## Detekcia zmien v pripojení k sieti

AIR aplikácia môže byť spustená v prostredí s neistým a meniacim sa pripojením k sieti. Adobe AIR odošle udalosť vždy, keď sa zmení stav pripojenia k sieti. `NativeApplication` objekt sa stará o kontrolu zmien v pripojení. Pokiaľ má aplikácia reagovať na túto udalosť, je potrebné pripojiť poslucháča.

---

```
air . NativeApplication . nativeApplication . addEventListener( air . Event . NETWORK_CHANGE,
    onNetworkChange);
function onNetworkChange(event)
{
    //Kontrola dostupnosti zdroja
}
```

---

Výpis 15: Sledovanie zmien v sieti

`Event.NETWORK_CHANGE` nezaznamenáva všetku aktivitu v sieti, ale iba to, že pripojenie k sieti sa zmenilo. AIR sa nesnaží interpretovať význam zmien v sieti. Počítač pripojený do siete môže mať viacero skutočných a virtuálnych pripojení, strata pripojenia nemusí znamenať stratu k pripojeniu k zdroju dát. Na druhej strane nové pripojenia nezaručujú zlepšenie dostupnosti zdroja dát. Niekedy nové pripojenie môže zablokovať prístup k zdroju dát (pripojenie k VPN).

Všeobecne platí, že jediný spôsob pre určenie dostupnosti zdroja dát je vyskúšanie pripojenia. Pre tento účel poskytuje balíček *air.net* udalosti, ktoré umožňujú reagovať na zmeny v pripojení na určitého hostiteľa (host).

## Kontrola HTTP pripojiteľnosti

`URLMonitor` trieda určuje, či môže byť zaslaná http požiadavka na zadanú adresu (typicky na port 80). Nasledujúci kód používa inštanciu triedy `URLMonitor` na overenie pripojiteľnosti k Adobe stránke.

---

```
<script src="servicemonitor.swf" type="application/x-shockwave-flash" />
<script>
    var monitor;
    function test ()
    {
        monitor = new air.URLMonitor(new air.URLRequest('http://www.adobe.com'));
        monitor.addEventListener( air . StatusEvent . STATUS, announceStatus);
        monitor.start ();
    }
    function announceStatus(e) {
        air.trace("Status_change. Current_status: " + monitor.available );
    }
</script>
```

---

Výpis 16: Zistenie dostupnosti http zdroja



## Kontrola socket pripojenia

V AIR aplikáciách je možné použiť aj socket pripojenie. Firewally a siet'ové smerovače môžu obmedzovať túto komunikáciu po sieti na nepovolených portoch z bezpečnostných dôvodov. Z toho dôvodu musí vývojár aplikácie zvážiť, že užívatelia nemusia mať možnosť vytvárania socket spojení.

Nasledujúci príklad používa inštanciu triedy *SocketMonitor* na kontrolu zmien pripojení na porte 6667. Tento port sa obvykle využíva na IRC.

---

```
<script src="servicemonitor.swf" type="application/x-shockwave-flash" />
<script>
  function test ()
  {
    socketMonitor = new air.SocketMonitor('www.adobe.com', 6667);
    socketMonitor.addEventListener(air.StatusEvent.STATUS, socketStatusChange);
    socketMonitor.start ();
  }
  function announceStatus(e) {
    air.trace("Status_change_ Current_status:" + socketMonitor.available );
  }
</script>
```

---

### 3.5.2 URL žiadosti a siete

V nasledujúcej časti je popísaný spôsob, akým môže Adobe Air aplikácia komunikovať po sieti, ako pristupovať k dátam z externých zdrojov, posielat' správy medzi serverom a AIR aplikáciou, nahrávať a sťahovať súbory.

#### Trieda URLRequest

Trieda URLRequest umožňuje vytvárať URL žiadosti. Táto trieda umožňuje vytvárať viac než jednoduchý URL reťazec. Runtime môže vytvárať URL požiadavky pomocou nových schém, okrem štandardných ako sú file a http.

Trieda URLRequest disponuje nasledujúcimi vlastnosťami, ktoré môžu byť použité v bezpečnom prostredí aplikácie (security sandbox). Vlastnosti sú uvedené v tabuľke č. 11.

Nasledujúce vlastnosti z tabuľky č. 12 môžu byť použité v ľubovoľnom prostredí.

#### Nastavenie prihlasovacích údajov

Prihlasovacie údaje je možné nastaviť pomocou funkcie *setLoginCredentialsForHost()*. Táto funkcia má tri vstupné parametre: **server** - určuje URL adresu, kde sa nachádza prihlasovanie, **user\_name** - prihlasovacie meno a **passwd** - prihlasovacie heslo. Nasledujúce adresy „example.com“, „www.example.com“, and „sales.example.com“ sa považujú za jedinečné. Preto je potrebné pre každú z adries zadať prihlasovacie údaje samostatne, ak sú vyžadované.

Vlastnosť	Popis
followRedirects	Určuje, či majú byť presmerovania nasledované (true, prednastavená hodnota) alebo nie (false). Táto vlastnosť je podporovaná iba v runtime.
manageCookies	Určuje, či protokol HTTP má spravovať cookies (true, prednastavená hodnota) alebo nie (false) pre požiadavku. Táto vlastnosť je podporovaná iba v runtime.
authenticate	Určuje, či majú byť overovacie žiadosti spracovávané (true) pre požiadavku. Táto vlastnosť je podporovaná iba v runtime. Štandardne je nastavené zobrazovanie autentifikačného okna, ak je vyžadované. Užívateľské meno a heslo môže byť nastavené pomocou triedy URLRequest.
cacheResponse	Určuje, či majú byť dáta úspešných požiadaviek uložené do medzipamäte (cache). Táto vlastnosť je podporovaná iba v runtime. Štandardne je táto vlastnosť nastavená na true.
useCache	Určuje, či majú byť použité dáta z medzipamäte, ak sú dostupné. Východzia hodnota je nastavená na true. Táto vlastnosť je podporovaná iba v runtime.
userAgent	Určuje user-agent reťazec, ktorý bude použitý v HTTP požiadavke.

Tabuľka 11: Vlastnosti triedy URLRequest - bezpečné prostredie

Vlastnosť	Popis
contentType	MIME typ dát posielaých v URL požiadavke.
data	Objekt obsahujúci dáta, ktoré sa pošlú s URL požiadavkou.
method	Typ HTTP požiadavky. Môže byť nastavená hodnota "GET" alebo "Post". Prednastavená hodnota je "GET".
requestHeaders	Pole hlavičiek http požiadaviek, ktoré majú byť pripojené k http požiadavke.
url	Určuje URL, kde sa pošle požiadavka.

Tabuľka 12: Vlastnosti triedy URLRequest

---

```
air .URLRequestDefaults.setLoginCredentialsForHost("www.example.com", "Ada", "love1816\${X}");
```

---

### Výpis 17: Nastavenie prihlasovacích údajov

## URL Schémy

Okrem štandardných URL schémom http a https, ktoré sú dostupné v akomkoľvek bezpečnom prostredí, je v AIR aplikáciách možné použiť nasledujúce URL schémy:

- **file** - Pomocou tejto schémy je možné zadať cestu k súboru v súborovom systéme. Napríklad: `file://c:/AIR Test/test.txt`
- **mailto** - mailto schéma v objekte `URLRequest` bude predaná funkcii `navigateToURL()` - otvorenie URL v internetovom prehliadači.
- **app** - Pomocou app je možné zadať relatívnu cestu k súborom umiesteným v adresári, kde je nainštalovaná aplikácia a je podadresároč. Nasledujúci príklad ukazuje relatívnu cestu zložke `resources` v adresári, kde je nainštalovaná aplikácia. Napríklad: `app:/resources/test.txt`
- **app-storage** - Pomocou tejto URL schémy je možné zadať relatívnu cestu k uložísku aplikácie (application store directory). Pre každú nainštalovanú aplikáciu, AIR definuje unikátne úložisko pre každého užívateľa samostatne. Napríklad nasledujúca cesta ukazuje na súbore `prefs.xml` v zložke `settings` uložísku aplikácie. Uložisko aplikácie je závislé od užívateľského mena, ID aplikácie a vydavateľa. Napríklad: `app-storage:/settings/prefs.xml`

Úložisko aplikácie sa nachádza:

- *Mac OS* - `/Users/user name/Library/Preferences/applicationID.publisherID/Local Store/`
- *Windows* - `user name/Application Data/applicationID.publisherID/Local Store/`
- *Linux* - `/home/user name/.appdata/applicationID.publisherID/Local Store/`

### 3.5.3 Práca s externými dátami

Runtime obsahuje mechanizmy na načítanie dát z externých zdrojov. Tieto zdroje môžu mať statický obsah, ako je textový súbor, alebo dynamický obsah, ktorý je generovaný pomocou rôznych skriptov na serveri a načítaný z databázy. Dáta môžu byť rôzne formátované. Dáta je možné posielat' aj na server.

Runtime obsahuje `URLLoader` a `URLVariables` pre načítanie externých dát. Trieda `URLLoader` sťahuje dáta z požadovanej URL ako text, binárne dáta, alebo URL zakódované premenné. Trieda `URLLoader` sa dá použiť na sťahovanie textových a XML

súborov alebo iných informácií použitých v aplikácii. URLLoader využíva runtime event-manipulace model, ktorý umožňuje poslúchať udalosti: complete, httpStatus, ioError, open, progress, a securityError.

URLLoader dáta nie sú dostupné, pokiaľ st'ahovanie nie je dokončené. Je možné sledovať stav st'ahovania dát (stiahnuté bajty a celkové bajty) pri čakaní na dokončenie požiadavky. Po stiahnutí súboru sa vyvolá udalosť o dokončení st'ahovania.

Metóda URLLoader.load() a konštruktor triedy URLLoader má jediný parameter - požiadavku, ktorá je inštanciou triedy URLRequest.

URLRequest obsahuje všetky informácie pre jednu HTTP žiadosť, ako je cieľová URL, metóda, požiadavky (GET alebo POST), MIME typ (napríklad XML) a ďalšie informácie.

---

```

var secondsUTC = new Date().time;
var dataXML = "<login>"
    + "<time>" + secondsUTC + "</time>"
    + "<username>Ernie</username>"
    + "<password>guru</password>"
    + "</login>";
var request = new air.URLRequest("http://www.example.com/login.cfm");
request.contentType = "text/xml";
request.data = dataXML;
request.method = air.URLRequestMethod.POST;
var loader = new air.URLLoader();
loader.load(request);

```

---

Výpis 18: Poslanie XML dát na server

Predchádzajúci príklad vytvorí XML inštanciu s názvom dataXML, ktorá je odoslaná na server. Ďalej je nastavená vlastnosť *contentType* z URLRequest na „text/xml“ a data, ktoré budú obsahovať XML paket. Na koniec sa vytvorí inštancia triedy URLLoader a požiadavka sa odošle na vzdialený skript pomocou metódy URLLoader.load().

Pomocou triedy *URLLoader* je možné nielen posielat' dáta na server, ale ich aj st'ahovať. Čítanie XML z externého súboru je podobné ako posielanie dát. Najskôr je potrebné vytvoriť inštanciu request triedy URLRequest so zadanou URL. Následne sa vytvorí inštancia loader triedy URLLoader. Do inštancie loader sa pridá poslucháč, pomocou ktorého URLLoader oznámi kompletne stiahnutie požadovaných dát. Nakoniec je potrebné poslať požiadavku pomocou metódy URLLoader.load ().

---

```

var request = new air.URLRequest("http://www.example.com/data.xml");
var loader = new air.URLLoader();
loader.addEventListener(air.Event.COMPLETE, completeHandler);
loader.load(request);

function completeHandler(event)
{
    var dataXML = event.target.data;
    air.trace(dataXML);
}

```

---

Výpis 19: Získanie dát z externého zdroja

Pri dynamicky generovanom obsahu je často potrebné zasielať spolu s požiadavkou aj informácie, ktoré upresňujú požadovaný obsah. Na špecifikáciu požadovaného obsahu slúži trieda **URLVariables**.

Ak sa pri vytváraní inštancie triedy `URLVariables` zadá ako parameter reťazec vo formáte meno = hodnota (napr. `var variables = new air.URLVariables("name=Franklin");`), bude vytvorená vlastnosť objektu `URLVariables` so zadaným menom a hodnotou. Tento spôsob je dobrý na vytvorenie jednej premennej, ktorá sa má poslať na server. Pri viacerých premenných je lepšie použiť nasledujúci zápis:

---

```
var variables = new air.URLVariables();
variables.name = "John";
variables.surname = "Smith";
```

---

Výpis 20: Vytvorenie viacerých premenných objektu `URLVariables`

Nasledujúci fragment kódu vytvorí objekt `URLVariables`, premenné `name` a `surname`. Následne je vytvorený objekt `URLRequest` so zadanou URL serverového skriptu, na ktorý budú zaslané premenné, nastavené metóda poslanie požiadavky (POST), pridané vytvorené premenné k požiadavke. Ako posledný je vytvorený objekt `URLLoader` s poslucháčom na úspešné prijatie požadovaných dát.

---

```
var variables = new air.URLVariables();
variables.name = "John";
variables.surname = "Smith";
var request = new air.URLRequest();
request.url = "http://www.example.com/search.php";
request.method = air.URLRequestMethod.POST;
request.data = variables;
var loader = new air.URLLoader();
loader.dataFormat = air.URLLoaderDataFormat.VARIABLES;
loader.addEventListener(air.Event.COMPLETE, completeHandler);
try
{
    loader.load(request);
}
catch (error)
{
    air.trace("Unable to load URL");
}

function completeHandler(event)
{
    air.trace(event.target.data.welcomeMessage);
}
```

---

Výpis 21: Poslanie http požiadavky s parametrami

Trieda `URLStream` ponúka na nízkej úrovni (low-level) prístup k st'ahovaným dátam. Dáta sú k dispozícii okamžite, ako sa stiahnu na miesto čakania na celé dáta ako pri `URLLoader`. **URLStream** umožňuje zatvorenie spojenia pred ukončením st'ahovania. Obsah st'ahovaného súboru je dostupný k surovým binárnym dátam (raw binary data).

### 3.5.4 Socket pripojenia

V AIR aplikácii je možné použiť dva *typy socket pripojení*: XML socket a binárne socket pripojenie.

**Socket XML** umožňuje pripojiť sa k vzdialenému serveru a vytvoriť spojenie so serverom, ktorý zostáva otvorený, pokiaľ sa nezatvorí. Toto umožňuje výmenu XML dát medzi serverom a klientom bez nutnosti neustále otvárať nové pripojenie serveru. Ďalšou výhodou použitia socket XML serveru je, že užívateľ nemusí vždy explicitne požadovať dáta. Je možné poslať dáta zo serveru bez žiadosti, a je možné posielat dáta ku každému klientovi pripojenému k serveru socket XML.

**Binárne socket pripojenie** je podobné ako XML socket s výnimkou, že klient a server si nemusia vymieňať XML pakety. Namiesto toho môže spojenie prenášať binárne informácie. Tento typ socket sa umožňuje pripojiť k širokej škále služieb, vrátane poštových serverov (POP3, SMTP, a IMAP), ale aj spravodajských serverov (protokol NNTP).

### 3.5.5 Medziaplikačná komunikácia

Pomocou triedy *LocalConnection* môžu medzi sebou komunikovať AIR aplikácie rovnako ako medzi apliáciami AIR a SWF obsahom pusteným v internetovom prehliadači.

Objekty tried z *LocalConnection* môžu iba komunikovať medzi aplikáciami AIR a SWF súbormi, ktoré sú spustené na rovnakom klientskom počítači, ale môžu bežať v rôznych aplikáciách. Dve AIR aplikácie môžu rovnako medzi sebou komunikovať ako AIR aplikácia a SWF obsah spustený v internetovom prehliadači.

Najjednoduchší spôsob ako využiť *LocalConnection* objekt, je umožniť komunikáciu iba medzi *LocalConnection* objektmi nachádzajúcimi sa v rovnakej doméne alebo rovnaké AIR aplikácie. Týmto spôsobom nebude potrebné riešiť bezpečnosť. Pokiaľ je potrebné povoliť komunikáciu medzi doménami, je viacero možností ako túto komunikáciu zabezpečiť pomocou *connectionName* parametra metódy *send()* a *allowDomain()* a doménových záznamov v *LocalConnection* triedy.

## 3.6 Distribúcia a aktualizácia aplikácie

AIR aplikácie sú distribuované ako jeden inštalačný súbor, ktorý obsahuje kód aplikácie a jej obsah. Tento inštalačný balíček môže byť distribuovaný všetkými bežnými spôsobmi - email, fyzické médiá alebo stiahnutý z internetu. Užívateľia si môžu aplikáciu nainštalovať dvojitým kliknutím na AIR súbor. Na inštaláciu aplikácie môže byť použitá bezrámová inštalácia (*seamless install feature*), ktorá umožňuje inštalovať aplikáciu priamo z internetovej stránky kliknutím na odkaz (link).

Pred distribuovaním AIR aplikácie, musí byť každá aplikácia digitálne podpísaná. Digitálny podpis poskytuje istotu, že aplikácia nebola zmenená. Pokiaľ sa aplikácia podpíše certifikátom vydaným dôveryhodnou certifikačnou autoritou, má užívateľ istotu, že aplikácia pochádza od vydavateľa aplikácie.

### 3.6.1 Inštalácia aplikácie z pracovnej plochy

Inštalračný balíček môže byť distribuovaný všetkými bežnými spôsobmi - príloha emailu, stiahnutá z internetu alebo skopírovaná z nejakého prenosového média. Po stiahnutí aplikácie stačí vykonať nasledujúce kroky:

1. Na stiahnutý balíček kliknúť dvojitým kliknutím. Na počítači musí byť nainštalovaný Adobe AIR.
2. V inštalračnom okne stačí ponechať prednastavené hodnoty a kliknúť na tlačidlo Pokračovať (Continue).

Na operačnom systéme Windows inštalračtor automaticky vykoná:

- Nainštaluje aplikácie do zložky Program Files.
- Vytvorí odkaz na spustenie aplikácie na pracovnej ploche.
- Vytvorí odkazy do Štart ponuky
- Pridá záznam na odobratie aplikácie do Pridať/ odobrať programy v Ovládacom paneli.

Na Mac OS bude aplikácia pridaná do zložky Applications.

Ak už bola aplikácia nainštalovaná, inštalračtor ponúkne užívateľovi možnosť spustenia nainštalovanej aplikácie alebo aktualizáciu stiahnutou verziou.

3. Po skončení aplikácie je potrebné kliknúť na Tlačidlo Koniec (Finish).

Na inštalračiu, aktualizáciu aplikácie musím mať používateľ potrebné systémové oprávnenia (administráčorské práva). Aplikácia môže nainštalovať novú verziu aplikácie pomocou ActionScript alebo JavaScript (Aktualizácia aplikácie).

Nainštalovaná aplikácia sa spúšťa dvojkliknutím na odkaz aplikácie, rovnakým spôsobom ako klasická desktopová aplikácia

### 3.6.2 Inštalácia aplikácie z internetovej stránky

AIR aplikácie môžu byť nainštalované na počítač užívateľa bez toho, aby boli uložené na disk a potom bol spustený inštalračtor. Na inštalračiu aplikácie priamo z internetového prehliadača musí byť na stránke SWF súbor, ktorý umožňuje inštalovanie aplikácie, inštalračiu a kontrolu potrebnej verzie Adobe AIR a spúšťanie aplikácie. Na nainštalovanie aplikácia priamo z internetového prehliadača je potrebné mať nainštalovaný Flash Player (verziu 9 update 3 alebo novšiu pre Windows a Mac OS, alebo verziu 10 pre Linux).

### 3.6.3 Inštalácie AIR aplikácie z internetového prehliadača

SWF súbor môže nainštalovať AIR aplikáciu zavolaním metódy `installApplication()` z `air.swf` súboru načítaného z `http://airdownload.adobe.com/air/browserapi/air.swf`. Keď je súbor `air.swf` načítaný, SWF súbor môže zavolať metódu `installApplication()`, ako v nasledujúcom kóde:

---

```
var url:String = "http://www.example.com/myApplication.air";
var runtimeVersion:String = "1.0";
var arguments:Array = ["launchFromBrowser"]; // Optional
airSWF.installApplication (url, runtimeVersion, arguments);
```

---

#### Výpis 22: Inštalácia AIR aplikácie z internetového prehliadača

Metóda má nasledujúce parametre:

- **url** - absolútna URL cesta k AIR inštalačnému súboru `runtimeVersion` - špecifikuje minimálnu verziu runtime požadovanú aplikáciou
- **arguments** - (voliteľné) pole argumentov, ktoré budú predané aplikácii, pokiaľ bude spustená po inštalácii. V argumentoch sú rozpoznané iba alfanumerické znaky.

#### Spustenie nainštalovanej AIR aplikácie z internetového prehliadača

Pre spustenie AIR aplikácie z internetového prehliadača je potrebné nastaviť v popisovom súbore (*descriptor file*) nasledujúce nastavenie `allowBrowserInvocation` na `true`.

SWf súbor môže spustiť aplikáciu zavolaním metódy `launchApplication()` z `air.swf` načítaného z `http://airdownload.adobe.com/air/browserapi/air.swf`. Po načítaní súboru `air.swf`, SWF súbor zavolá metódu `launchApplication()`.

---

```
var applID:String = "com.example.air.myTestApplication";
var pubID:String = "02D88EEED35F84C264A183921344EEA353A629FD.1";
var arguments:Array = ["launchFromBrowser"]; // Optional
airSWF.launchApplication(applID, pubID, arguments);
```

---

#### Výpis 23: Spustenie AIR aplikácie z internetového prehliadača

Parametre metódy `launchApplication()`:

- **appID** - indetifikátor aplikácie (Defining the application identity)
- **pubID** - indetifikátor vydavateľa aplikácie (About AIR publisher identifiers)
- **arguments** - (voliteľné) pole argumentov, ktoré budú predané aplikácii

### 3.7 Lokalizácia názvu, popisu aplikácie a inštalátora

Lokalizácia je prispôsobovanie počítačových programov pre rôzne prostredie, hlavne iné národnosti a kultúry, tak, aby bol čo najviac kompatibilný s daným prostredím. Zobrazované texty budú v miestnom jazyku a pre zobrazovanie budú používané miestne zvyklosti, napríklad miestne zvyklosti pre fyzikálne jednotky.

Adobe AIR aplikácie môžu byť lokalizované pomocou HTML Localization Framework obsiahnuté tho v súbore `AIRLocalizer.js`. Tento framework obsahuje API, ktoré pomáhajú práci s viacerými jazykmi.



### 3.7.1 Lokalizácia názvu a popisu aplikácie

Aplikácia môže mať uvedený názov a popis vo viacerých jazykoch. Názov a popis je možné nastaviť v popisovom súbore aplikácie. Pre nastavenie viacjazyčného názvu aplikácie je potrebné do elementu name vložiť element `<text>` s atribútom `xml:lang` s jazykovým kódom definovaným podľa **RFC4646** (<http://www.ietf.org/rfc/rfc4646.txt>).

Nasledujúci príklad špecifikuje názov aplikácie v troch jazykoch - angličtina, francúzština a nemčina.

---

```
<name>
  <text xml:lang="en">Sample 1.0</text>
  <text xml:lang="fr">Échantillon 1.0</text>
  <text xml:lang="de">Stichprobe 1.0</text>
</name>
```

---

Výpis 24: Lokalizácia názvu aplikácie

Rovnakým spôsobom môžeme nastaviť popis aplikácie vo viacerých jazykoch. Popis aplikácie sa nastavuje v elemente description v popisovom súbore aplikácie. Tieto nastavenia sa objavia iba v inštalačnom súbore aplikácie.

### 3.7.2 Lokalizácia HTML obsahu

Od verzie AIR 1.1 je možné lokalizovať HTML obsah aplikácie pomocou frameworku *AIRLocalizer.js*. Tento súbor obsahuje triedu *air.Localizer*, ktorá poskytuje funkcie pre pomoc pri vytváraní aplikácií s viacerými lokalizáciami.

#### Lokalizácia dátumu, času a meny

Niekedy pri tvorbe aplikácie je potrebné zohľadniť okrem jazyka aplikácie aj zobrazovanie dátumu, času alebo meny podľa miestnych štandardov. Napríklad americký štandard pre reprezentáciu dátumu je *mesiac/ deň/ rok*, zatiaľ čo európsky štandard používa zobrazenie dátumu v tvare *deň/ mesiac/ rok*.

Pomocou lokalizácie aplikácie je možné upraviť kód aplikácie tak, aby používala miestny štandard. Nasledujúci kód ukazuje zobrazenie dátumu v americkom (mesiac/ deň / rok) alebo európskom (deň/ mesiac/ rok) štandarde. Pokiaľ je premenná locale nastavená na hodnotu „en\_US“, vráti funkcia dátum vo formáte mesiac/ deň / rok. Pre všetky národné prostredia funkcia vráti dátum vo formáte deň/ mesiac/ rok.

---

```
function convertDate(date)
{
  if ( locale == "en_US" )
  {
    return (date.getMonth() + 1) + "/" + date.getDate() + "/" + date.getFullYear();
  } else {
    return date.getDate() + "/" + (date.getMonth() + 1) + "/" + date.getFullYear();
  }
}
```

---

Výpis 25: Lokalizácia dátumu

## 4 Vývojové nástroje

Pre vývoj AIR aplikácií je potrebné mať nainštalovaný AIR SDK alebo použiť vývojové prostredie s podporou pre vývoj AIR aplikácií. Existujú rozšírenia pre vývoj AIR aplikácií do vývojových prostredí Flex Builder, Dreamweaver, Flash CS3 a Aptana Studio.<sup>3</sup>

Adobe AIR SDK obsahuje nasledujúce nástroje určené pre príkazový riadok, ktoré sú potrebné na spustenie a vytvorenie aplikácie:

*AIR Debug Launcher (ADL)* umožňuje spustenie aplikácie bez inštalácie. *AIR Development Tool (ADT)* vytvorí výsledný inštalačný balíček aplikácie. Pomocou ADT sa dá vytvoriť vlastný podpisový certifikát, ktorým sa digitálne podpíše inštalačný balíček. Tieto nástroje vyžadujú, aby na počítači, na ktorom prebieha vývoj, bola nainštalovaná Java (JRE alebo the JDK 1.4 alebo novšia verzia). Java nie je potrebná na spúšťanie aplikácií na používateľovom počítači.

### 4.1 AIR Debug Launcher (ADL)

AIR Debug Launcher slúži na spúšťanie SWF a HTML založených aplikácií počas ich vývoja. Vyvíjaná aplikácia je spúšťaná bez inštalácie. ADL spúšťa runtime obsiahnutý v SDK. ADL vypisuje ladiace a chybové hlášky do štandardného výstupu, ale nepodporuje miesta prerušenia (breakpoints) alebo iné ladiace funkcie.

Spustenie ADL *adl [-runtime runtime-directory] [-pubid publisher-id] [-nodebug] application.xml [root-directory] [- arguments]*

- **runtime runtime-directory** - špecifikuje zložku runtime, ktorý bude použitý. Ak nie je zadaná zložka, použije sa runtime z SDK. V operačnom systéme Windows a Linux je potrebné určiť zložku, kde sa nachádza Adobe AIR. Na Mac OS X je potrebné zadať zložku, ktorá obsahuje AIR.framework.
- **pubid publisher-id** - priradí zadanú hodnotu ako ID vydavateľa aplikácie pre beh aplikácie. ID vydavateľa je použitý ako identifikátor aplikácie.
- **nodebug** - vypne podporu ladenia aplikácie. Ak je táto možnosť zapnutá proces aplikácie sa nemôže pripojiť na Flash debugger a hlásenia o neošetrených výnimkách sú potlačené. Sledovacie hlásenie (trace statements) bude zobrazované v konzolovom okne.
- **application.xml** - popisový súbor aplikácie
- **root-directory** - určuje koreňovú zložku aplikácie. Ak nie je zadaná použije sa zložka obsahujúca popisový súbor, ktorý bol zadaný.
- **arguments** - ľubovoľné znaky nachádzajúce sa za „—“ sú spracované ako vstupné argumenty aplikácie.

<sup>3</sup>Popis vývojových nástrojov je prevzatý z literatúry [1].

Návratová hodnota	Popis
0	Úspešné sputenie, ADL sa ukončí po ukončení AIR aplikácie.
1	Úspešné vyvolanie bežiackej AIR aplikácie. ADL bude okamžite ukončený.
2	Chyba použitia. Argumenty určené pre ADL sú neplatné.
3	Runtime nemôže byť nájdený.
4	Runtime nemôže byť spustený. Táto chyba nastáva, ak sa líši verzia runtime uvedená v aplikácii so spúšťanou verziou.
5	Neznáma príčina chyby.
6	Popisový súbor aplikácie nebol nájdený.
7	Obsah popisového súboru nie je validný. Chyba nastáva, ak nie je xml súbor správne zapísaný.
8	Hlavný súbor aplikácie (určený v elementoch <content> v popisovom súbore) nemôže byť nájdený.
9	Hlavný súbor aplikácie HTML/ SWF nie je validný.

Tabuľka 13: ADL návratové hodnoty

## 4.2 AIR Developer Tool (ADT)

Inštalačný súbor pre AIR aplikácie tvorené HTML alebo Flash obsahom sa vytvára pomocou nástroja AIR Developer Tool (ADT). Každá AIR aplikácia musí obsahovať minimálne popisový súbor a jeden súbor tvoriaci obsah aplikácie. Akékoľvek iné súčasti aplikácie musia byť zabalené v inštalačnom balíčku aplikácie.

Všetky inštalačné balíčky musia byť digitálne podpísané. Inštalátor používa podpis na overenie, či nedošlo k zmenám súborov od ich digitálneho podpísania. Môžeme použiť certifikát vydaný certifikačnou autoritou alebo vlastný podpisový certifikát.

Certifikát vydaný certifikačnou autoritou poskytuje užívateľom aplikácie uistenie o vydávateľovej identite. Vlastnoručne podpísaný certifikát nie je možné použiť na overenie identity podpisujúcej osoby. Tento nedostatok oslabuje istotu, že balíček nebol zmenený, pretože inštalačný súbor by mohol byť nahradený falzifikátom predtým, ako sa dostane k užívateľovi. Pri aktualizácii balíčka je potrebné podpísať aktualizovaný balíček tým istým certifikátom ako originálna aplikácia.

## 4.3 Vytvorenie a podpísanie inštalačného balíčka

Na vytvorenie a podpísanie inštalačného balíčka slúži aplikácia `adt` s parametrom `-package` a nasledujúcim zápisom:

```
adt -package SIGNING_OPTIONS air_file app_xml [file_or_dir | -C dir file_or_dir | -e file_dir ...] ...
```

- **air\_file** - Názov AIR súboru, ktorý bude vytvorený.
- **app\_xml** - Cesta k popisovému súboru. Cesta môže byť určená relatívnou alebo absolútnou cestou.

- **file\_or\_dir** - Cesta k súboru alebo zložke, ktorá bude pridaná do inštalačného balíčka.
- **C dir file\_or\_dir** - Zmení pracovnú zložku na zložku určenú hodnotou dir, pred spracovaním nasledujúcich súborov a zložiek do aplikačného balíčka. Súbory alebo zložky sú pridávané do hlavnej zložky balíčka. Voľbu -C je možné použiť ľubovoľne viackrát, aby boli zahrnuté súbory z viacerých miest v súborovom systéme.
- **e file dir** - Určuje cestu, kde bude uložený výsledný súbor.
- **SIGNING\_OPTIONS** - Určuje type a cestu k certifikátu. (napr. *storetype pkcs12 -keystore certificate.p12*, kde certificate.p12 je názov certifikátu.)

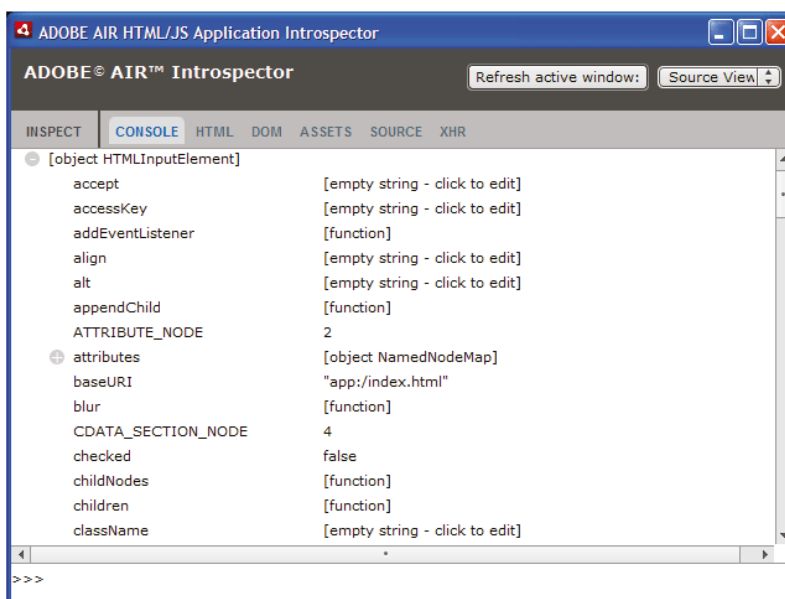
#### 4.4 Ladenie aplikácie pomocou AIR HTML Introspector

Adobe AIR HTML / JavaScript Aplikace Introspector (tzv. AIR HTML Introspector) poskytuje užitočné funkcie na ladenie HTML založenej aplikácie.

##### Načítanie AIR Introspector

AIR Introspector kód je uložený v súbore *AIRIntrospector.js*, ktorý je uložený zložke FRAMEWORK AIR SDK. Tento súbor je potrebné prekopírovať do zložky aplikácie a načítať externý skript AIRIntrospector.js v hlavičke hlavného HTML súboru aplikácie.

Odkaz na AIR HTML Introspector je potrebné odstrániť pred zostavením finálnej verzie aplikácie, ktorá sa bude publikovať.






Obr. 2: AIR HTML Introspector

V trieda Console je definovaných 5 metód: *log()*, *warn()*, *info()*, *error()*, a *dump()*. Metódy *log()*, *warn()*, *info()*, *error()* umožňujú zobrazit' vlastností objektu v okamihu

Parameter	Popis
dumpObject	Objekt, ktorý bude serializovaný.
levels	Maximálny počet úrovní, ktoré majú byť skúmané v objekte strome okrem koreňovej úrovne. Prednastavená hodnota je 1. Tento parameter je voliteľný.

Tabuľka 14: Vstupné parametre ladiacich metód

Metóda	Ikona
Info()	
Error()	
Warn()	

Tabuľka 15: Ikony ladiacich metód

prehliadania. Na serializáciu objektu slúži metóda `dump()`. Táto metóda má dva vstupné parametre (tabuľka č. 14).

Jednotlivé metódy sú v konzole odlišené pomocou nasledujúcich ikon:

### Záložka Console

Záložka Console zobrazuje hodnoty a vlastnosti objektu, ktorý je predávaný ako parameter nejakej metódy triedy `air.Introspector.Console`.

### Záložka HTML

HTML záložka umožňuje zobrazit' celý HTML DOM v stromovej štruktúre. Po kliknutí na prvok sa zobrazia jeho vlastnosti na pravej strane karty. Atribúty jednotlivých elementov je možné meniť, zmeny sa prenesú do aplikácie okamžite.

### Záložka DOM

Záložka DOM zobrazuje objekt v stromovej štruktúre. Je tu možné upravovať a meniť ľubovoľný reťazec. Zmeny sa prejaví v aplikácii.

### Záložka Assets

Záložka Assets umožňuje skontrolovať odkazy, obrázky, CSS a JavaScript súbory načítané v okne aplikácie. Po otvorení uzla sa zobrazí obsah súboru alebo aktuálne použitý obrázok.

### Záložka Source

Obsahuje 3 časti:

- *Actual source* - Zobrazí HTML kód stránky, ktorá sa načíta po spustení aplikácie.

- *Parsed source* - zobrazí značky aplikácie, ktoré tvoria užívateľské rozhranie aplikácie, ktoré sa môžu líšiť od skutočného zdroja, pretože aplikácia generuje tagy za behu aplikácie pomocou JavaScript-u (AJAX-u).
- *Application files* - Vypíše súbory v zložke aplikácie. Tento zoznam je k dispozícii pre AIR Introspector pri spustení v bezpečnostnom prostredí (sandbox).

### Záložka XHR

Na tejto záložke sú zobrazené všetky XMLHttpRequest komunikačné žiadosti a zaznamenané údaje. Toto umožňuje zobraziť XMLHttpRequest vlastnosti vrátane responseText a responseXML (ak je k dispozícii) v stromovom zobrazení.

## 4.5 Vytvorenie HTML založenej aplikácie

Každá HTML založená AIR aplikácia musí obsahovať minimálne nasledujúce súbory:

- Popisový súbor (descriptor file), ktorý obsahuje metadata aplikácie
- top-level HTML stránku, ktorá "spúšťa" aplikáciu
- JavaScriptový AIRAliases.js s AIR API triedami

### 4.5.1 Popisový súbor AIR aplikácie

Na začiatku tvorby aplikácie je potrebné vytvoriť popisový XML súbor s nasledujúcou štruktúrou:

---

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/1.5">
  <id>... </id>
  <version>... </version>
  <filename>... </filename>
  <initialWindow>
    <content>... </content>
    <visible>... </visible>
    <width>... </width>
    <height>... </height>
  </initialWindow>
</application>
```

---

Výpis 26: Popisový súbor AIR aplikácie

`<application>` špecifikuje, ktorá verzia runtime sa použije. V ukázkovom príklade sa jedná o verziu 1.5.

`<id>` jedinečný identifikátor aplikácie s identifikátorom vydávateľa aplikácie. Odporúčaný tvar je reverzný zápis DNS "com.company.AppName". Id aplikácie používame pri inštalácii aplikácie, pri prístupe k súkromným dátam a súborom, medzi aplikačnou komunikáciou.

Element `<version>` pomáha užívateľom verziu aplikácie, ktorú inštalujú.

Element `<filename>` určuje názov, ktorý sa použije na pomenovanie spúšťačného súboru,

zložky, kde sa aplikácia nainštaluje a iných referencií v operačnom systéme.

Element `<initialWindow>` špecifikuje nastavenia okna aplikácie pomocou nasledujúcich potomkov:

`<content>` Identifikuje hlavný HTML súbor aplikácie, ktorý sa načíta po jej spustení.

`<visible>` nastaví viditeľnosť okna aplikácie.

`<width>` nastaveí šírku okna v pixeloch.

`<height>` nastaví výšku okna v pixeloch.

Ukážka popisového súbor HelloWorld-app.xml s povinnými elementami:

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://ns.adobe.com/air/application/1.5">
  <id>examples.html.HelloWorld</id>
  <version>0.1</version>
  <filename>HelloWorld</filename>
  <initialWindow>
    <content>HelloWorld.html</content>
    <visible>true</visible>
    <width>400</width>
    <height>200</height>
  </initialWindow>
</application>
```

Výpis 27: Popisový súbor AIR aplikácie HelloWorld

#### 4.5.2 Vytvorenie HTML stránky aplikácie

HTML stránka aplikácie musí byť tvorená nasledujúcou štruktúrou HTML elementov:

```
<html>
  <head>
    <title></title>
  </head>
  <body>
  </body>
</html>
```

Výpis 28: Štruktúra HTML dokumentu

`<html>` a `</html>` označujú začiatok a koniec HTML dokumentu.

`<head>` a `</head>` označujú oblasť hlavičky stránky, ktorá sa nezobrazuje, ale obsahuje názov okna aplikácie, odkazy na kaskádové štýly a JavaScriptové súbory.

`<title>` a `</title>` označujú názov okna aplikácie, môže sa líšiť od názvu aplikácie.

`<body>` a `</body>` určujú obsah aplikácie, čo sa bude zobrazovať v okne.

```
<html>
<head>
  <title>Hello World</title>
  <script type="text/javascript" src="AIRAliases.js"></script>
```

```
<script type="text/javascript">
  function appLoad(){
    air.trace("Hello World");
  }
</script>
</head>
<body onLoad="appLoad()">
  <h1>Hello World</h1>
</body>
</html>
```

Výpis 29: HelloWorld.html

### 4.5.3 Testovanie aplikácie

Na spustenie testu aplikácie slúži aplikácia v príkazovom riadku AIR Debug Launcher (ADL). Spustiteľný súbor ADL sa nachádza v zložke bin Adobe AIR SDK. Testovanie sa spustí v príkazovom riadku príkazov *adl HelloWorld-app.xml*.

### 4.5.4 Vytvorenie inštalačného súboru

Na vytvorenie inštalačného súboru aplikácie slúži nástroj ADT z Adobe AIR SDK. AIR inštalačný súbor obsahuje všetky súbory, ktoré tvoria aplikáciu. Na zaistenie bezpečnosti všetky AIR inštalačné súbory musia byť digitálne podpísané. Pre vývojové účely je možné vygenerovať základný vlastný podpisový (self-signed) certifikát pomocou ADT alebo iného nástroja pre generovanie certifikátov. Pri inštalácii aplikácie, ktorá bola podpísaná vlastným certifikátom je zobrazovaný neznámy (unknown) vydavateľ aplikácie. Vlastný certifikát garantuje iba to, že súbory aplikácie neboli zmenené od vytvorenia inštalačného súboru. Pre konečné zverejnenie inštalačného súboru sa odporúča použitie komerčného certifikátu.

## 4.6 Vytvorenie certifikátu

Pre vývojové účely sa dá pomocou aplikácie ADT vytvoriť vlastný podpisový certifikát. Na vytvorenie certifikátu slúži *adt -certificate -cn SelfSigned 1024-RSA sampleCert.pfx samplePassword*.

Argumenty programu sú:

- *sampleCert.pfx* - názov výstupného súboru
- *samplePassword* - heslo, ktoré chráni certifikát

Tento príklad používa minimálny počet atribútov, ktoré môžu byť pri tvorbe certifikátu. Typ kľúča musí byť 1024-RSA alebo 2048-RSA.



## 4.7 Vytvorenie inštalačného súboru

Inštalačný súbor AIR aplikácií sa vytvára pomocou nástroja `adt -package -storetype pkcs12 -keystore sampleCert.pfx HelloWorld.air HelloWorld-app.xml HelloWorld.html AIRAliases.js`.

Po zadaní potrebných vstupných parametrov je potrebné zadať heslo k podpisovému certifikátu. `HelloWorld.air` určuje názov výstupného inštalačného súboru, `HelloWorld-app.xml` je popisový súbor aplikácie. Nasledujúca postupnosť argumentov je zoznam súborov, ktoré tvoria aplikáciu. Pri tvorbe inštalačného súboru môže byť použitý ľubovoľný počet súborov a zložiek.

Inštalačný súbor môže byť spustený dvojklikom na súbor `HelloWorld.air`, alebo zadaním jeho názvu do príkazového riadku.

## 5 Popis portálu a klientskej aplikácie

Serverová a administrátorská časť geografického informačného portálu je postavená na technológií ASP.NET verzie 3.5 s databázovým serverom Microsoft SQL Server 2005.<sup>4</sup>

Klientska časť využíva technológiu Adobe AIR. Aplikácia je tvorená pomocou technológií určených pre internetové stránky **HTML, CSS, AJAX (Javascript)**. Pre mapové podklady bola zvolená internetová služba *Google maps*. [8]

Geografický informačný portál slúži na zdieľanie bodov záujmu (tzv. POI - Point of interest), prejdených GPS trás a fotografií. Jednotlivé geografické informácie a fotky sú usporiadané v kategóriách.

Portál rozlišuje 2 typy užívateľov, *užívateľ* a *administrátor*. Užívateľ môže po zaregistrovaní sa na portáli pristupovať a vytvárať zdieľaný obsah. Pridané body záujmu musia byť pred zverejnením schválené administrátorom. Užívateľ nemá prístup do administratívnej zóny portálu. Administrátor môže pristupovať k zdieľanému obsahu, pridávať nový obsah na portál, schvaľovať novopridané body záujmu a určovať nových administrátorov.

### 5.1 Klientska aplikácia

Klientska aplikácia je vytvorená pomocou technológií určených pre internetové stránky *HTML, CSS, AJAX (Javascript)*. HTML slúži na vytvorenie základného vzhľadu a obsahu aplikácie. Pomocou CSS je upravený vzhľad aplikácie do konečnej podoby. Klientska aplikácia komunikuje so serverovou časťou pomocou HTTP požiadaviek POST zasielaných na serverové ASP.NET HTTP Handler-y (ashx stránka). Získaný obsah je stiahnutý do aplikácie v textovej forme a následne pomocou JavaScriptu zobrazený. Na prihlasovacej obrazovke a na zobrazenie väčších náhľadov zdieľaných fotografií bol použitý Ajaxový framework Adobe Spry. Všetky vstupy zasielané užívateľom sú validované na strane klientskej aplikácie a aj na strane servera.

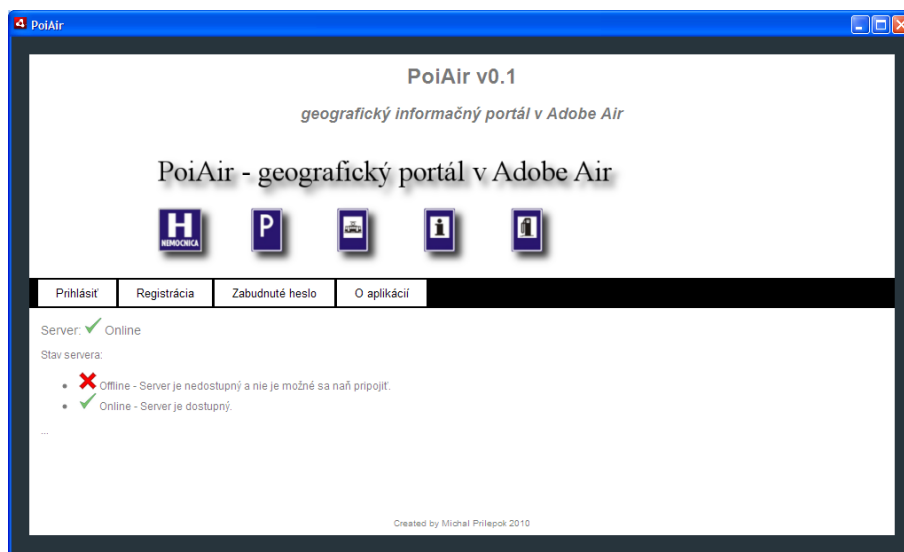
Aplikácia sa spúšťa v samostatnom okne. Toto zobrazenie eliminuje optimalizáciu, úpravu vzhľadu a JavaScriptového kódu aplikácie pre každý internetový prehliadač. HTML obsah je zobrazovaný pomocou zobrazovacieho jadra WebKit. Týmto je zaistený jednotný vzhľad a funkčnosť na rôznych operačných systémoch, ktoré sú podporované frameworkom Adobe AIR.

V aplikácii je možné pridávať nový obsah na portál, zobrazovať obsah portálu - body záujmu, GPS trasy, fotografie a editovať užívateľský profil.

Aplikácia obsahuje hlavičku, ktorá je jednotná, trvalé menu s odkazmi na zobrazenie kategórii bodov záujmu, GPS trás, Fotografií a užívateľského profilu. Pod touto ponukou sa zobrazuje ponuka pre zvolenú časť aplikácie.

Znovu načítanie dát slúži na opätovné zaslanie požiadavky na server, ak nastala pri predchádzajúcom poslaní požiadavky chyba.

<sup>4</sup>V tejto kapitole je použitá literatúra [3, 4, 6, 7, 8].



Obr. 3: poiAir klientska aplikácia

### 5.1.1 Body záujmu

Po úspešnom prihlásení sa zobrazí stránka *poiKategoria.html* so zoznamom kategórií bodov záujmu. Dáta pre zoznam kategórií sú stiahnuté automaticky po zobrazení stránky zavolaním funkcie `load()` alebo kliknutím na odkaz POI zoznam.

Zoznam kategórií je zatriedený abecedne vzostupne podľa názvu kategórie. Každá kategória má vlastnú ikonu. Názov kategórie môže byť dlhý až 50 znakov a môže obsahovať iba znaky malej a veľkej anglickej abecedy s číslami a medzerou. Ku kategórii môžeme doplniť stručný popis, ktorý môže byť dlhý maximálne 100 znakov. Ikony kategórií sú sťahované zo serveru v JPEG formáte.

Po kliknutí na názov kategórie sa nám zobrazí stručný zoznam bodov záujmu vo zvolenej kategórii. Zoznam bodov záujmu je možné preniesť do externej aplikácie alebo navigačného zariadenia pomocou odkazu `Stiahnuť GPX`. Po kliknutí na tento odkaz bude vygenerovaný súbor vo formáte GPS Exchange Format (GPX). Tento formát je veľmi dobre podporovaný aj v starších aplikáciách, ktoré nepodporujú KML formát.

KML formát je od 14. apríla 2008 oficiálny formát pre prenos geografických informácií. Tento formát je spravovaný štandardizačným združením Open Geospatial Consortium. Tento formát bol pôvodne vyvinutý pre program Google Earth. Slúži na pridávanie užívateľských dát na základnú vrstvu s leteckými snímkami. [9]

### Zoznam bodov záujmu v kategórii

Zoznam bodov je usporiadaný podľa id bodu záujmu vzostupne. Najnovší bod záujmu sa zobrazí vždy na vrchu zoznamu. Tento stručný zoznam obsahuje iba názov bodu záujmu, zemepisnú dĺžku, zemepisnú šírku, mesto, adresu a poznámku.

Časť aplikácie	Doplnková ponuka
POI kategórie	<ul style="list-style-type: none"> <li>• Nová POI kategória</li> <li>• Nový POI</li> <li>• Odkaz pre návrat z detailu POI do kategórie</li> <li>• Znovu načítanie dát</li> </ul>
GPS trasy	<ul style="list-style-type: none"> <li>• Nová kategória</li> <li>• Nová trasa</li> <li>• Odkaz pre návrat z detailu trasy do kategórie</li> <li>• Znovu načítanie dát</li> </ul>
Fotky	<ul style="list-style-type: none"> <li>• Nový album</li> <li>• Pridať fotku</li> <li>• Znovu načítanie dát</li> </ul>
Môj profil	Neobsahuje

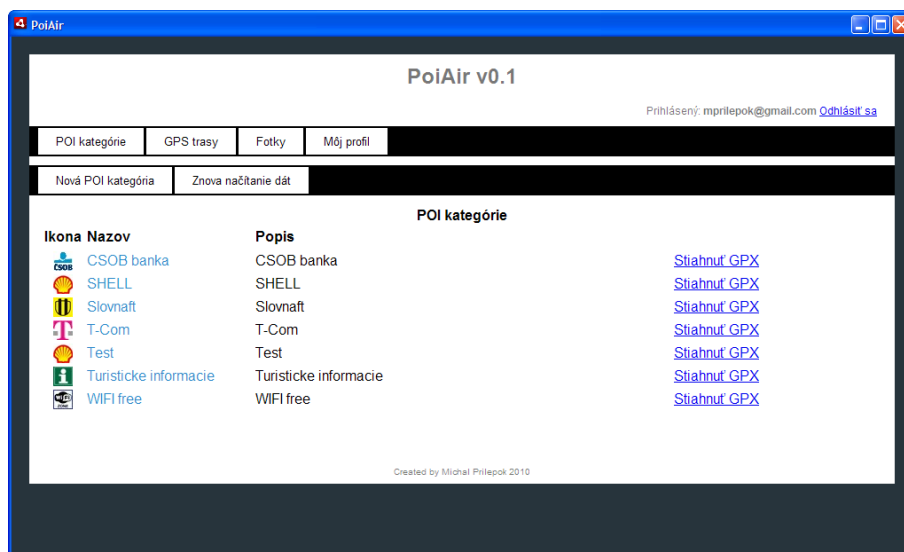
Tabuľka 16: Menu aplikácie

GPS súradnice bodu sa zobrazujú vo formáte **DD** - decimálne stupne, napr. (49.5000°N, 23.5000°E). Po prejdení kurzorom myši nad zemepisnú dĺžku alebo šírku sa zobrazí malá bublinková nápoveda s ďalšími dvoma formátmi súradníc. V nápovede sa zobrazí formát GPS súradníc **DM** (stupne:decimálne minúty, napr. : 49°30.0'N, 23°30.0'E) a **DMS** (stupne:minúty:sekundy, napr. 49°30'00"N, 23°30'00"E).

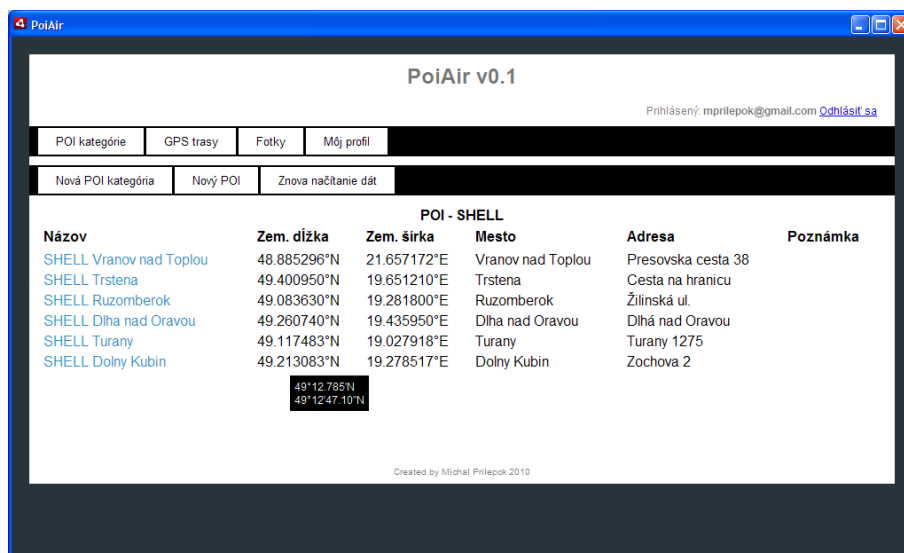
Detaily zvoleného bodu záujmu môžeme zobrazit' po kliknutí na názov bodu záujmu. V detailoch sa zobrazia všetky informácie o zvolenom bode záujmu. V detailoch sa nachádza názov, zemepisná dĺžka a šírka vo všetkých 3 formátoch GPS súradníc (DD, DM, DMS), mesto, adresa, odkaz na internetovú stránku, číslo telefónu, poznámka a mapa so zobrazeným bodom.

Každý zadaný bod záujmu musí obsahovať minimálne názov, zemepisnú dĺžku a šírku. Ostatné údaje sú nepovinné.

Mapa je interaktívna. Dá sa zmeniť jej mierka, presúvať zobrazená oblasť a meniť typ mapy. Na výber je mapa v štýle klasickej papierovej mapy, letecké snímky, digitálny

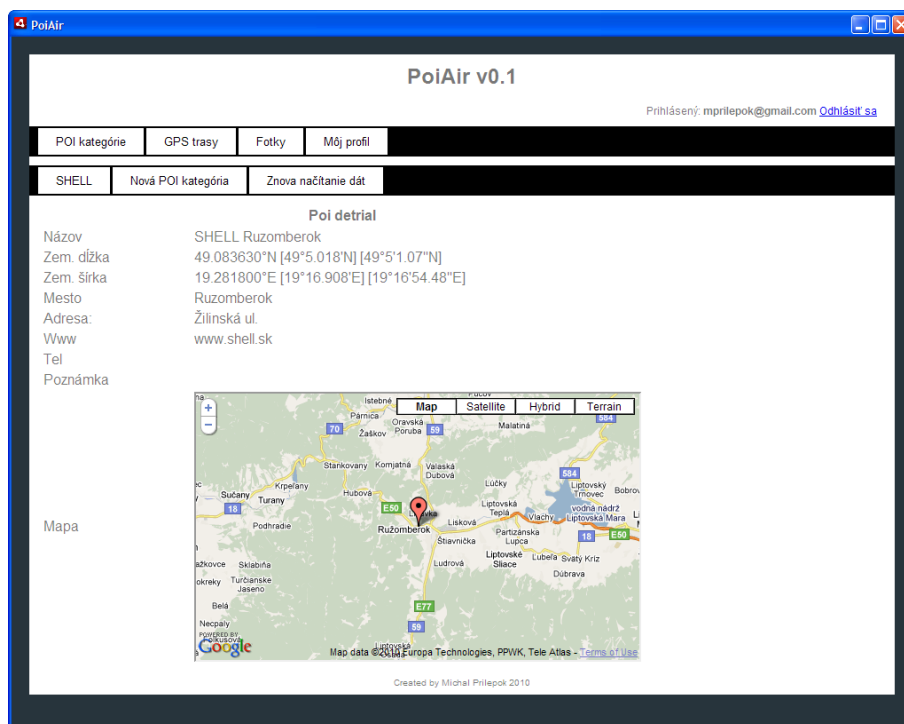


Obr. 4: Zoznam kategórií POI



Obr. 5: Zoznam POI v kategórií

„3D“ model terénu alebo hybridné zobrazenie cestnej mapy a leteckých snímok. Mapové podklady sú načítavané pomocou Google maps API z internetu.



Obr. 6: Detail POI

### Nová kategória

Pre každú novú kategóriu, ktorú chceme vytvoriť, je potrebné zadať jej názov a vybrať číslo ikony zo zoznamu dostupných ikon. Do názvu kategórie môžeme zadať iba povolené znaky a-zA-Z0-9 a medzeru. Dĺžka názvu je obmedzená na 50 znakov. Dĺžka a znaky popisu sú tiež obmedzené. Maximálna dĺžka je nastavená na 100 znakov a môže byť tvorená iba znakmi.

Zoznam dostupných ikon sa zobrazí v novom okne po kliknutí na odkaz uvzoznam dostupných ikon. Ikonky sú zobrazené v tabuľke po 10 ikonách v riadku. Nad každou ikonou sa nachádza číslo ikony. Toto číslo ikony je potrebné zadať do textového poľa formuláru. Nové ikony môže pridať iba administrátor cez administrátorské rozhranie na serveri.

### Nový bod záujmu

Nový bod záujmu je možné pridať do existujúcej kategórie. Pre novopridávaný bod záujmu je potrebné zadať jeho názov a zemepisnú dĺžku a šírku. Ostatné údaje ako sú adresa, mesto, telefón, odkaz na internetovú stránku a poznámka sú voliteľné.

Názov môže obsahovať znaky a-zA-Z0-9 a medzeru. Dĺžka je obmedzená na 50 znakov. Zemepisná dĺžka a šírka sa zadávajú vo formáte DD (desatinné stupne). Napríklad 49.000000 pre zemepisnú dĺžku a 19.000000 pre zemepisnú šírku. Kladná hodnota zemepisnej dĺžky určuje miesto na severnej pologuli a záporná na južnej. Kladná hodnota zemepisnej šírky určuje miesto na východ od nultého poludníka (Greenwichský poludník) a záporná miesto na západ od nultého poludníka. Po zadaní názvu a GPS súradníc sa automaticky aktualizuje náhľad mapy. Pomocou náhľadu na mape môžeme skontrolovať správnosť zadaných súradníc. Zadaný bod sa na mape zobrazí červenou značkou.

The screenshot shows a web browser window titled 'PoiAir v0.1'. The page has a navigation menu with 'POI kategórie', 'GPS trasy', 'Fotky', and 'Mój profil'. The current page is 'Nová POI kategória'. The main content area is titled 'Pridanie nového POI' and contains a form with the following fields:

- Názov:
- Zem. dĺžka:
- Zem. šírka:
- Adresa:
- Mesto:
- Tel:
- Www:
- Poznámka:

Below the form is a map preview with a red pin and a 'Map' dropdown menu. A 'Pridat' button is located at the bottom left of the form area. The footer of the page reads 'Created by Michal Fulecek 2010'.

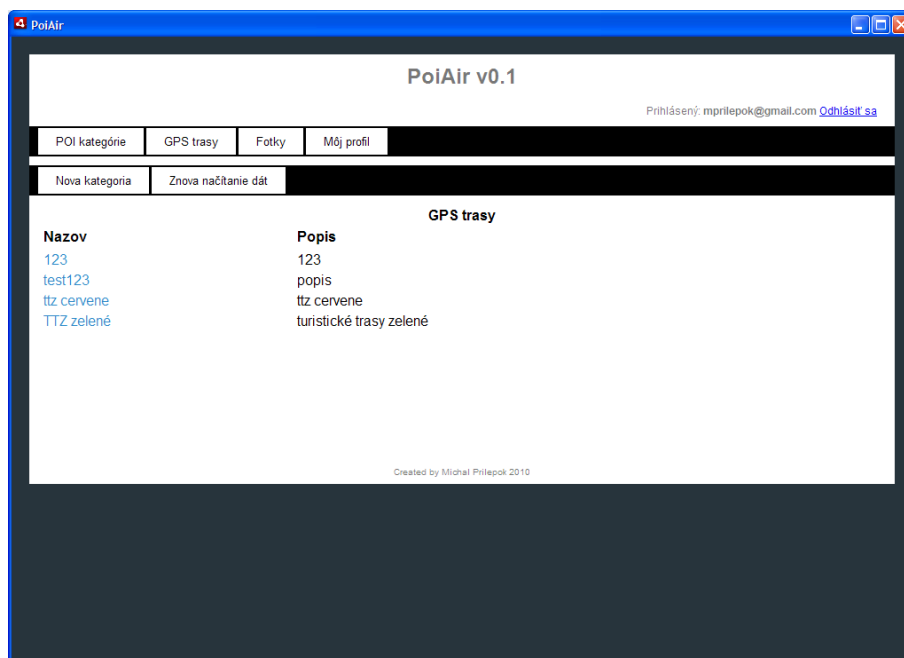
Obr. 7: Formulár pre pridanie POI

### GPS trasy

GPS trasy sú usporiadané v kategóriách. Každá trasa musí byť zaradená lenv jednej kategórii. Zoznam kategórií je automaticky stiahnutý so servera. Kategórie sú usporiadané vzostupne podľa názvu. Každú kategóriu môžeme doplniť o krátky 100 znakový popis. Po kliknutí na názov kategórie sa nám stiahne a zobrazí zoznam jednotlivých GPS trás vo zvolenej kategórii.

### Nová kategória

Na vytvorenie novej kategórie musíme zadať len je názov. Názov je obmedzený ako v aj iných častiach na 50 znakov a znaky a-zA-Z0-9 a medzeru. Popis kategórie je voliteľný. Platia tu rovnaké obmedzenia ako na všetky popisy kategórií v aplikácii. Po kliknutí na tlačidlo pridať sa vykoná kontrola vstupu pomocou regulárnych výrazov. Akákoľvek chyba bude



Obr. 8: Kategorie GPS trás

oznámená užívateľovi. Údaje z formulára sa pošlú na server. Následne budeme informovaní o úspešnom pridaní kategórie. Po potvrdení oznámenia aplikácia zobrazí aktualizovaný zoznam kategórií.

Tlačidlom Vymazať môžeme zmazať všetky údaje z formulára. Nová kategória nebude vytvorená.

### Pridanie novej trasy

Novú GPS trasu môžeme pridať len do zvolenej kategórie. K pridávanej trase je potrebné vyplniť jej názov, miesto, zvoliť súbor so zaznamenanou trasou. Zaznamenaná trasa môže byť vo formáte GPX.

Po kliknutí na tlačidlo Pridať, aplikácia skontroluje vstupy zadané užívateľom pomocou regulárnych výrazov. Po overení správneho formátu údajov ich aplikácia zašle na server vytvorením http požiadavky s parametrami a pripojeným súborom. O stave posielania súboru budeme informovaní textom uvNahrávam ... s informáciou o stave poslania súboru v percentách. Po úspešnom nahratí trasy na server budeme informovaní a následne presmerovaní na aktualizovaný zoznam trás vo zvolenej kategórii.

### Detaily GPS trasy

Detaily zvolenej trasy zvolíme kliknutím na názov trasy. V detailoch trasy uvidíme jej názov, miesto, kde sa trasa nachádza a popis. Pomocou tlačidla stiahnuť si ju môžeme uložiť do počítača a s ňou pracovať podľa potreby. Trasa bude uložená vo formáte GPX.



### 5.1.2 Fotografie

Na geografický informačný portál môžeme nahrávať aj fotografie a organizovať ich do albumov. Ide o jednoduchú galériu. V aplikácii môžeme vytvárať len nové albumy a nahrávať do nich fotografie.

O zobrazenie zoznamu albumov sa stará stránka *fotkaKategorie.html*. Zoznam albumov je realizovaný len v textovej forme. Albumy sú usporiadané podľa abecedy vzostupne. V zozname albumov sa nachádza názov albumu a jeho popis, stručná charakteristika. Po kliknutí na názov albumu sa nám zobrazí prehľad fotografií zvoleného albumu.

Jednotlivé fotky sú usporiadané do tabuľky po dvoch fotkách v riadku. Každá fotografia je zmenšená na šírku 300 pixelov. Výška fotografie je automaticky dopočítaná tak, aby bol zachovaný pomer strán. Pod fotografiou je umiestnený jej názov. Pod fotografiou sa zobrazuje meno autora, ktorý ju nahral do albumu. Ide o zobrazované meno z profilu užívateľa.



Obr. 9: Zobrazenie fotoalbumu

Po presunutí kurzoru na obrázok sa zobrazí jeho zväčšený náhľad. Náhľad obrázku je 600 pixelov. Výška obrázku sa opäť automaticky dopočíta, aby bol zachovaný pomer strán originálnej fotografie.

Zväčšený náhľad je realizovaný pomocou widgetu Tooltip z Ajaxového frameworku **Adobe Spry**. Náhľad je presúvaný v smere pohybu kurzora myši. Pri zobrazení alebo ukrytí náhľadu je použitý efekt postupného zobrazenia alebo skrytia.

## Pridanie fotografie do albumu

Pri pridávaní fotografie do albumu je potrebné zadať názov fotografie a zvoliť súbor. Meno autora sa získa zo šifrovaného uložiska aplikácie, kde sa uloží po úspešnom prihlásení. Názov fotografie s spolu fotografiou je poslaný po úspešnej validácii na server. O stave posielania fotografie budeme informovaní správou „Nahrávam...“ a stavom posielania fotografie vyjadreným percentami. Do galérie môžeme nahrávať iba fotky vo formáte JPEG.

### 5.1.3 Profil používateľa

Každý používateľ si môže zobrazit' alebo upraviť svoj profil pomocou odkazu „Môj profil“. Po kliknutí na odkaz sa automaticky načítajú informácie o používateľovi. Tieto informácie je možné upravovať. Upraviť sa nedá len členstvo v skupine. Používateľ si tu môže zmeniť email, mesto, zobrazované meno a heslo.

Obr. 10: Zobrazenie profilu používateľa

## 5.2 Serverová časť

Serverová časť portálu je rozdelená na 3 časti. V prvej časti sa nachádzajú verejné informácie, kde sa nachádza registračný formulár, formulár pre obnovenie zabudnutého hesla, návody a postupy, stránka s odkazmi na stiahnutie klientskej aplikácie a Adobe AIR frameworku a zoznam najčastejších otázok s odpoveďami.

V druhej časti sú uložené ASP.NET HTTP Handler-y pre komunikáciu serverovej časti s klientskou aplikáciou. Tieto Handler-y sú umiestnené v podložke Aplikacia.

Posledná časť portálu je určená administrátorom. V tejto časti môžu kontrolovať a upravovať celý obsah portálu. Administrátorská časť je umiestnená v zložke Admin.



Obr. 11: Zobrazenie úvodnej stránky portálu

### 5.2.1 Databáza

Všetky dáta okrem ikon pre kategórie bodov záujmu sú uložené v databáze. Databáza obsahuje nasledujúce tabuľky: Všetky operácie na databáze sa vykonávajú pomocou uložených procedúr v SQL serveri.

### 5.2.2 ASP.NET HTTP Handler-y pre klientsku aplikáciu

Všetky potrebné ASP.NET Handler-y sú uložené v zložke Aplikácia. Pred začatím používania aplikácie sa musí každý používateľ prihlásiť. Klientska aplikácia pošle prihlasovací email a SHA256 hash hesla na handler Login.ashx. Handler zvaliduje vstupy pomocou regulárnych výrazov. Ak nebol poslaný email v správnom formáte alebo hash hesla nemá správnu dĺžku alebo obsahuje nepovolené znaky, handler pošle klientovi chybovú správu. Ak sú vstupy validné, server sa pokúsi vyhľadať používateľa v databáze pomocou uloženej procedúry UserFind. Ak sa používateľa podarí nájsť, aplikácia zašle klientskej aplikácii id nájdeného používateľa. Ak sa používateľa nepodarí nájsť, server pošle chybovú správu.

Názov tabuľky	Popis
group	Obsahuje informácie o skupinách používateľov.
passwordRequest	Obsahuje informácie o obnove zabudnutého hesla.
photo	Obsahuje informácie o fotografiách.
photoCategory	Obsahuje informácie o kategóriách fotografií.
poi	Obsahuje informácie o bodov záujmu.
poiCategory	Obsahuje informácie o kategóriách bodov záujmu.
track	Obsahuje informácie o GPS trasách.
trackCategory	Obsahuje informácie o kategóriách GPS trás.
user	Obsahuje základné informácie o používateľoch.
userDetails	Obsahuje detaily o používateľoch.
group	Obsahuje informácie o skupinách používateľov.
passwordRequest	Obsahuje informácie o obnove zabudnutého hesla.
photo	Obsahuje informácie o fotografiách.
photoCategory	Obsahuje informácie o kategóriách fotografií.
poi	Obsahuje informácie o bodov záujmu.
poiCategory	Obsahuje informácie o kategóriách bodov záujmu.
track	Obsahuje informácie o GPS trasách.
trackCategory	Obsahuje informácie o kategóriách GPS trás.
user	Obsahuje základné informácie o používateľoch.
userDetails	Obsahuje detaily o používateľoch.

Tabuľka 17: Databázové tabuľky

### Body záujmu

O vytvorenie zoznamu kategórií POI sa stará Kategoria.ashx handler. Handler sa pripojí na databázu a získa z nej zoznam kategórií POI. Tento zoznam je zoradený vzostupne podľa názvu kategórie. Na začiatok textovej odpovede je pridaný reťazec „OK:“, ktorý predstavuje úspešné získanie dát. Za týmto reťazcom nasleduje zoznam kategórií naformátovaný do reťazca id kategórie; názov ikony; názov kategórie; popis kategórie. Jednotlivé kategórie sú od seba oddelené ukončením riadka \n.

---

```
OK:6;10.jpg;CSOB SR;banka
8;6.jpg;OMW SR;cerpacie stanice
```

---

Výpis 30: Zoznam kategórií POI

O výpis zoznamu POI v kategórii sa stará handler ZoznamPoi.ashx. Tento handler má jediný vstupný parameter a tým je identifikátor kategórie. Tento vstupný parameter je skontrolovaný pomocou regulárneho výrazu  $\text{^}([0-9]\{1,6\})\text{\$}$ . Následne sú získané dáta z databázy zavolaním uloženej procedúry Listpoi. Na začiatok textovej odpovede je pridaný reťazec „OK:“, ktorý predstavuje úspešné získanie dát. Výsledné dáta sú naformátované do textového reťazca id; názov poi; zem. dĺžka; zem. šírka; mesto; adresa; odkaz na internetovú

stránku; telefón; popis. Jednotlivé záznamy o POI sú od seba oddelené ukončením riadka \n.

---

```
OK:11;SHELL Povazska Bystrica;49,102939;18,438119;Povazska Bystrica;Sladkovicova; Otvorene
nonstop;www.shell.sk;
5;SHELL Trstena;49,400950;19,651210;Trstena;Cesta na hranicu;Otvorene nonstop;www.shell.sk;
```

---

#### Výpis 31: Výpis zoznamu POI pre zvolenú kategóriu

Detaily poi sa získavajú pomocou handlera Detailpoi.ashx. Tento handler má jeden vstupný parameter, ktorý je použitý ako identifikátor bodu záujmu. Vstupný parameter je kontrolovaný rovnakým regulárnym výrazom ako identifikátor kategórie bodov záujmu. Na získanie detailov z databázy je zavolaná procedúra PoiDetail. Detaily poi sú naformátované do reťazca id; názov, zem. dĺžka; zem. šírka; popis, odkaz na internetovú stránku; mesto; telefón; adresa.

---

```
OK:2;SHELL Turany;49,117483;19,027918;;www.shell.sk;Turany; Otvorene nonstop;Turany
```

---

#### Výpis 32: Detail POI

Pridanie novej kategórie a pridanie nového bodu záujmu funguje podobne. V oboch prípadoch sú všetky vstupy kontrolované pomocou regulárných výrazov. Pre vytvorenie novej kategórie je zavolaná procedúra PoiCategoryAdd v handleri PoiCategoryAdd.ashx so vstupnými parametrami názov, ikona a popis kategórie. V handleri Poipridaj.ashx sú vstupnými parametrami názov, identifikátor kategórie, zemepisná dĺžka, zemepisná šírka, adresa, mesto, telefón, url adresa na internetovú stránku a poznámka. Uložená procedúra PoiAdd po skontrolovaní vstupných dát priradí nový poi do databázy. Úspešné uloženie dát do databázy je signalizované návratovým reťazcom „OK“ a chybné stavy „ERROR“.

### 5.2.3 GPS trasy

Handler-y starajúce sa o sprostredkovanie dát GPS trás pre klientsku aplikáciu sú:

#### TrackCategoryList.ashx

Tento handler nemá žiadne vstupné parametre. Z databázy získa zoznam kategórií, do ktorých môžu byť usporiadané GPS trasy. Zoznam kategórií je usporiadaný podľa abecedy vzostupne.

Spolu s názvom kategórie je prenášaný aj jej identifikátor id. Na začiatok správy je pridaný text „OK:“. Tento text symbolizuje úspešné získanie dát z databázy a aplikácia ich môže spracovať. V prípade akejkoľvek chyby bude vrátená správa obsahovať text „ERROR:“. Táto správa symbolizuje chybu a aplikácia nespracuje a nezobrazí žiadne dáta zo serveru.

---

```
OK:2;TT cervene SR;turisticke trasy cervene
1;TT zelene SR;turisticke trasy zelene
6;TT zlte SR;turisticke trasy zlte
```

---

#### Výpis 33: Výpis zoznamu kategórií GPS trás

### TrackList.ashx

Tento handler má jeden vstupný parameter idCategory. Vstupný parameter je kontrolovaný pomocou regulárneho výrazu. Môže obsahovať iba čísla o dĺžke 1 až 6 znakov. Po skontrolovaní vstupného parametra sa zavolá uložená procedúra TrackList. Výstupný zoznam kategórií je naformátovaný do riadkov, každá kategória je v samostatnom riadku oddelená znakom \n. Jednotlivé atribúty sú usporiadané v nasledujúcom poradí: id, názov, lokalita, popis.

---

```
OK:1;Trasa 1; lokalita 1;popis 1
2;Trasa 2; lokalita 2;popis 2
```

---

Výpis 34: Výpis zoznamu GPS trás v kategórií

### TrackDetail.ashx

O spracovanie detailov sa stará handler TrackData.ashx. Tento handler vyžaduje na vstup jeden parameter id, ktorý identifikuje detaily požadovanej trasy. Tento identifikátor tak isto je kontrolovaný pomocou regulárneho výrazu na správny formát. Identifikátor môže obsahovať iba číslice 0 - 9, dĺžka identifikátora môže byť od jedného až po 10 znakov. Po úspešnom skontrolovaní sa získajú detaily z databázy. Získané informácie sú naformátované do textového reťazca v tvare názov, lokalita, popis a názov súboru uloženej trasy. V prípade úspešného získania dát bude návratový reťazec začínať prefixom „OK:“ a v prípade zlyhania „ERROR:“. Samotný obsah GPS trasy je prenášaný samostatne.

### TrackData.ashx

Tento handler zaisťuje získanie a prenos obsahu GPS trasy do klientskej aplikácie. Vstupným parametrom je identifikátor GPS trasy. Identifikátor je kontrolovaný rovnakým regulárnym výrazom ako handler TrackData.ashx.

Obsah GPS trasy je prenášaný binárne. HTML hlavička má zmenený typ na application/octet-stream. Následne sú posielané dáta obsahu súboru. Obsah súboru získava uložená procedúra TrackData z databázy.

Ak je zadaný identifikátor, ktorý neprejde validáciou, handler vráti chybovú správu „ERROR“

### TrackNewCategory.ashx

Handler TrackNewCategory.ashx vytvára novú kategóriu pre GPS trasy. Má dva vstupné parametre. Prvým parametrom je názov kategórie. Správnosť názvu kategórie je kontrolovaná regulárnym výrazom `[a-zA-Z0-9\ - ]{1,50}`\$. Názov môže obsahovať iba znaky anglickej abecedy s pomlčkou a medzerou. Dĺžka názvu je minimálne jeden znak a maximálne 50. Popis kategórie je voliteľný a nemusí byť zadaný. Správnosť formátu popisu sa skontroluje pomocou regulárneho výrazu `^[a-zA-Z0-9\ - ]{0,100}`\$. O pridanie záznamu do databázy sa postará uložená procedúra TrackCategoryAdd. Úspešné vytvorenie databázy je v klientskej aplikácii oznámené návratovou správou „OK“, chybný stav správou „ERROR“.

### TrackAdd.ashx

Na pridanie novej GPS trasy slúži handler TrackAdd.ashx. Pre úspešné pridanie trasy je potrebné zadať názov trasy, jej lokalitu (približné miesto), binárny obsah zaznamenej trasy a identifikátor kategórie, do ktorej má byť zaradená. Voliteľne môže byť pridaný popis trasy. Na popis sú kladené rovnaké požiadavky ako na popis kategórie GPS trás. Všetky vstupné parametre okrem dát GPS trasy sú kontrolované regulárnymi výrazmi. Trasa sa pridá do databázy iba vtedy, keď je so vstupnými parametrami zaslaný aj binárny obsah GPS trasy. Úspešné vloženie záznamu trasy bude potvrdené návratovou správou „OK“. Pri nesprávnom počte alebo údajoch v nesprávnom formáte bude aplikácii zaslaná správa „ERROR“.

### 5.2.4 Fotografie

Sprostredkovaním dát medzi serverovou časťou a klientskou aplikáciou je fotografia a albumy podobné GPS trasám. Každá fotografia musí byť zaradená v jednom z albumov (kategórií). Pre spracovanie Fotografií sú na serveri vytvorené nasledujúce ASP.NET HTTP Handler-y:

#### PhotoAlbum.ashx

Pomocou PhotoAlbum.ashx sa z databázy získa abecedne usporiadaný zoznam fotoalbumov. Každý album má pridelený svoj identifikátor id, názov a voliteľne popis. Informácie o jednotlivých fotoalbumoch sú naformátované do textového reťazca id;názov;popis. Reťazce tvoriace albumy sa oddeľujú ukončením riadka \n.

Na začiatku návratovej správy je pridaný znak úspešného získania dát „OK:“ alebo chybový stav „ERROR“.

---

```
OK:1;Fotoalbum 1;popis fotoalbumu 1
2;Fotoalbum 2;popis fotoalbumu 2
```

---

Výpis 35: Zoznam fotoalbumov

#### PhotoList.ashx

PhotoList.ashx zaist'uje výpis zvoleného fotoalbumu. Zvolený fotoalbum sa rozlišuje pomocou identifikátora idCategory. Tento identifikátor môže obsahovať iba číslice 0 až 9. Dĺžka identifikátora môže byť 1 - 10 znakov. Správny formát je kontrolovaný pomocou regulárneho výrazu.

Po úspešnom skontrolovaní sa zavolá uložená procedúra PhotoList, ktoré zaistí výber fotografií pre zvolený fotoalbum. Fotografie sú triedené podľa id vzostupne. Najnovšia fotografia sa zobrazí ako prvá.

Na začiatku zoznamu fotografií je pridaný znak úspešného získania dát „OK:“ alebo chybový stav „ERROR“.

Fotografie sú získané samostatne pomocou handlera PhotoData.ashx so vstupným identifikátorom fotografie. „Obsah“ fotografie je prenášaný v binárnej forme.

### PhotoAdd.ashx

Pomocou handlera PhotoAdd.ashx sa ku fotografii z databázy pridávajú nasledujúce informácie: názov fotografie, popis, autor a identifikátor albumu, do ktorého bude fotografia zaradená. Tieto užívateľom zadávané informácie sú kontrolované na správny formát regulárnymi výrazmi.

Po úspešnej kontrole vstupných dát, sa zavolá uložená procedúra PhotoAdd. Obsah súboru je ukladaný do databázy v binárnej forme. Úspešné uloženie sa je signalizované klientskej aplikácii návratovou správou „OK“. Neúspech v prípade nesprávnych vstupných dát alebo chybe v komunikácii so SQL serverom signalizuje návratová správa „ERROR“.

### PhotoNewAlbum.ashx

Vytváranie nových fotoalbumov je rovnaké ako vytváranie nových kategórií GPS trás v handleri TrackNewCategory.ashx. Rozdiel je iba v zavolanej uloženej procedúre. V tomto prípade je použitá procedúra PhotoCategoryAdd.

## 5.3 Administrátorské rozhranie

Administrátorské rozhranie na serverovej časti slúži na správu dát. Je tu možné vytvárať, upravovať a mazat' obsah portálu. Do tejto časti majú prístup len užívatelia, ktorí sú priradení do skupiny Administrátor. Administrátorské rozhranie portálu je vytvorené pomocou ASPX stránok a štandardných ovládacích prvkov. Administrátorské rozhranie na serverovej časti slúži na správu dát. Je tu možné vytvárať, upravovať a mazat' obsah portálu. Do tejto časti majú prístup len užívatelia, ktorí sú priradení do skupiny Administrátor. Administrátorské rozhranie portálu je vytvorené pomocou ASPX stránok a štandardných ovládacích prvkov.

Do administratívneho rozhrania sa dostaneme kliknutím na odkaz „Admin“ v spodnej časti stránky alebo zadaním url adresy [http://adresa\\_servra/Admin/](http://adresa_servra/Admin/). Prístup do tejto časti portálu majú iba oprávnení používateľa zaradení do skupiny Administrátor.

V administráčnom rozhraní je možné upravovať obsah pre jednotlivé časti portálu.

### POI

Pomocou tejto stránky je možné vytvárať, upravovať alebo mazat' kategórie POI bodov. Zoznam kategórií je tvorený pomocou GridView objektu. Vo výslednej tabuľke sa zobrazí ikona, názov a popis kategórie. Štvrtý stĺpec sa zobrazuje iba pri úprave, mazaní kategórie. V tomto stĺpci sa nachádza tlačidlo pre zobrazenie detailov kategórie pomocou DetailView objektu, v ktorom je možné kategóriu upraviť, alebo tlačidlo pre zmazanie.

Nové kategórie sa pridávajú pomocou samotného formulára. Textové polia formulára sú pred odoslaním skontrolované nielen na strane klienta, ale aj na strane servera pomocou regulárnych výrazov. O chybách je používateľ informovaný vedľa textového poľa, v ktorom sa nachádza neplatný formát zadaných dát a v celkovom sumári nad formulárom.

Stránka POI slúži aj na správu jednotlivých POI bodov. Môžeme ich tu pridávať, schvaľovať, upravovať a mazat'. V zozname POI bodov sa nachádzajú iba POI body



zvolenej kategórie. Kategóriu môžeme zmeniť na tabuľku bodov. Pri zobrazení POI bodov, ktoré ešte neboli schválené sa nerozlišuje kategória. Kategória bodu je uvedená v zozname.

Pri pridávaní POI bodu sa na pozadí názvu a zemepisnej dĺžky a šírky zaktualizuje náhľad na mapu s vyznačeným bodom. Všetky textové polia sú kontrolované na strane klienta a servera.

Úprava a mazanie POI bodu je zhodné s úpravou a mazaním kategórie bodov.

## **Ikony**

V tejto časti administrátorského rozhranie môžeme pridávať a mazať nepotrebné ikony. Ikony sa pridávajú pomocou jednoduchého formulára. V tomto formulári sa nachádza iba tlačidlo s textovým poľom pre výber obrázka a tlačidlo Pridať. Pridávať sa môžu iba obrázky v JPEG formáte. Odporúčaný rozmer je 22 pixelov pre každý rozmer.

Pod formulárom na pridanie ikony je zoznam už existujúcich ikon. Na pravej strane riadka sa nachádza tlačidlo pre jej zmazanie. Po kliknutí na toto tlačidlo sa súbor fyzicky zmaže z disku.

## **GPS trasy**

V tejto časti administratívneho rozhrania portálu môžeme spravovať kategórie GPS trás a jednotlivé GPS trasy.

Kategórie môžeme vytvárať, mazať alebo upravovať ich názov a popis. Pri vytváraní musíme zadať len názov kategórie, jej popis je voliteľný. Po zadaní názvu a popisu sa zavolá uložená procedúra TrackAdd, ktorá vytvorí záznam do databázovej tabuľky *trackCategory*. Pri mazaní kategórie sa zmažú aj jednotlivé GPS trasy, priradené mazanej kategórii. Po kliknutí na tlačidlo "zmazať vybranú kategóriu" sa zavolá uložená procedúra *trackCategoryDelete*, ktorá odstráni GPS trasy zaradené v kategórii a potom samotnú trasu. Pri upravovaní informácií o kategórii a trase môžeme meniť len informácie o kategórii a trase. Nemôžeme zmeniť obsah trasy.

## **Fotografie**

Správa fotografií je dost' podobná správe GPS tras. Tak isto tu môžeme len pridávať, upravovať alebo rušiť fotoalbumy a pridávať, mazať, upravovať fotografie. Pri mazaní fotoalbumu sa zmažú aj v ňom obsiahnuté fotografie. Pri úprave fotografií nie je možné vymeniť fotografiu za inú, len zmeniť je popis.

## 6 Zaver

Technológia Adobe AIR umožňuje vytvárať aplikácie, ktoré nebudú závislé na operačnom systéme. Aplikácie môžu byť spúšťané na operačných systémoch Windows, Linux a Mac OS X. Aplikácia si môže v týchto operačných systémoch zachovať vzhľad natívnej aplikácie, mať jednotný vzhľad vytvorený technológiou Adobe Flex alebo mať úplne vlastný vzhľad. Okno aplikácie nemusí mať zaužívaný obdĺžnikový tvar.

Aplikácie vytvorené pomocou tejto technológie majú prístup k lokálnemu systéme a interakcii s operačným systémom. Môžu pristupovať k lokálnemu súborovému systému, komunikovať po sieti, využívať lokálne šifrované úložisko, lokálnu databázu, pracovať so zvukom, zobrazovať PDF a SWF obsah. Aplikácie môžu byť spustené vo vlastnom okne alebo v internetovom prehliadači. Ak je aplikácia spustená vo vlastnom okne, je na jej zobrazenie použité zobrazovacie jadro WebKit. Pri tomto type zobrazenia aplikácie odpadá nutnosť optimalizácie vzhľadu a funkčnosti aplikácie pre každý internetový prehliadač. Aplikácia spustená v internetovom prehliadači je obmedzená oproti aplikácii v samostatnom okne o spustenie na pozadí (aplikácia sa ukončí zatvorením internetového prehliadača), vzhľad a implementácia niektorých prvkov užívateľského rozhrania je ovplyvnená internetovým prehliadačom, aplikácia má obmedzené úložisko dát.

Medzi výhody tejto technológie zaradíme nezávislosť na operačnom systéme, odstránenie izolovanosti RIA aplikácií, možnosť prispôsobenia si vzhľadu aplikácie podľa vlastnej potreby. Táto technológia je vhodná na vytvorenie desktopových aplikácií, pri ktorých sa vyžaduje jednotný vzhľad so serverovou časťou. Týmto spôsobom je možné drobnými úpravami vytvoriť úplne identickú aplikáciu. Desktopové aplikácie založené na technológii Adobe AIR môžu začať vyvíjať ľudia so znalosťami technológií určených pre internetové stránky HTML, CSS, JavaScript (Ajax), Flash.

Medzi nevýhody by som zaradil nevýhody interpretovaného jazyka JavaScript. Aplikáciu vytvorenú technológiami HTML, CSS, JavaScript je možné ľahko upraviť a zmeniť jej funkčnosť. Nainštalovaná aplikácia je tvorená textovými súbormi.

Ukážková aplikácia spĺňa podmienky stanovené pred začiatkom písania tejto práce. Ide o aplikáciu, ktorá využíva pre dáta komunikáciu server - klient. Tento typ komunikácie zaručuje dostupnosť vždy aktuálnych dát v aplikácii. Na portáli môžeme zhromažďovať informácie o bodoch záujmu, GPS trasách a fotografiách, ktoré sú prehľadne usporiadané do kategórií. Body záujmu je možné vyexportovať v GPX formáte po jednotlivých kategóriách. Tento formát je veľmi dobre podporovaný aj v starších aplikáciách určených na spracovanie geografických informácií.

## 7 Literatúra

- [1] *Developing Adobe AIR 1.5 Applications with HTML and Ajax* [online]. Developing Adobe AIR 1.5 Applications with HTML and Ajax [cit. 2010-05-04]. Dostupné z WWW: <[http://help.adobe.com/en\\_US/AIR/1.5/devappshtml/index.html](http://help.adobe.com/en_US/AIR/1.5/devappshtml/index.html)>.
- [2] *Adobe AIR Language Reference for HTML Developers* [online]. 2009 [cit. 2009-12-04]. Dostupné z WWW: <[http://help.adobe.com/en\\_US/AIR/1.5/jslr/](http://help.adobe.com/en_US/AIR/1.5/jslr/)>.
- [3] CHAMBERS, Mike; DURA, Daniel; HOYT, Kevin. *Adobe Integrated Runtime (AIR) for JavaScript Developers : Pocket Guide*. Sebastopol : O'Reilly Media, 2007. 176 s. ISBN 978-0-596-51519-5.
- [4] CHAMBERS, Mike; DURA, Daniel; Georgita, Dragos; HOYT, Kevin. *Adobe AIR for JavaScript Developers : Pocket Guide*. Sebastopol : O'Reilly Media, 2008. 204 s. ISBN 978-0-596-51837-0.
- [5] *Browser vs. desktop* [online]. 2009 [cit. 2009-04-24]. Dostupné z WWW: <<http://www.adobe.com/products/air/comparison/>>.
- [6] *OpenSource JavaScript implementation of the Secure Hash Algorithms, SHA-256-384-512* [online]. 2004 [cit. 2010-01-20]. Dostupné z WWW: <<http://anmar.eu.org/projects/jssha2/>>.
- [7] *Adobe Labs : Spry framework for Ajax* [online]. 2010 [cit. 2010-04-04]. Dostupné z WWW: <<http://labs.adobe.com/technologies/spry/>>.
- [8] *Google Code : Google Maps API* [online]. 2010 [cit. 2010-01-10]. Dostupné z WWW: <<http://code.google.com/intl/sk-SK/apis/maps/>>.
- [9] *Open Geospatial Consortium : KML* [online]. 2010 [cit. 2010-05-05]. Dostupné z WWW: <<http://www.opengeospatial.org/standards/kml/>>.

## **A DVD-ROM**

Obsah DVD-ROM:

- Zdrojové kódy aplikácie
- Sputiteľná aplikácia
- Zdrojové kódy portálu
- Text diplomovej práce