

# RSS offline čtečka pro mobilní platformu

Jiří Mačát

---

Bakalářská práce  
2014



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2013/2014

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří MAČÁT**  
Osobní číslo: **A10173**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Informační a řídicí technologie**  
Forma studia: **kombinovaná**

Téma práce: **RSS offline čtečka pro mobilní platformu**

Zásady pro vypracování:

1. Vytvořte aplikaci pro mobilní zařízení s OS Android, resp. iOS, která bude číst RSS kanály.
2. Aplikace bude podporovat synchronizaci dat z článku v off-line režimu.
3. Implementujte možnost stahování na sekundárním vláknu na pozadí.
4. Součástí aplikace bude správa kanálu s okamžitým ukládáním bez zásahu uživatele.
5. Vytvořenou aplikaci otestujte a vytvořte uživatelskou dokumentaci.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. KOSEK, Jiří. XML pro každého: Podrobný průvodce. Praha: Grada Publishing, 2000. ISBN 80-7169-860-1. Dostupné také z: <http://www.kosek.cz/xml/xmlprokazdeho.pdf>
2. SCHILDT, Herbert. JAVA7: Výukový kurz. Brno: COMPUTER PRESS, 2012. ISBN 978-80-251-3748-2
3. GRANT, Allen. Android 4: Průvodce programováním mobilních aplikací. Brno: COMPUTER PRESS, 2013. ISBN 978-80-251-3782-6
4. HOLZNER, Steven. RSS: Automatické doručování obsahu vašich www stránek. Brno: COMPUTER PRESS, 2007. ISBN 80-251-1479-1
5. MEIER, Reto. Professional Android 4 Application Development. New York City: John Wiley & Sons, Inc. ISBN 978-1-118-10227-5
6. Android Developers[online]. Dostupné z: <http://developer.android.com/>

Vedoucí bakalářské práce:

**Ing. Dalibor Slovák**

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

**28. února 2014**

Termín odevzdání bakalářské práce:

**13. června 2014**

Ve Zlíně dne 28. února 2014

prof. Ing. Vladimír Vašek, CSc.  
*děkan*



prof. Ing. Vladimír Vašek, CSc.  
*ředitel ústavu*

### **Prohlašuji, že**

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

### **Prohlašuji,**

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

Ve Zlíně

.....  
podpis diplomanta

## **ABSTRAKT**

Cílem této bakalářské práce je přiblížit čtenáři způsob vývoje aplikací pro operační systém Android a vytvořit ukázkovou aplikaci, která bude stahovat kanály RSS a zpřístupňovat jejich obsah k off-line užití. Práce je členěna do několika částí. První kapitola popisuje jazyky, které byly užity k programování ukázkové aplikace. Druhá kapitola se zaměřuje na popis operačního systému Android, popisuje jednotlivé verze tohoto systému a jeho architekturu. Poslední kapitola teoretické části pak popisuje vývojové prostředí a základní komponenty Android aplikací. V praktické části je popsán vzhled aplikace a funkce jednotlivých aktivit.

Klíčová slova: Off-line RSS, operační systém Android, Android aplikace

## **ABSTRACT**

The purpose of this Bachelor's thesis is to define and interpret the development of applications for the Android operating system, and to create a prototype application. This prototype would download existing RSS channels and make their content available for off-line viewing. In the first chapter, the programming languages used for developing the prototype application are presented. The second chapter provides an overview of the Android operating system, its versions, and its architecture. The third chapter describes basic components and development environments of Android applications. The last part of this thesis explains and outlines the uses and functions of the prototype application.

Keywords: Off-line RSS, Android operating system, Android application

Děkuji vedoucímu mé bakalářské práce, Ing. Daliboru Slovákovi, za umožnění zpracování tohoto tématu, trpělivost a cenné rady, zejména při dokončování této práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 TECHNOLOGIE</b> .....	<b>11</b>
1.1    JAZYK JAVA.....	11
1.2    JAZYK XML.....	11
1.3    RSS.....	12
1.3.1    Využití.....	13
1.3.2    Elementy kanálu RSS 2.0.....	13
<b>2 PLATFORMA GOOGLE ANDROID</b> .....	<b>14</b>
2.1    HISTORIE VERZÍ.....	14
2.1.1    Android 1.0 – (API 1) .....	14
2.1.2    Android 1.1 – (API 2) .....	15
2.1.3    Android 1.5 „Cupcake“ (API – 3).....	15
2.1.4    Android 1.6 „Donut“ (API – 4).....	15
2.1.5    Android 2.0/2.1 „Eclair“ (API – 5 – 7) .....	16
2.1.6    Android 2.2 „Froyo“ (API – 8) .....	16
2.1.7    Android 2.3 „Gingerbread“ (API – 9 – 10).....	16
2.1.8    Android 3.0/3.1/3.2 „Honeycomb“ (API – 11 – 13).....	17
2.1.9    Android 4.0 „Ice Cream Sandwich“ (API – 14 – 15).....	17
2.1.10    Android 4.1/4.2/4.3 „Jelly Bean“ (API – 16 – 18).....	18
2.1.11    Android 4.4 „KitKat“ (API – 19).....	18
2.1.12    Podíl jednotlivých verzí na trhu .....	19
2.2    ARCHITEKTURA.....	19
2.2.1    Dalvik Virtual Machine.....	20
2.2.2    Android Framework .....	20
<b>3 PROGRAMOVÁNÍ – TVORBA APLIKACÍ</b> .....	<b>21</b>
3.1    VÝVOJOVÉ PROSTŘEDÍ .....	21
3.1.1    Emulátor .....	21
3.1.2    AVD Manager .....	22
3.2    TVORBA APLIKACÍ.....	23
3.2.1    Aktivita.....	23
3.2.2    Context .....	25
3.2.3    Intent a intent filter.....	25
3.2.4    Broadcast Receiver.....	26
3.2.5    Služba.....	26
3.2.6    AsyncTask.....	28
3.2.7    Ukládání dat .....	28
3.2.8    Suroviny .....	30
3.2.9    Uživatelské rozhraní.....	31
<b>II PRAKTICKÁ ČÁST</b> .....	<b>34</b>
<b>4 POPIS APLIKACE</b> .....	<b>35</b>

4.1	PRVNÍ SPUŠTĚNÍ .....	36
4.2	PŘIDÁNÍ ZDROJŮ .....	36
4.3	AKTUALIZACE ČLÁNKŮ .....	37
4.4	SEZNAM ČLÁNKŮ – HLAVNÍ AKTIVITA APLIKACE .....	39
4.5	MANIPULACE SE ZDROJI .....	41
4.6	DETAIL POLOŽKY .....	42
4.7	NASTAVENÍ .....	43
	<b>ZÁVĚR .....</b>	<b>45</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>47</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>49</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>51</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>52</b>

## ÚVOD

Tématem bakalářské práce je vývoj aplikace pro mobilní platformu. Zatímco v nedávné době byly mobilní telefony využívány pouze k telefonování a posílání textových zpráv a prohlížení internetových stránek bylo možné pouze prostřednictvím stolních počítačů nebo notebooků, stávají se dnes mobilní zařízení jedním z nejvyužívanějších prostředků pro získávání informací. K tomu, aby mohla mobilní zařízení přebírat funkce, které dříve byly dominantou pouze počítačů, musí být vytvářeny čím dál více sofistikovanější prostředky jak tohoto cíle dosáhnout. Jednou z těchto možností bylo navrzení plnohodnotného operačního systému, který by umožňoval nejen přístup k internetu, ale i možnost vývoje aplikací, které by tuto činnost usnadňovaly. Mezi nejvyužívanější operační systémy pro mobilní zařízení dnes patří především operační systém Android, dále iOS, který využívají výhradně výrobky společnosti Apple a Windows Phone, který je nástupcem platformy Windows Mobile.

Zadání této práce umožňovalo výběr vývoje aplikace pro mobilní platformu Android, respektive iOS. Zvolil jsem si operační systém Android. Vývoj aplikace pro tento operační systém jsem si vybral především z důvodu mého zájmu o pochopení způsobu vytváření aplikací pro tuto platformu, jejíž vývoj sleduji již od jejího počátku. Nemalou mírou se na mé volbě podílel i fakt, že operační systém Android je dnes nejvyužívanější a nejdynamičtěji se rozvíjející, otevřená mobilní platforma, kterou je navíc možné použít na jakémkoliv zařízení.

Hlavním cílem práce je vytvořit aplikaci, která bude umět číst informace z kanálů RSS a zpřístupňovat jejich obsah k off-line užití. Dílčím cílem práce je pak implementace stahování těchto dat na pozadí aplikace, automatická správa těchto kanálů a vytvoření uživatelské dokumentace.

Teoretická část práce bude vycházet především z poznatků získaných z oficiální dokumentace operačního systému Android a knihy Android 4, která byla přeložena do češtiny a jejíž autorem je Grant Allen, specialista na návrh nadnárodních systémů společnosti Google. Oficiální dokumentace je dostupná na stránkách společnosti Google, které jsou určeny vývojářům aplikací a obsahují nejaktuálnější informace o této mobilní platformě.

Praktická část pak bude představovat uživatelskou dokumentaci k vytvořenému programu.

Pro správné pochopení některých částí této práce je od čtenáře vyžadována alespoň základní orientace v oblasti vývoje aplikací v objektově orientovaném jazyku.

## **I. TEORETICKÁ ČÁST**

# 1 TECHNOLOGIE

## 1.1 Jazyk JAVA

Aplikace pro Android je možné psát přímo v nativním kódu jazyka C nebo C++ [1], avšak většina aplikací je psána v jazyku Java. Aby bylo možné zahájit vývoj v tomto jazyku, předpokládá se znalost tohoto jazyku alespoň v těchto oblastech:

- *„Základy jazyka (řízení toku programu atd.)*
- *Koncepce tříd a objektů*
- *Koncepce metod a datových členů*
- *Koncepce veřejných, privátních a chráněných entit*
- *Koncepce výjimek*
- *Koncepce vláken a řízení souběžnosti*
- *Koncepce kolekcí*
- *Koncepce obecných entit*
- *Vstupně-výstupní operace se soubory*
- *Koncepce reflexe*
- *Koncepce rozhraní“ [2, str. 34]*

Jazyk Java je objektově orientovaný jazyk, který v roce 1991 začali vyvíjet programátoři společnosti Sun Microsystems jako jazyk „Oak“ a v roce 1995 byl přejmenován a představen jako jazyk „Java“. Vychází z jazyku C, od kterého přebral jeho syntaxi a jazyku C++, od kterého převzal svůj objektový model. Hlavní výhodou jazyku Java je jeho přenositelnost, neboť kompilátor jazyku Java neprodukuje přímo spustitelný kód, ale takzvaný bajtový kód (Bytecode). Bytecode je sada instrukcí, která je zpracovávána virtuálním strojem Javy (Java Virtual Machine). [3] V systému Android pak tuto činnost přebírá Dalvik VM – vlastní virtuální stroj společnosti Google. [4]

## 1.2 Jazyk XML

Extensible Markup Language (XML) je otevřený značkovací jazyk vyvinutý konsorciem W3C, které se stará i o další technologie spojené s prezentací na Internetu. Byl zkonstruován pro potřeby označení významu jednotlivých částí dokumentu, který má být určen ke strojovému zpracování. Využívá 32 bitové znakové sady ISO 10646, která je schopná pojmout

všechny znaky dnes používaných jazyků (čeština, arabština, ruština a další). Dokument tak může obsahovat i texty několika jazyků najednou. Aby však nedocházelo ke zbytečnému plýtvání místem, může být dokument XML v libovolném kódování, které však musí být každému dokumentu přesně určeno. Jazyk XML umožňuje definování vlastní sady značek, které chceme v dokumentu využívat. Program, který pak dokument čte (parser) může pomocí této sady značek přesně určit význam jednotlivé položky. Definování vlastní sady značek však může způsobit problémy při zpracovávání dokumentu, který není určen pouze k našemu užití. Existují proto různá sdružení, která vydávají schémata, jež by se měly v dané oblasti používat – například XML.ORG provozovaný sdružením OASIS.

Každý dokument XML se skládá z elementů, které jsou do sebe vnořovány. Elementy se v textu vyznačují pomocí tzv. tagů. Většinou elementů pak náleží dva tagy. Počáteční <tag> a koncový </tag>. Některé elementy však nemusí mít žádný obsah - mohou mít například formátovací charakter. Tyto elementy pak používají pouze počáteční tag, na jehož konci je symbol „/“. Příkladem může být tag <br/>. [5]

*“Každý XML dokument musí obsahovat pro všechny počáteční tagy odpovídající ukončovací tag, nebo musí být počáteční tag zapsán jako element s prázdným obsahem.” [5, str. 24]*

Při programování aplikace bude využit jazyk XML ke dvěma účelům. Jednak při definování vzhledu jednotlivých aktivit aplikace a potom při samotném „parsování“ jednotlivých položek RSS kanálů.

### 1.3 RSS

Really Simple Syndication (RSS) je formát pro syndikaci informací na www stránkách. Je založen na jazyku XML a každý RSS dokument musí odpovídat specifikaci XML 1.0. První verzi tohoto formátu navrhla společnost Netscape pro své stránky my.netscape.com v roce 1999. Jednalo se o formát RSS 0.90. Následovaly formáty RSS 0.91, RSS 1.0, RSS 0.92, RSS 0.93, a nakonec RSS 2.0. Aktuální formát RSS 2.0.1 spravuje Berkman Center for Internet & Society na Právnické fakultě Harvardovy univerzity. [6]



Obr. 1 Ikona RSS

### 1.3.1 Využití

Technologie RSS je dnes využíváno především u informačních webů, které pomocí této technologie oznamují nové články. Uplatnění je však velmi různorodé. Inzertní web může například upozorňovat na nově vložené inzeráty, je možné sledovat různá diskuzní fóra a existují například i různé internetové stránky, které stahují RSS kanály jiných webů a návštěvníkovi pak nabízí tyto informace zařazené do různých kategorií. [7]

### 1.3.2 Elementy kanálu RSS 2.0

Obsah dokumentu musí být uzavřen v elementu <rss>, který obsahuje atribut version s odpovídající hodnotou verze formátu RSS. V něm je pak uzavřen element <channel>, ve kterém už jsou jednotlivé položky <item>.

RSS standard ve verzi 2.0 vyžaduje tyto elementy jako povinné:

- title – titulek článku
- link – odkaz na původní článek
- description – popis odkazovaného článku

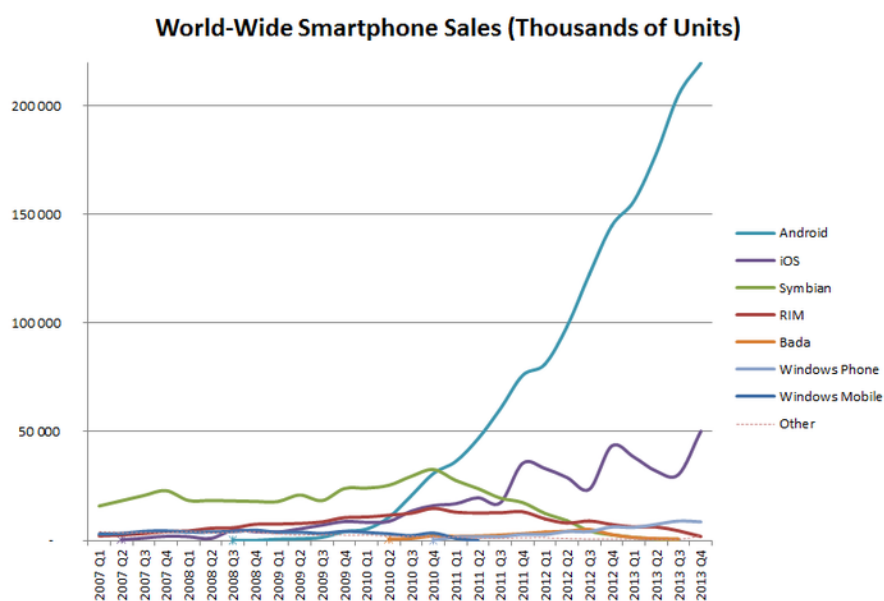
Dále tento standard nabízí i několik volitelných elementů:

- language – jazyk ve kterém je článek napsán
- copyright – copyright
- managingEditor – emailová adresa osoby zodpovědné za obsah článku
- webMaster – emailová adresa osoby zodpovědné za technické řešení kanálu
- pubDate – datum a čas publikování článku
- lastBuildDate – časový údaj o poslední změně obsahu
- category – specifikuje kategorii do které kanál nebo jeho položka spadá
- generator – řetězec indikující program, který kanál generoval
- docs – URL adresa odkazující na specifikaci použité verze RSS
- image – obrázek typu gif, jpeg nebo png, který je zobrazen v kanálu
- a další [6]

Kompletní seznam je k dispozici na adrese <http://cyber.law.harvard.edu/rss/rss.html>

## 2 PLATFORMA GOOGLE ANDROID

Android je operační systém určený především pro mobilní zařízení, který využívá stovky milionů zařízení ve více než 190 zemích po celém světě. Je nejrozšířenější mobilní platformou a každý den aktuálně přibde více než jeden milion nových uživatelů tohoto systému. [1] Je vyvíjen společností Google. Oficiální verze obsahují základní systémové aplikace pro obsluhu jednotlivých zařízení a základní aplikace společnosti Google určené pro propojení s jejich službami. [8] Uživatelské aplikace pak lze instalovat například pomocí služby Google Play, která aktuálně nabízí více než 1,5 miliardy aplikací. [1]



Obr. 2 Zastoupení mobilních operačních systémů na trhu s mobilními telefony

### 2.1 Historie verzí

Před samotným vývojem konkrétní aplikace je nutné rozhodnout se, na které verze systému Android bude aplikace vytvářena. K tomu, abychom mohli toto rozhodnutí učinit, je nutné mít alespoň základní přehled o historii systému Android, respektive o možnostech jednotlivých verzí, jejich API a jejich podílu na trhu.

#### 2.1.1 Android 1.0 – (API 1)

První verze operačního systému Android – Android 1.0 – byla uvedena na trh společně s telefonem T-Mobile G1 22. října 2008 a byla postavena na linuxovém jádru 2.6.25.

I když zdaleka neobsahovala funkce, na které je dnešní uživatel systému Android zvyklý, stavěla na prvcích, které můžeme najít i v aktuálních verzích tohoto systému:

- Stahovací oznamovací lišta
- Widgety na domácí (hlavní) ploše
- Hluboká integrace se službami Gmail
- Android Market (současný Google Play) [9]

### 2.1.2 Android 1.1 – (API 2)

První povýšení systému – únor 2009. Obsahovala opravy chyb a přidala možnost „over the air“ (OTA) aktualizací. [9]

### 2.1.3 Android 1.5 „Cupcake“ (API – 3)

Květen 2009. Povýšení linuxového jádra na verzi 2.6.27. Přinášela zlepšení výkonu a přidala několik nových funkcí:

- Softwarová klávesnice
- Rozšíření funkčnosti widgetů
- Rozšíření možností schránky (kopírovat/vložit)
- Nahrávání a přehrávání videa z kamery telefonu
- Nahrávání videa a fotografií na Youtube/Picasa přímo z telefonu

Kompletní seznam je k dispozici na adrese <http://developer.android.com/about/versions/android-1.5-highlights.html>

### 2.1.4 Android 1.6 „Donut“ (API – 4)

Září 2009. Povýšení linuxového jádra na verzi 2.6.29. Změny:

- Quick Search Box (umožňuje vyhledávání z více zdrojů v telefonu nebo na internetu)
- Update UI a rozšíření funkcionality kamery, fotoaparátu a galerie
- Podpora technologie VPN, 802.1x a CDMA
- Podpora gest
- Text-to-speech engine (syntéza řeči)
- V nastavení přidána volba pro sledování využití baterie

Kompletní seznam je k dispozici na adrese <http://developer.android.com/about/versions/android-1.6-highlights.html>

### 2.1.5 Android 2.0/2.1 „Eclair“ (API – 5 – 7)

Říjen 2009. Změny:

- Nový seznam kontaktů
- Podpora Microsoft Exchange
- Vylepšení emailového klienta
- Vyhledávání v SMS, MMS a možnost nastavení uchovávání maximálního počtu zpráv
- Vylepšení pro aplikaci fotoaparát
- Nové uživatelské prostředí internetového prohlížeče a podpora HTML 5
- Optimalizace grafické architektury, která umožnila lepší hardwarovou akceleraci.

Kompletní seznam je k dispozici na adrese <http://developer.android.com/about/versions/android-2.0-highlights.html>

### 2.1.6 Android 2.2 „Froyo“ (API – 8)

Květen 2010. Povýšení linuxového jádra na verzi 2.6.32. Změny:

- Nová domácí plocha (Launcher)
- Wi-Fi Hotspot – umožňující sdílení internetového připojení až pro 8 zařízení
- Možnost instalace aplikací na paměťovou kartu
- Vylepšena softwarová klávesnice – podpora více jazyků
- Vylepšení výkonu – Dalvik JIT, internetový prohlížeč, správa RAM
- Podpora OpenGL ES2.0

Kompletní seznam je k dispozici na adrese <http://developer.android.com/about/versions/android-2.2-highlights.html>

### 2.1.7 Android 2.3 „Gingerbread“ (API – 9 – 10)

Prosinec 2010. Povýšení linuxového jádra na verzi 2.6.35. Změny:

- Softwarová klávesnice upravena pro lepší zadávání a editaci textu
- Vylepšené sledování využití baterie
- Podpora SIP při internetovém telefonování
- Podpora technologie NFC (Near Field Communication)
- Přidán správce pro stahování

- Vylepšená správa aplikací
- Rozšíření podpory nativního kódu

Kompletní seznam je k dispozici na adrese <http://developer.android.com/about/versions/android-2.3-highlights.html>

### **2.1.8 Android 3.0/3.1/3.2 „Honeycomb“ (API – 11 – 13)**

Únor 2011. Jednalo se o verzi určenou především pro internetové tablety a zařízení s větší obrazovkou. Přinesla zcela nové uživatelské rozhraní a další vylepšení:

- Aplikační panel akcí
- Vylepšená správa přizpůsobení domovské plochy
- Uživatelsky přívětivá správa spuštěných aplikací
- Aktualizace systémových aplikací (prohlížeč, fotoaparát, galerie, emailový klient, kontakty)
- Nové funkce pro vývojáře (fragments, akční panel, vylepšená správa upozornění, drag-and-drop, nástroje pro 2D a 3D grafiku, podpora architektury procesorů s více jádry, zpětná kompatibilita s existujícími aplikacemi)

Kompletní seznam je k dispozici na adrese <http://developer.android.com/about/versions/android-3.0-highlights.html>

### **2.1.9 Android 4.0 „Ice Cream Sandwich“ (API – 14 – 15)**

Říjen 2011. Přináší jednotné uživatelské rozhraní pro telefony, tablety a ostatní zařízení. Novinky:

- Možnost změny velikosti widgetů na domácí ploše
- Nový zámek obrazovky
- Možnost ukončení spuštěných úloh, upozornění a oken prohlížeče přejetím prstu po obrazovce
- Nástroj pro převod řeči na text
- Kontrola množství přenesených dat
- Panoramatické snímkování
- Android Beam – sdílení dat pomocí NFC technologie
- Odemknutí zařízení obličejem
- Wi-Fi P2P (peer-to-peer)

- Další nové funkce pro vývojáře

Kompletní seznam je k dispozici na adrese <http://developer.android.com/about/versions/android-4.0-highlights.html>

### **2.1.10 Android 4.1/4.2/4.3 „Jelly Bean“ (API – 16 – 18)**

Červenec 2012. Změny:

- Rychlejší, plynulejší, více responzivní (Project Butter)
- Rozšířené oznámení
- Možnost uložení fotografií ke kontaktům ve vysokém rozlišení
- „Chytrá“ aktualizace aplikací pomocí Google Play
- Google Now
- Více uživatelských účtů
- Vylepšení fotoaparátu o technologie PhotoSphere 360° a HDR
- Widgety na zámku obrazovky
- Podpora OpenGL ES 3.0

Kompletní seznam je k dispozici na <http://developer.android.com/about/versions/jelly-bean.html>

### **2.1.11 Android 4.4 „KitKat“ (API – 19)**

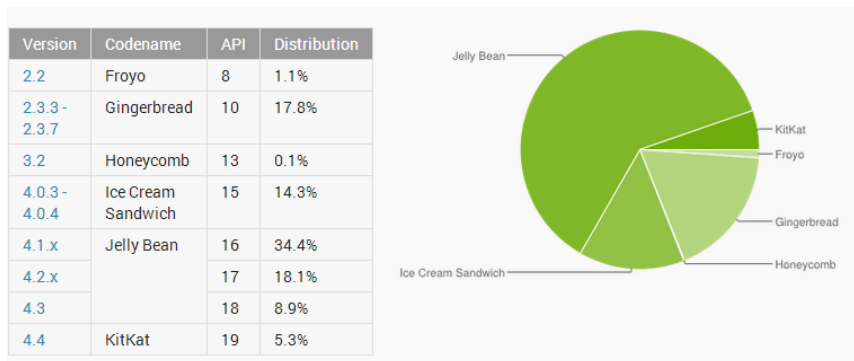
Září 2013. Aktuální verze systému Android:

- Zefektivnění využití paměti RAM
- Framework pro tisk dokumentů pomocí tiskáren s Wi-Fi nebo cloudové služby
- Framework pro jednoduchý přístup k datovým úložištím
- Optimalizace spotřeby energie hardwarových senzorů
- Krokoměř
- SMS provider
- Rozšířený Full Screen
- Vylepšený přístup k oznámením
- WebView založené na Chromium
- Možnost nahrávání obrazovky – umožňuje návrhářům vytváření video návodu

Kompletní seznam je k dispozici na <http://developer.android.com/about/versions/kitkat.html>

### 2.1.12 Podíl jednotlivých verzí na trhu

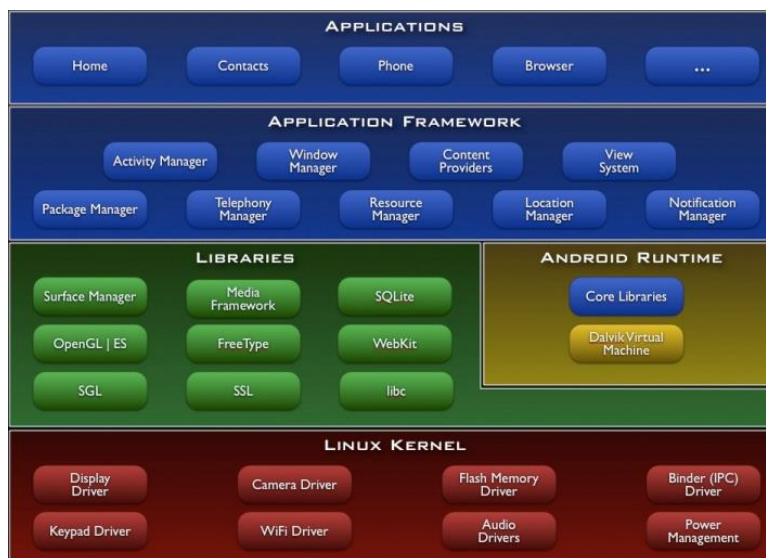
Podíl jednotlivých verzí systému Android na trhu ke dni 1.4.2014 ukazuje následující obrázek – Obr.3 [1]



Obr. 3 Podíl jednotlivých verzí systému Android na trhu ke dni 1.4.2014

## 2.2 Architektura

Jádro operačního systému Android tvoří linuxový Kernel. Linuxový Kernel spravuje ovladače zařízení, procesy, paměť zařízení, přidělování zdrojů a další systémové služby. I když je jádro systému založené na Linuxu, většina aplikací pro Android je programována v jazyce Java a běží na Dalvik Virtual Machine. Dalvik VM je součástí další vrstvy systému společně s Android knihovny a dalšími knihovny (Libc, SGLite, SSL, Webkit ..), které jsou napsány v jazycích C a C++. Následuje vrstva Android Framework (Android API), která zprostředkovává služby nejvyšší vrstvě – vrstvě aplikační. Viz obr.4. Architektura operačního systému Android. [4]



Obr. 4 Architektura operačního systému Android

### 2.2.1 Dalvik Virtual Machine

Jedním z klíčových prvků systému Android je Dalvik VM. Obsluhuje chod .dex souborů, které jsou vytvářeny pomocí Android SDK a jsou součástí .apk balíčku aplikace. Na rozdíl od Java VM byl Dalvik VM navržen přímo pro potřeby operačního systému Android. Dalvik VM využívá linuxový kernel ke zprostředkovávání nízko úrovněových funkcí jako je zabezpečení, správa vláken a procesů a správa paměti a vytváří tak abstraktní vrstvu, pomocí které lze přistupovat k hardwarovým a systémovým zdrojům bez nutnosti implementace obsluhy jednotlivých hardwarových konfigurací. [4]

### 2.2.2 Android Framework

Architektura systému Android umožňuje sdílet komponenty vlastní aplikace a využívat sdílené komponenty jiných aplikací dle definovaných bezpečnostních pravidel aplikace. Tato vlastnost poskytuje vývojářům veliký potenciál. Při vývoji aplikací je možné využít například následujících komponent: [4]

- Activity/Fragment Manager – řídí životní cyklus aktivity a fragmentů
- View System – využíván k tvorbě uživatelského prostředí aplikace
- Notification Manager – správa upozornění
- Content Providers – umožňuje aplikacím sdílet svá data
- Resource Manager – zpřístupňuje zdroje dat typu textové řetězce, grafické elementy, xml soubory ...
- Intents – poskytují mechanismus pro přenos dat mezi aplikacemi a jejich komponenty [4]

## 3 PROGRAMOVÁNÍ – TVORBA APLIKACÍ

### 3.1 Vývojové prostředí

Společnost Google na svých internetových stránkách určených pro vývoj aplikací pro operační systém Android aktuálně nabízí ke stažení a doporučuje sadu nástrojů - ADT Bundle. Jedná se o vývojové prostředí Eclipse, rozšířené o plugin ADT (Android Development Tools), který umožňuje vývoj, ladění a testování aplikací přímo z prostředí Eclipse. Tato sada také obsahuje Android SDK Tools, které jsou nutné například pro emulování Android zařízení a obsahuje také dokumentaci, image poslední verze systému Android určenou pro tento emulátor a další nástroje potřebné pro vývoj aplikací. Předchozí verze systému Android a volitelné nástroje pak lze stáhnout prostřednictvím Android SDK Manager, který je také součástí tohoto balíčku. Další hlavní úlohou Android SDK Manager je pak vytváření jednotlivých virtuálních zařízení, jejich správa a spouštění v emulátoru. [1]

Alternativní možností je využití jiných vývojových prostředí, jako například Android Studio, které před nedávnem společnost Google představila a na kterém pracuje, nebo některé z jiných vývojových prostředí (Rhodes, Flash, Flex a AIR, JRuby a Ruboto, Mono pro Android, App Inventor, Titanium Mobile a další). [2] První zmiňovaná možnost (Android Studio) bude pravděpodobně do budoucna vývojáři upřednostňována pro svou rozšířenou funkcionalitu určenou přímo pro vývoj Android aplikací. Umožňuje například jednoduchou lokalizaci aplikace, okamžitý náhled aplikace v samostatném okně při definování rozložení pomocí jazyka XML a to i s možností Preview All Screen Sizes, která zobrazí náhledy ve všech možných rozlišeních obrazovek. Android Studio je však aktuálně ve verzi „early access preview“ a některé funkce nejsou ještě dokončeny a mohou obsahovat chyby. [1] Proto jsem se pro samotný vývoj aplikace rozhodl použít oficiálně doporučené vývojové prostředí, tedy Eclipse s ADT pluginem.

#### 3.1.1 Emulátor

Emulátor je jednou z nejdůležitějších součástí sady Android SDK. Mohou se na něm testovat aplikace bez nutnosti instalace aplikace na fyzické zařízení a lze v něm emulovat v podstatě libovolné zařízení s libovolnou nabízenou konfigurací. Pomocí ADT pluginu jej lze integrovat do prostředí Eclipse a díky této vlastnosti lze aplikace na emulátoru spouštět a ladit přímo z tohoto prostředí.

Emulátor také umožňuje emulovat různé hardwarové konfigurace jako například:

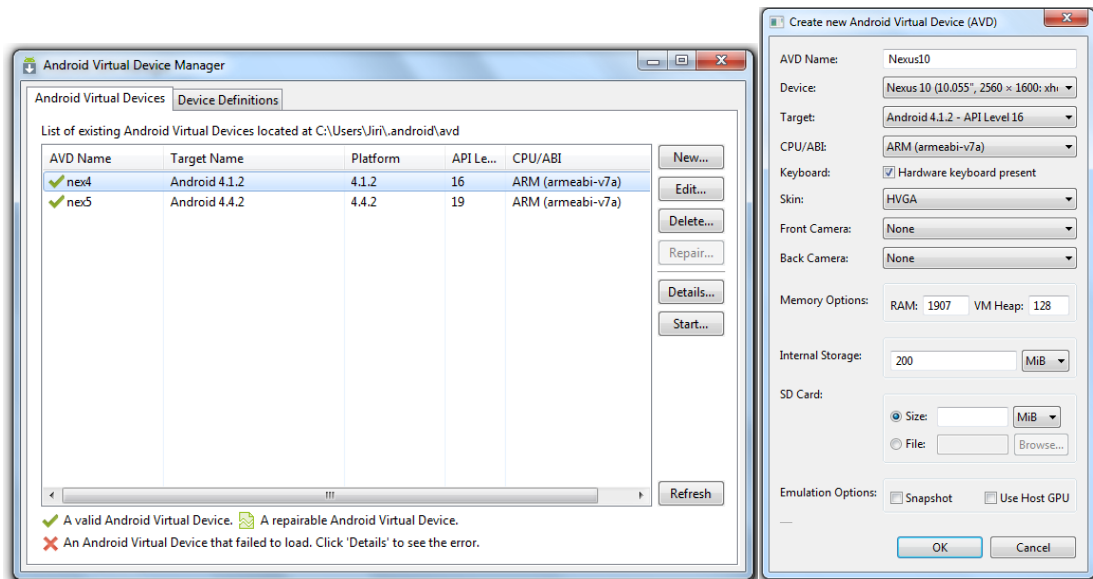
- „AMRv5 CPU s odpovídající MMU (Memory-Management Unit)
- 16 – bit LCD Display
- Jednu nebo více hardwarových klávesnic (QWERTY a hardwarová tlačítka)
- Zvukový chip se vstupem i výstupem
- Paměť typu Flash (jako diskový image)
- GSM modem umožňující simulaci SIM
- Fotoaparát/kameru (využívá webkameru počítače)
- Senzory“ [10]



Obr. 5 Android emulátor

### 3.1.2 AVD Manager

Pomocí AVD Manageru lze vytvářet a spravovat jednotlivá virtuální zařízení (AVD – Android Virtual Device). Každé AVD pak reprezentuje jedno zařízení, které má vlastní hardwarovou a softwarovou konfiguraci. Každé zařízení musí mít své jméno, Target API a emulovaný CPU. Dále je nutno nastavit velikost RAM a interní a virtuální paměti. Libovolně se pak může nastavovat vzhled, přítomnost přední a zadní kamery, hardwarové klávesnice a přítomnost (velikost) SD karty.



Obr. 6 Android Virtual Device Manager

## 3.2 Tvorba aplikací

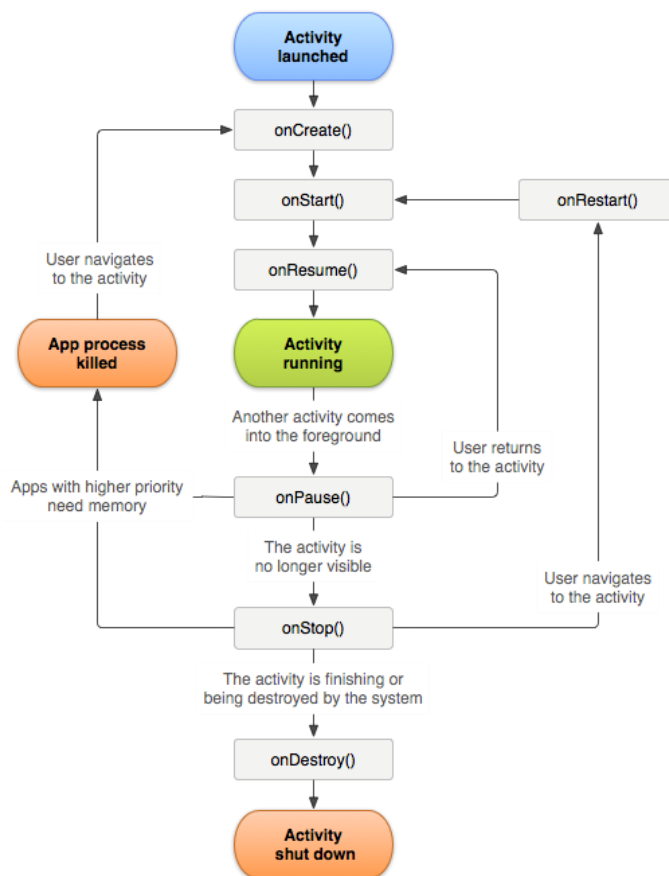
Jak již bylo zmíněno, většina aplikací pro operační systém Android je psána v jazyce Java. Pomocí nástrojů sady Android SDK jsou pak kódy psané v tomto jazyce společně s dalšími „surovinami“ kompilovány do balíčků .apk. Balíčky .apk pak využívají zařízení s tímto operačním systémem k instalaci dané aplikace. [1]

Každá aplikace je složena z několika komponent. V rámci objektově orientovaného programování lze tyto komponenty chápat jako třídy, pomocí kterých může operační systém Android vstupovat do aplikace nebo jako třídy obsluhující určité události.

### 3.2.1 Aktivita

Základní komponentou každé aplikace je aktivita. Aktivitu si lze z uživatelského hlediska představit jako jednu obrazovku, která umožňuje uživateli interakci s aplikací. Každá aplikace má svou hlavní aktivitu, která je zobrazena při spuštění aplikace a pomocí které lze spouštět další aktivity vlastní aplikace, popřípadě aktivity aplikací jiných. Při každém spuštění nové aktivity je předchozí aktivita pozastavena a uložena do zásobníku, který spravuje operační systém. Odtud pak může být - při stisknutí tlačítka zpět - aktivita znovu vyvolána do popředí, v pořadí „last in, first out“. [1] Aktivita také může vyvolat jinou aktivitu za účelem získání dat z uživatelských vstupů. Takto vyvolaná aktivita pak při svém správném ukončení může požadovaná data vrátit.

Životní cyklus aktivity popisuje následující obrázek – obr. 7.



Obr. 7 Životní cyklus aktivity

Klíčové metody, které jsou volány při změně stavu aktivity:

- `onCreate()` – volána při prvním vytvoření aktivity

Využívá se k inicializaci statických dat, vytvoření uživatelského rozhraní, vložení dat do seznamů a podobně

- `onResume()` – volána když se aktivita dostane do popředí

Využívá se k zaregistrování naslouchacích objektů, spuštění videa, audia, animace ...

- `onPause()` – volána když se aktivita dostane do pozadí

Uložení dat, uvolnění nepotřebných zdrojů ...

- `onDestroy()` – volána při ukončení aktivity

Odregistrování naslouchacích objektů, uvolnění všech prostředků [1]

### 3.2.2 Context

*„Context je abstraktní třída, kterou poskytuje operační systém Android. Umožňuje přístup ke zdrojům a třídám, které jsou specifické k dané aplikaci nebo k volání operací na úrovni aplikace jako je spouštění aktivit a odesílání a přijímání záměrů.“ [11]*

### 3.2.3 Intent a intent filter

Intent (dále záměr) je objekt typu zpráva, kterou jednotlivé komponenty aplikace využívají k tomu, aby si vyžádali akci jiné komponenty aplikace a to ať už komponenty aplikace vlastní nebo komponenty jiné aplikace, která je schopná daný záměr zpracovat. Z tohoto pohledu můžeme záměry dělit na záměry implicitní a záměry explicitní. Explicitní záměr je využíván především ke spuštění komponenty vlastní aplikace. Tedy v případech, kdy je přesně znám název třídy, kterou má daný záměr spustit. Záměr implicitní je pak naopak využíván v případech, kdy není požadováno, aby daný záměr zpracovala nějaká konkrétní třída, ale komponenta jakékoliv aplikace, která je schopná daný záměr zpracovat. [1]

K tomu, aby mohl daný záměr vyjádřit, jaká akce má být vyvolána, slouží dvě základní součásti záměru. Akce a k ní příslušná data. Akce může být například typu ACTION\_VIEW (pro spuštění prohlížeče prostředku) nebo ACTION\_PICK (pro výběr dostupné položky z kolekce) a podobně. Data pak mohou být typu URI a (nebo) MIME type (informace kde data získat popřípadě typ dat které mají být zpracovány), Category (informace o typu komponenty, která má daný záměr zpracovat - například CATEGORY\_LAUNCHER), Extras (další data, která mají být volanou komponentou zpracována - data typu klíč-hodnota) a Flags (data jakým způsobem požadovanou komponentu spustit - například vytvořit novou instanci). [1]

Aplikace nebo komponenty systému Android, které chtějí vyslané explicitní záměry zpracovávat, musí deklarovat své filtry záměrů (intent filters). Ty se zapisují do souboru Android-Manifest.xml. [1]

```
<activity android:name="ShareActivity">
  <intent-filter>
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="text/plain"/>
  </intent-filter>
</activity>
```

Obr. 8 Ukázka deklarace intent filtru v souboru AndroidManifest.xml

Posledním, speciálním typem záměrů je Pending Intent. Pending Intent je záměr, který navíc obsahuje informaci, jakou metodou bude spuštěn. [1]

### 3.2.4 Broadcast Receiver

Pomocí objektů, které rozšiřují třídu BroadcastReceiver lze vytvořit objekt, který naslouchá záměrům (intentům) vyslaných pomocí metod typu sendBroadcast(). Lze tak například vytvořit objekt, který bude naslouchat událostem zapnutí zařízení, změně typu připojení k internetu, zapnutí displeje a podobně a na základě těchto událostí bude vykonávat kód, který je definován v jeho jediné metodě onReceive(). BroadcastReceiver musí být opět zapsán v souboru AndroidManifest.xml a existuje pouze po dobu nutnou k provedení metody onReceive().[1]

Alternativou ke globálnímu Broadcast Receiveru je užití Local Broadcast Manageru. LocalBroadcastManager poskytuje metodu registerReceiver(), pomocí níž lze vytvořit objekt typu BroadcastReceiver, který je lokální k dané aplikaci a jehož životnost je závislá na životnosti jeho hostitele. [2] Toho lze využít například v případech, kdy je požadováno provedení nějaké akce pouze za předpokladu, že daný hostitel aktuálně existuje. Odregistrování tohoto objektu je prováděno metodou unregisterReceiver().

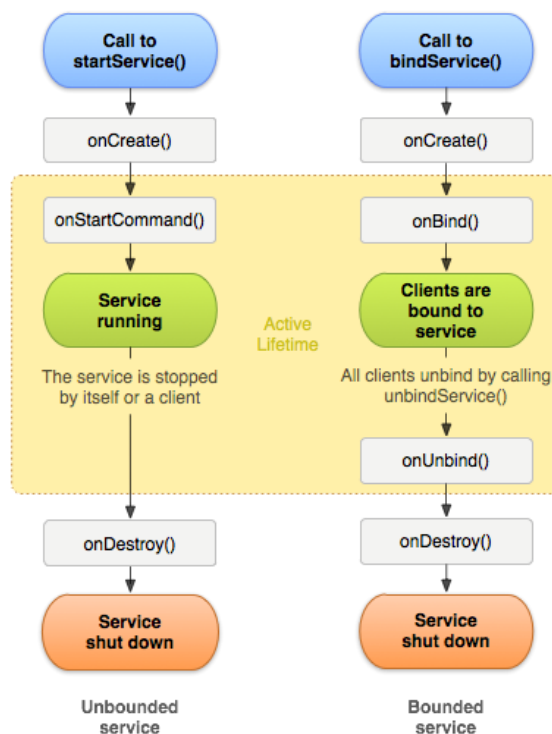
### 3.2.5 Služba

Služby (service) se v operačním systému Android využívají pro vykonávání operací, které běží na pozadí a které mají zůstat spuštěné, i když se uživatel přepne do jiné aplikace. Služba tedy může vykonávat operace, i když není spojena s žádnou aktivitou. Při využití služby je

však nutno brát na zřetel fakt, že i když služba nemusí být spojena s žádnou aktivitou, běží v hlavním vlákně hostujícího procesu. Pro blokující operace, nebo operace vytěžující větší mírou CPU, je tedy vhodné a v některých případech i nutné vytvořit vlastní vlákno.

Službu je možné spustit prostřednictvím metody `startService()`. Takto spuštěná služba pak vykonává operace, které jsou deklarovány v její metodě `onStartCommand()` a nevrací žádné hodnoty komponentě, která službu spustila. Po provedení těchto operací by služba měla být ukončena, například metodou `stopSelf()`. [1]

Službu je také možné provázat s jinou komponentou aplikace pomocí metody `bindService()`. Toto provázání umožňuje službě poskytovat rozhraní typu klient-server a komponenta, která je pomocí tohoto rozhraní se službou provázána pak může službě odesílat požadavky, které má služba zpracovat a přijímat jejich výsledky. Takto provázaná služba pak běží pouze tak dlouho, dokud jsou nějaké komponenty k této službě přihlášeny. Odhlášení je prováděno pomocí metody `unbindService()`. [1]



Obr. 9 Životní cyklus služby

### 3.2.6 AsyncTask

Třída AsyncTask je dostupná od verze Android 1.5 a byla vytvořena pro snadné spouštění operací mimo hlavní uživatelské vlákno. Použitím této třídy je možné provádět operace na pozadí, bez nutnosti vlastní manipulace s vláknem. [2]

*„Systém Android toto vlákno běžící na pozadí sám alokuje a odstraní a udržuje malou frontu úloh, díky které je použití třídy AsyncTask způsobem „spust' a zapomeň“ ještě jednodušší.“*  
[2, str. 269]

Třída AsyncTask je definována třemi generickými parametry <Params, Progress a Result> a je vykonávána ve 4 krocích – onPreExecute(), doInBackground(Params...), onProgressUpdate(Progress...) a onPostExecute(Result...). [1]

Generické typy parametrů třídy AsyncTask:

- Params – parametry určené ke zpracování požadované úlohy metodou doInBackground (například adresy URL)
- Progress – parametry, které jsou předávány v rámci úlohy, které indikují její průběh
- Result – parametry, které se předávají po dokončení úlohy doInBackground [2]

Metody třídy AsyncTask:

- onPreExecute() – metoda, která je volána před metodou doInBackground. Využívá se například k inicializaci progress dialogu
- doInBackground(Params...) – metoda provádějící požadovanou akci na pozadí
- onProgressUpdate(Progress...) – metoda, jejíž pomocí lze indikovat průběh akce na pozadí
- onPostExecute(Result...) – tato metoda je volána po dokončení operace doInBackground. Využívá se například k ukončení progress dialogu a aktualizaci uživatelského prostředí [1]

Metody onPreExecute(),onProgressUpdate(Progress...) a onPostExecute(Result...) jsou pouze metody pomocné a jejich implementace není nutná. [2]

### 3.2.7 Ukládání dat

Každá aplikace potřebuje ke své činnosti ukládat svá data. Systém Android v této oblasti nabízí hned několik možností.

Pro ukládání základních datových typů se strukturou klíč-hodnota je možné využít instanci třídy `SharedPreferences`. Jedná se o soubor, který je spravován frameworkem systému Android a poskytuje rozhraní, pomocí kterého lze zapisovat a číst hodnoty základních datových typů. K zápisu hodnot je využíván objekt `SharedPreferences.Editor`, který se vytvoří zavoláním metody `edit()` na objekt typu `SharedPreferences`. `SharedPreferences.Editor` pak poskytuje metodu `put*(String id, * value)`, kde `*` představuje datový typ, který je ukládán a metodu `commit()`, která vložení nových dat dokončí. Uložená data lze získat pomocí metody `get*(String id, * value)`, kde `*` představuje datový typ, který je požadován. [1]

Další možností jak ukládat svá data je využití relačního databázového systému SQLite. Databáze SQLite je založena na dotazovacím jazyku SQL a je naprogramována v jazyku C. [2] „*Největší rozdíl mezi databází SQLite a ostatními relačními databázemi spočívá v typování dat.*“ [2, str. 435] Při vytváření nové tabulky je sice možné definovat datové typy sloupců, avšak tyto datové typy slouží pouze jako doporučení. „*Databáze SQLite se na tuto svoji funkci odkazuje jako na typování prostřednictvím manifestu – viz jeho popis v dokumentaci: Při typování pomocí manifestu je datový typ vlastností samotné hodnoty, a ne vlastností sloupce, ve kterém je hodnota uložena. Databáze SQLite proto umožňuje uživateli ukládat jakékoliv hodnoty jakýchkoliv datových typů do jakýchkoliv sloupců nezávisle na jejich datovém typu.*“ [2, str. 435]

Každá aplikace může vytvořit svou vlastní databázi, která je následně uložena do adresáře `/data/data/balík-aplikace/databases/(název-databáze)`. Nejlepším způsobem vytvoření a získání přístupu k SQLite databázi je vytvoření podtřídy rozšiřující třídu `SQLiteOpenHelper`. K vytvoření této instance je nutné definovat její konstruktor a přepsat její metody `onCreate()` a `onUpdate()`. Konstruktor této třídy přijímá objekt typu `Context`, název databáze, instanci třídy `CursorFactory` (nejčastěji `null`) a číslo verze databáze. Metody `onCreate()` a `onUpdate()` pak předávají objekt typu `SQLiteDatabase` a je v nich nutno definovat logiku vytvoření respektive aktualizace databáze. Takto vytvořený objekt typu `SQLiteOpenHelper` je pak možno využít ke čtení nebo zápisu dat do databáze. Poskytuje totiž metody `getReadableDatabase()`, která otevře databázi pouze ke čtení a `getWritableDatabase()`, pro získání přístupu k zápisu do databáze. [2]

Ukládání dat do databáze lze provést dvěma způsoby. Využitím metody `execSQL()`, které se využívá i při vytváření databáze, s využitím syntaxe jazyka SQL nebo pomocí metod `insert()`,

respektive `update()` objektu `SQLiteDatabase`, které přijímají název tabulky, `nullColumnHack` (může být `null`) a objekt `ContentValues`. Tedy data určená k zápisu, s informacemi do kterých sloupců jednotlivá data uložit. Obdobně je pak možné data z databáze získat. Voláním metody `rawQuery()` (získání dat přímo - užitím příkazu `SELECT`), nebo pomocí metody `query()`, kde je dotaz opět vytvářen pomocí jednotlivých komponent. Obě zmíněné metody vrací objekt typu `Cursor`, který poskytuje prostředky potřebné k průchodu získaných výsledků. [2]

Poslední možností pro čtení a zápis dat (pomineme-li možnost získávání dat ze vzdálené databáze, `Cloud` a podobně) je využití binárních souborů. Binární soubory je možno ukládat do interní nebo externí paměti. Externí paměť se v tomto případě rozumí paměť, která je výrobcem zařízení definována jako externí a neslouží přímo potřebám systému `Android`. Nemusí se tedy jednat o paměť, která je k dispozici například na kartě `SD card`, ale může jít i o paměť zařízení, která je vyčleněna pro potřeby uživatele zařízení. Oba zmíněné typy paměti mají svá specifika, která jsou přehledně vysvětlena na stránkách `Android Developers` [13] a které je vhodné, před rozhodnutím kam binární soubory zapisovat, znát. Samotné čtení a zápis do binárních souborů je pak prováděn obdobně jako u desktopových aplikací napsaných v jazyce `Java`, tedy prostřednictvím knihovny `java.io`. [1]

### 3.2.8 Zdroje

Zdroje (`resources`) jsou statická data (prostředky aplikace), která zdrojový kód využívá k sestavení aplikace a která by měla být z praktických důvodů oddělena od samotného kódu. Jak již bylo zmíněno, operační systém `Android` využívá mnoho typů zařízení s rozličnými hardwarovými konfiguracemi, ve všech zemích po celém světě. A právě koncepce oddělení surovin od zdrojového kódu umožňuje vývojářům vytvářet aplikace, které se jednotlivým konfiguracím mohou přizpůsobit. [1]

`Resources` se ukládají do podadresářů adresáře `/res`, který je během kompilace programu zpracován a přibalen do balíčku aplikace `.apk`. [2]

Struktura těchto podadresářů je následovná:

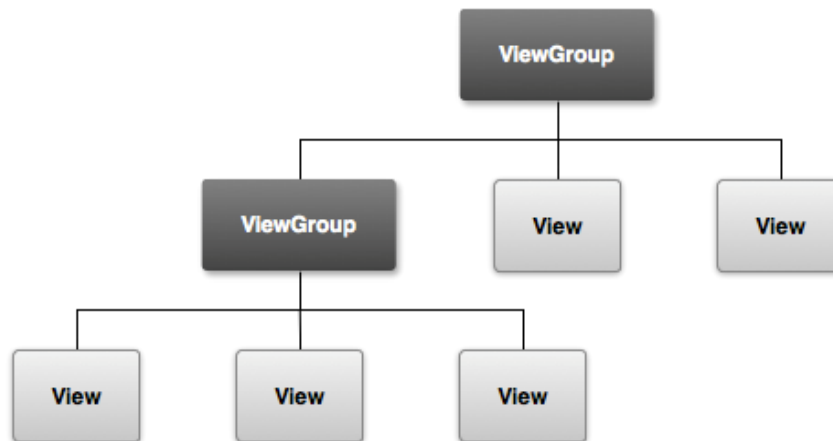
- podsložky `/anim` a `/animator` – pro ukládání XML souborů definujících animace
- podsložka `/color` – XML soubory definující seznam barev
- podsložka `/drawable` – soubory `.png`, `.9.png`, `.jpg`, `.gif`
- podsložka `/layout` – XML soubory definující rozložení uživatelské rozhraní

- podsložka /menu – XML soubory definující aplikační menu
- podsložka /raw – jakékoliv soubory ukládané v jejich „surové“ podobě
- podsložka /values – XML soubory obsahující hodnoty (řetězce, barvy, pole, styly, rozměry)
- podsložka /xml – další XML soubory [1]

### 3.2.9 Uživatelské rozhraní

Při vývoji uživatelského prostředí (UI) je nezbytné vzít v úvahu několik základních věcí. První a nejdůležitější z nich je, že vyvíjíme aplikace především pro mobilní zařízení, které mají většinou malou zobrazovací plochu, na kterou je nutné vměstnat veškerou logiku aplikace. Uživatelské rozhraní by tak mělo být jednoduché a mělo by být rozčleněno podle jednotlivých akcí. Zároveň však nesmí být zapomenut fakt, že operační systém Android nejsou jen mobilní telefony. Android dnes pohání především mobilní telefony a tablety. Trendem dnešní doby je i implementace tohoto systému do zařízení jako jsou televizory, set-top boxy, herní konzole, ale i lednice, pračky, mikrovlnné trouby a podobně. Chceme-li tedy, aby naše aplikace podporovala i zařízení tohoto typu, měla by být aplikace pro tato zařízení přizpůsobena. Toho lze dosáhnout především využitím základních témat aplikací, která přímo nabízí vývojové prostředí Eclipse se zásuvným modulem ADT, implementací zdrojů pro různé rozměry obrazovek a v neposlední řadě také využitím fragmentů, které jsou schopny se zobrazovat nebo skrývat v závislosti na velikosti obrazovky. Aplikace navíc nemusí využívat pouze prostor, který je jí dán pro zobrazení obsahu (Content Area). Moderní aplikace hojně využívají i panel akcí (Action Bar) a zásuvný navigační panel (Navigation Drawer).

Základním prostředkem pro vytváření aplikačního UI jsou grafické objekty (widgety), které jsou potomky třídy View. [1] Ačkoli je technicky možné připojovat widgety jednotlivým aktivitám pouze prostřednictvím kódu jazyka Java, obvyklejší přístup spočívá ve využití souborů s definicemi rozložení ve formátu XML, které jsou uloženy ve složce /res/layout. [2] Ke každé aktivitě náleží právě jeden z těchto souborů definující strukturu a vlastnosti prvků, které daná aktivita zobrazuje. Jednotlivé elementy těchto souborů zastupují prvky View a ViewGroup. Použité atributy pak definují parametry těchto prvků.



Obr. 10 Hierarchie UI

Základní prvky View, které je možné použít k vývoji aplikace:

- TextView – needitovatelné textové popisky
- EditText – editovatelná textová pole
- ListView – vertikální seznam
- ScrollView – kontejner pro posouvání obsahu
- WebView – prohlížeč založený na jádře Webkit/Chromium
- Button – tlačítko
- CheckBox – zaškrtačací políčko
- RadioButton – umožňuje výběr jedné položky z nabídky
- Spinner – rozevíratelný seznam
- Switch – vypínač (dostupný od API 14) [2]

Základní kontejnery:

- LinearLayout – jednotlivé prvky jsou dle atributu řazeny do sloupce nebo řádku
- RelativeLayout – prvky jsou umístěny relativně k okrajům kontejneru nebo okrajům jiného prvku
- TableLayout – tabulkové rozložení prvků
- GridLayout – prvky jsou umístěny do mřížky tvořené nekonečně tenkými čarami
- FrameLayout – umožňuje překrývání prvků podle osy z
- NavigationDrawer – zásuvný navigační panel
- Fragment – umožňuje přizpůsobit vzhled aplikace velikosti zobrazovací plochy [2]

Ostatní často užívané prvky:

- Kontextové menu – zobrazí se při dlouhém stisknutí položky
- Dialogové okno – upozornění, které je nutné potvrdit
- Toast – upozornění zobrazující definovanou dobu
- ProgressBar – zobrazuje progres akce
- Upozornění v oznamovací liště

## **II. PRAKTICKÁ ČÁST**

## 4 POPIS APLIKACE

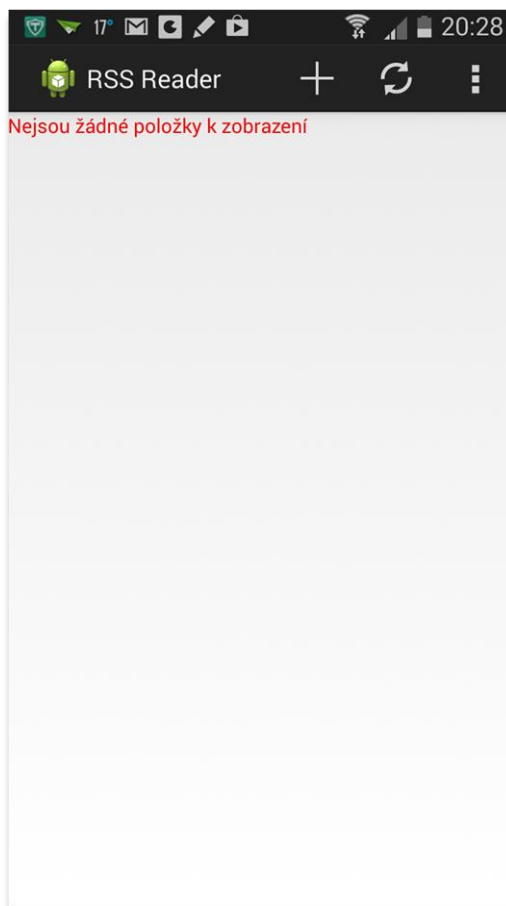
Aplikace byla vyvíjena pro zařízení s operačním systémem Android ve verzi 2.3 a vyšší a podporuje všechna základní rozlišení obrazovek. Uživatelské prostředí je postaveno na fragmentech, které umožňují přizpůsobit vzhled aplikace velikosti zobrazovací plochy. Aplikace spravuje uživatelem definované RSS kanály, které automaticky stahuje v uživatelem nastavených časových intervalech a zpřístupňuje jejich obsah k off-line užití. O výsledcích stahování pak informuje pomocí upozornění v oznamovací liště. Stahování RSS kanálů je prováděno pomocí služby, která běží na pozadí aplikace. Služba je automaticky spouštěna i po restartu zařízení, čímž je zabezpečeno stahování v nastavených intervalech, i když bylo zařízení vypnuto a znovu zapnuto. Pro stažení zdrojů je vyžadováno připojení k síti Internet. Aby bylo možné předejít k nechtěnému stahování dat prostřednictvím datového spojení mobilního operátora, byla v aplikaci implementována možnost nastavit stahování pouze pomocí připojení k síti Wi-Fi. Výsledky stahování jsou ukládány do databáze SQLite, která uchovává stažená data po dobu nastavenou uživatelem (maximálně 1 týden). Tímto je zabezpečena minimální paměťová náročnost aplikace.

Vzhled aplikace vychází z tématu Theme Holo Light DarkActionBar, který přímo poskytuje ADT plugin. Základní ovládací prvky aplikace jsou umístěny v panelu akcí. Zastupují je ikony „přidat“, „aktualizovat“ a „menu“. Ikony jsou v tomto panelu zobrazovány v závislosti na dostupném místě v panelu. Pokud by zařízení nemohlo některou z těchto ikon z důvodu nedostatku místa zobrazit, zobrazí se nedostupná funkce po stisknutí hardwarového tlačítka menu. Správa zdrojů je umožněna pomocí vysouvacího navigačního panelu, který se zobrazí přejetím po obrazovce od levé strany směrem ke středu zobrazovací plochy.

Aplikace využívá pro zpracování (parsování) stažených zdrojů knihovnu `horrorss.jar` [14], kterou je možné využít pod licencí GNU Lesser GPL. Tato knihovna by měla být schopna zpracovat feedy formátu RSS 0.9x, RSS 1.0, RSS 2.0 a feedy formátu Atom. Právě kvůli této široké kompatibilitě byla uvedená knihovna v aplikaci použita. Při testování aplikace jsem však bohužel zaznamenal (v některých případech) nekompatibilitu s formáty Atom. Formáty RSS, jsou však zpracovávány bez problémů.

## 4.1 První spuštění

Aplikace neobsahuje žádné předdefinované RSS kanály. Při prvním spuštění je tedy uživateli zobrazena pouze plocha s informací o nedostupnosti jakýchkoliv položek k zobrazení. Uživatel přidá nový RSS kanál pomocí ikony „+“, která se nachází v panelu akcí.

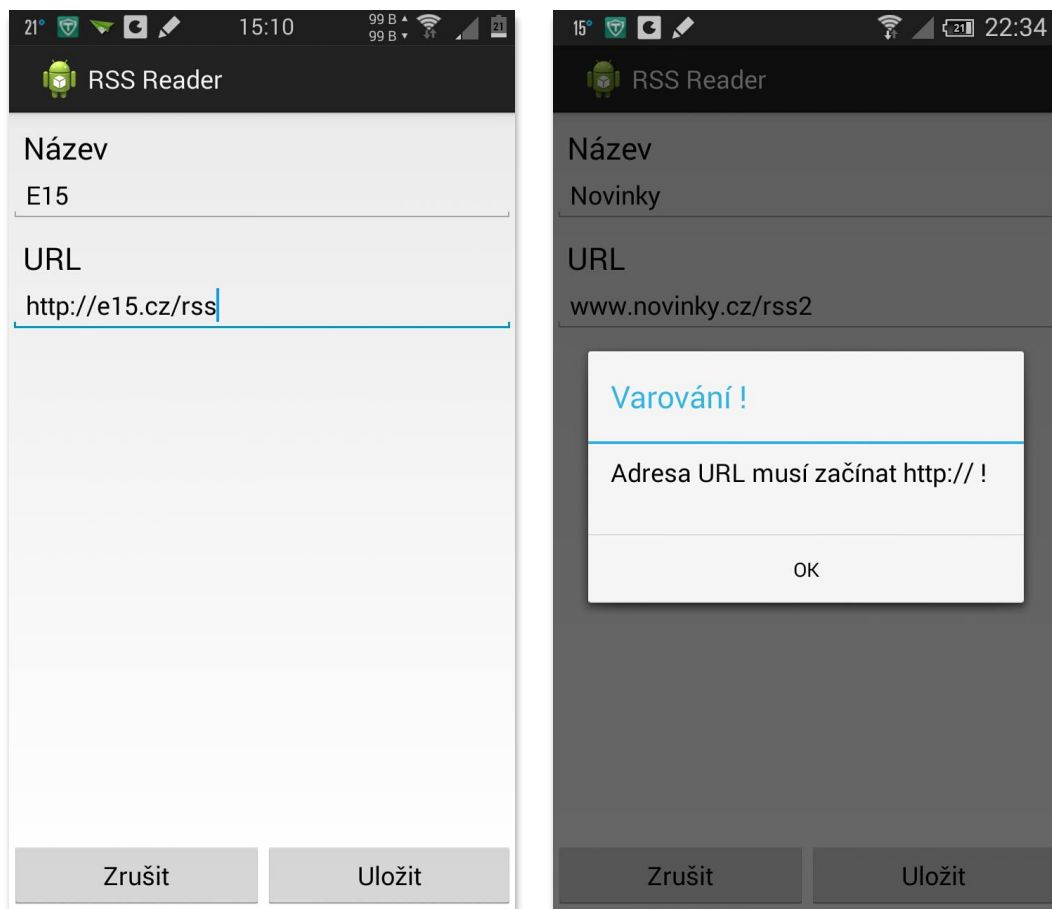


Obr. 11 První spuštění aplikace

## 4.2 Přidání zdrojů

Po stisknutí ikony „přidat“ je spuštěna nová aktivita, která umožní uživateli přidat nový zdroj. Zobrazí se dvě editovatelné oblasti, jejich popisky a tlačítka „uložit“ a „zrušit“. Pro uložení nového zdroje je nutno vyplnit obě položky. Položka „název“ je povinná z důvodu možnosti pozdější manipulace se zdrojem. Adresu URL zdroje je nutno napsat ve formátu `http://zdroj`. Zadání sekvence `http://` a názvu zdroje je kontrolováno algoritmem aplikace, který v případě zjištění nedostatku zobrazí dialogové okno s varováním, které popisuje zjištěnou chybu. Při nesplnění podmínek pro řádné uložení zdroje (viz zmíněné podmínky) není

uložení zdroje uživateli umožněno. Duplicitní adresy URL jsou opět kontrolovány algoritmem aplikace a nejsou uloženy.



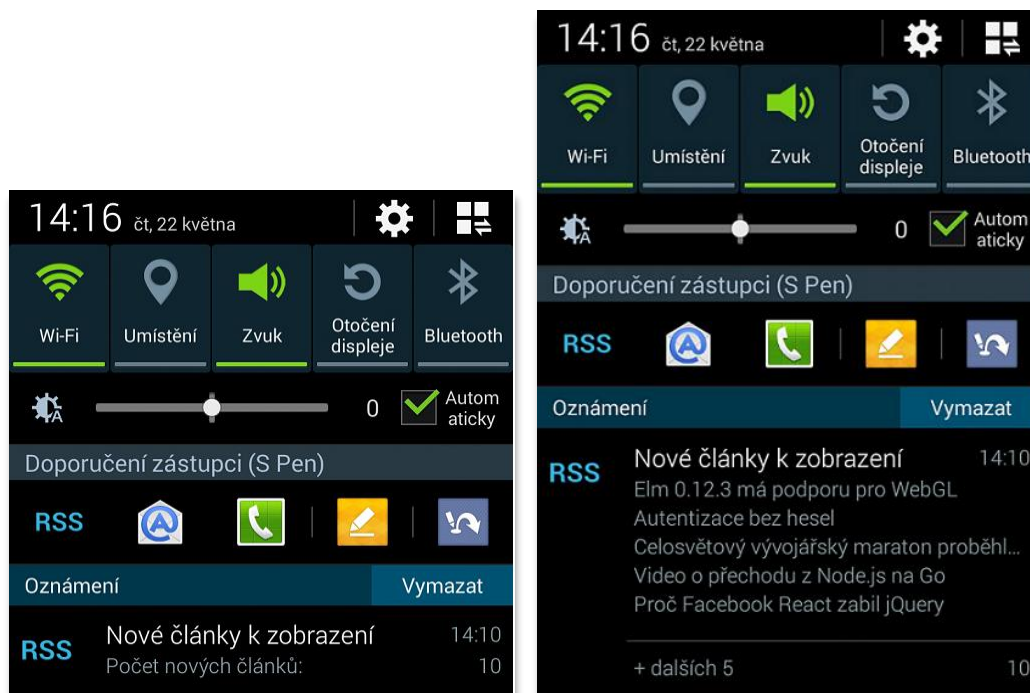
Obr. 12 Přidání zdroje a hlášení v případě chybného zadání

### 4.3 Aktualizace článků

Aktualizace článků je spouštěna třemi způsoby.

První způsob je automatická aktualizace pomocí služby na pozadí – při nespouštění aplikaci. O spuštění aktualizace touto cestou není uživatel nijak informován. Pokud však má uživatel nastaveno stahování pouze pomocí Wi-Fi a zařízení není k této síti připojeno, zobrazí se upozornění typu toast, které uživatele informuje o vzniklé situaci a vybídne ho ke změně nastavení aplikace. Upozornění typu toast se však využívá pouze u informací, které nemají nepostradatelný informační charakter a zobrazuje se pouze po definovanou dobu, po které samo zmizí. Toto upozornění tedy nijak uživatele neobtěžuje. Příklad upozornění typu toast je znázorněn na obr. 15. Je-li však stahování úspěšně dokončeno a jsou-li dostupné nové články, je o této situaci uživatel informován pomocí upozornění v oznamovací liště. Po

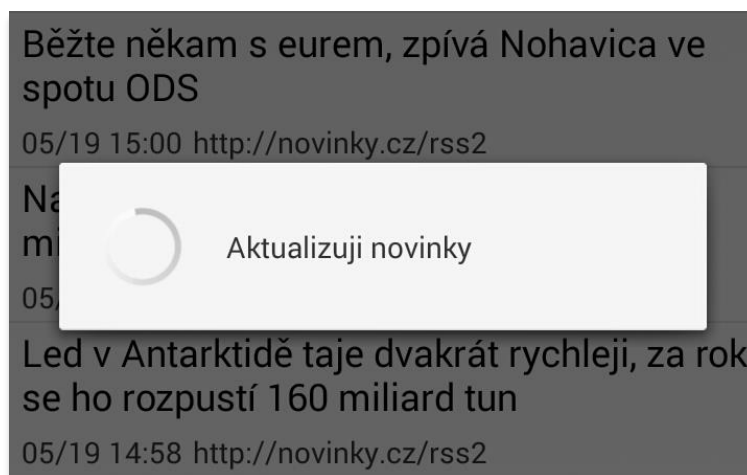
kliknutí na toto upozornění je spuštěna hlavní aktivita aplikace, která zobrazí titulky stažených článků, čas jejich publikování a adresu zdroje. Viz kapitola 4.4.



Obr. 13 Upozornění na nové články v základním a rozšířeném tvaru

Články jsou také aktualizovány po přidání nového zdroje. Stahování článků po přidání nového zdroje je opět závislé na dostupnosti připojení k internetu a nastavení aplikace. Aktualizace je však spuštěna v případě, kdy je aplikace na popředí a uživatel tedy musí být informován o prováděné aktualizaci. Toho je docíleno zobrazením dialogového okna progressBar, které je znázorněno na obr. 14 a které zmizí, jakmile je aktualizace článků dokončena. Po dokončení aktualizace je provedena aktualizace UI a je opět zobrazeno upozornění v oznamovací liště. Není-li aktualizace z důvodu nastavení aplikace nebo nedostupnosti internetu možná, je o této situaci uživatel informován pomocí upozornění typu toast, které zobrazuje důvod, proč nebyla aktualizace provedena.

Poslední možností vyvolání aktualizace článků je ruční spuštění aktualizace pomocí stisknutí ikony „aktualizovat“, která je umístěna v panelu akcí. Aktualizace spuštěná touto cestou je prováděna stejným způsobem, jakým je prováděna aktualizace po přidání nového zdroje.



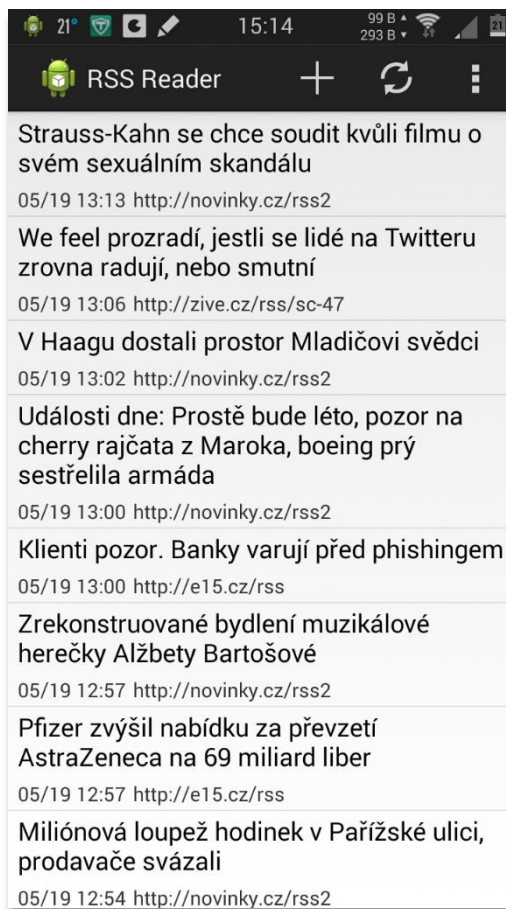
Obr. 14 Dialogové okno progressBar



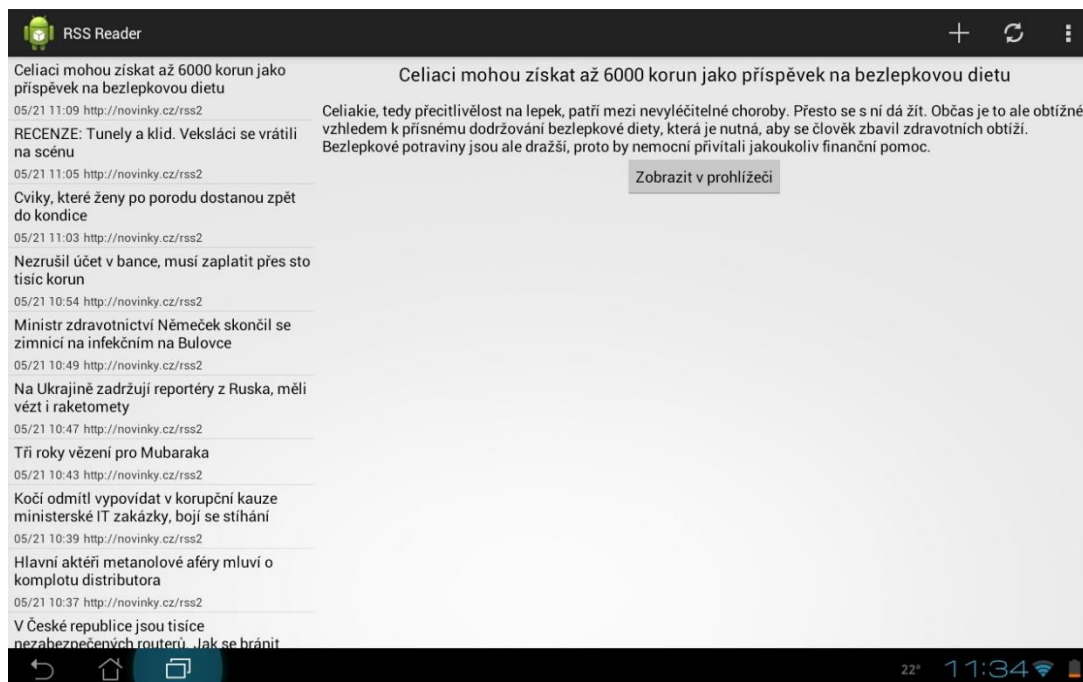
Obr. 15 Upozornění typu toast

#### 4.4 Seznam článků – hlavní aktivita aplikace

Hlavní okno aplikace je založeno na fragmentech, které jsou zobrazovány v závislosti na velikosti zobrazovací plochy zařízení, na kterém aplikace běží. Pokud je aplikace spuštěna na zařízení s menší zobrazovací plochou – na telefonu – a jsou-li v databázi již uloženy nějaké záznamy, zobrazuje hlavní okno aplikace seznam nadpisů stažených článků, datum a čas jejich publikování a adresu jejich zdroje. Viz obr. 16. Položky jsou tříděny podle data a času jejich publikování. Detail jednotlivé položky je v tomto případě zobrazen po kliknutí na položku, vytvořením nové aktivity. Pokud je však aplikace spuštěna na zařízení s větší zobrazovací plochou – tablet a větší zařízení – je seznam článků zobrazen v levé části obrazovky a v pravé části obrazovky je pak zobrazován detail aktuálně zobrazované položky viz obr. 17.



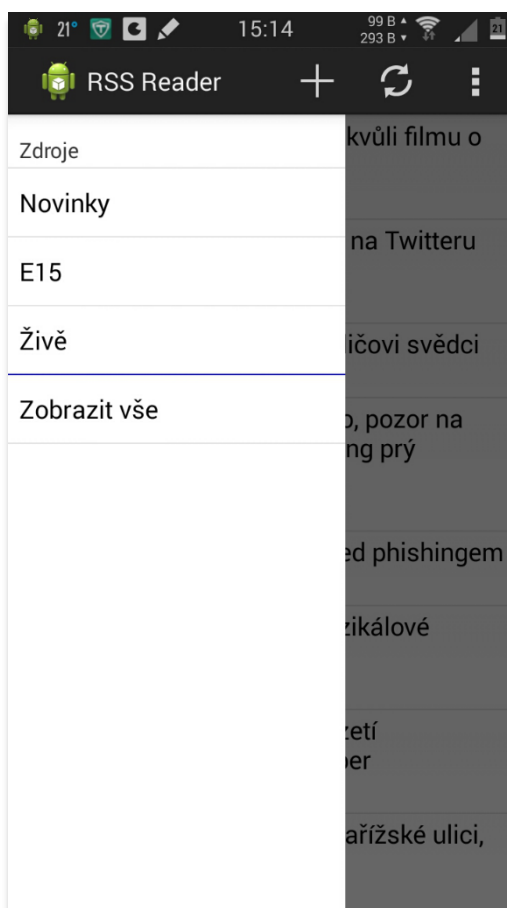
Obr. 16 Hlavní okno aplikace spuštěné na telefonu se 4,5 palcovou obrazovkou



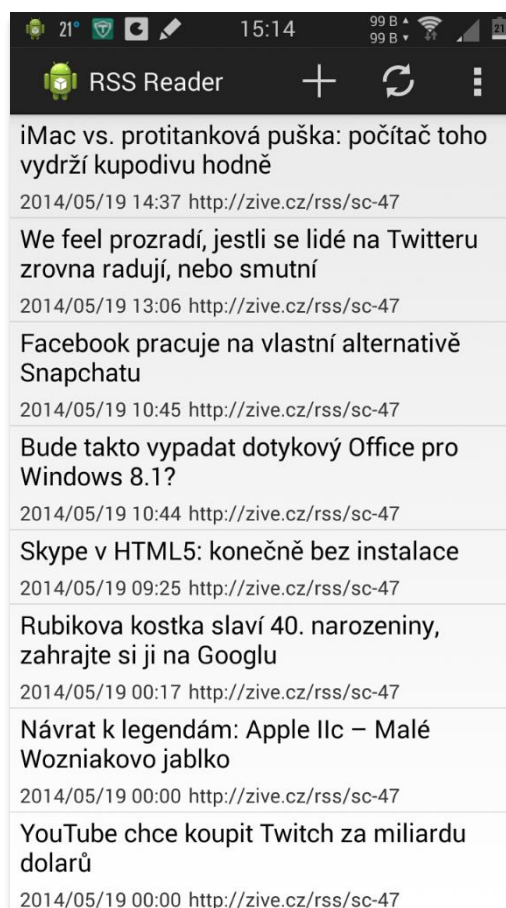
Obr. 17 Hlavní okno aplikace spuštěné na tabletu s 10 palcovou obrazovkou

## 4.5 Manipulace se zdroji

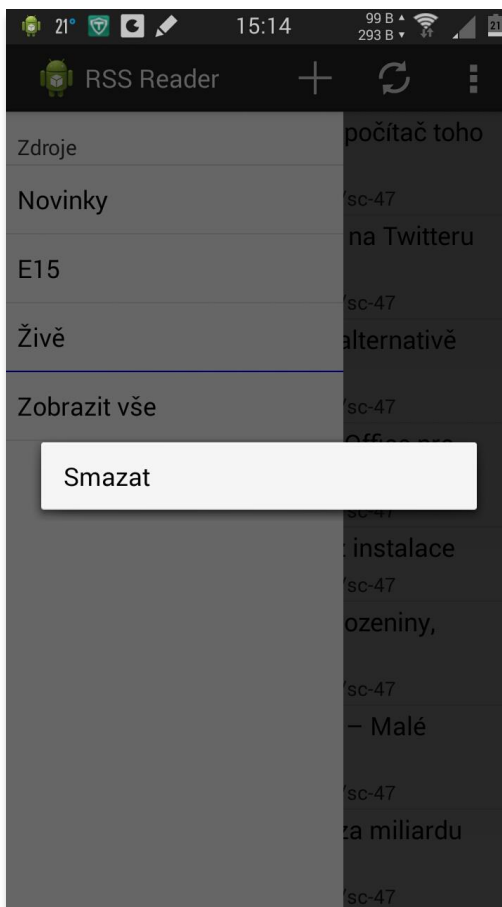
Manipulace se zdroji je prováděna pomocí levého výsuvného panelu (viz obr. 18), který lze vyvolat přejetím po obrazovce od levé strany směrem ke středu zobrazovací plochy. Výsuvný panel zobrazuje seznam jednotlivých zdrojů a položku „Zobrazit vše“. Po krátkém kliknutí na zdroj, je daný zdroj vyfiltrován a hlavní obrazovka aplikace pak zobrazuje pouze položky tohoto zdroje. Viz obr. 19. Zrušení filtru lze provést kliknutím na položku „Zobrazit vše“ nebo restartem aplikace. Po dlouhém kliknutí na zdroj je zobrazeno kontextové menu, které nabídne smazání tohoto zdroje. Viz. obr. 20. Články ze smazaného zdroje jsou však nadále uchovávány pro jejich off-line užití, až do chvíle, dokud nejsou smazány algoritmem aplikace, po uplynutí doby uchování článků, která je nastavena uživatelem. Přidání nového zdroje je prováděno postupem popisovaným v kapitole 4.2.



Obr. 18 Navigační panel



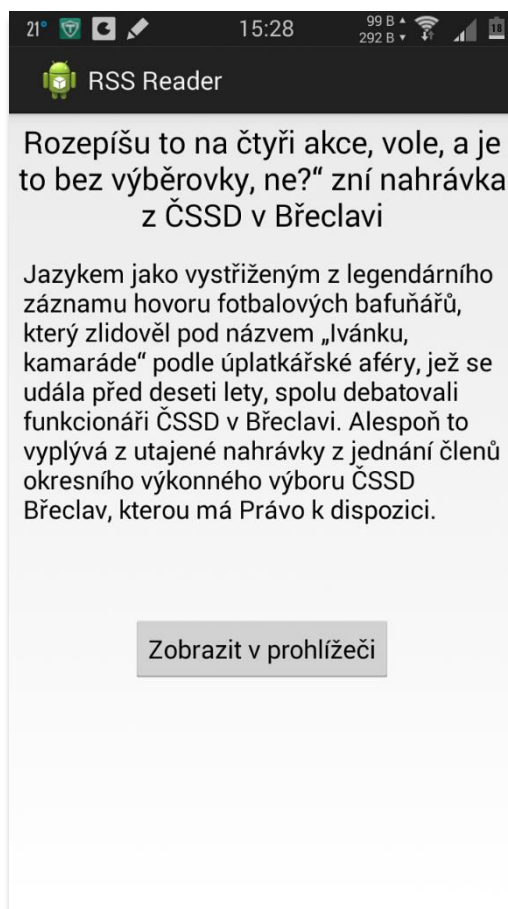
Obr. 19 Seznam vyfiltrovaného zdroje



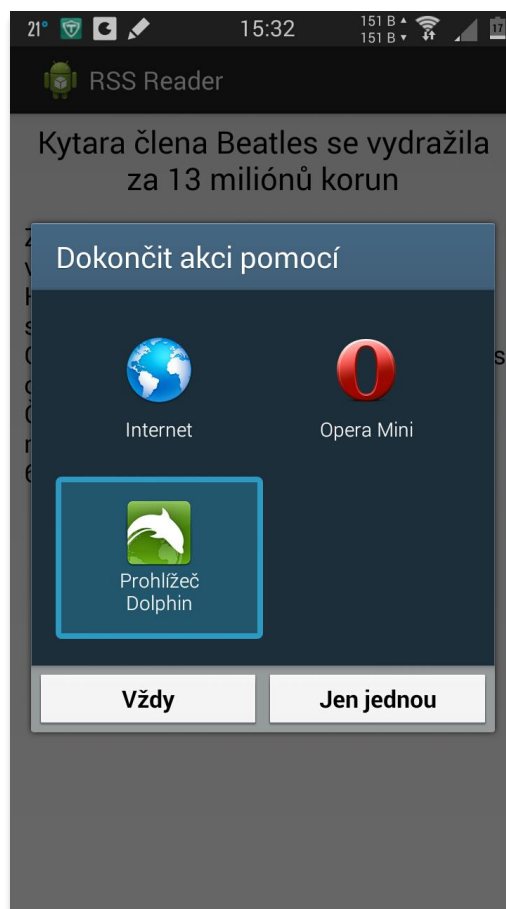
Obr. 20 Kontextové menu smazat

#### 4.6 Detail položky

Po kliknutí na jednotlivou položku se zobrazí její obsah – položka description. Na telefonech je tato činnost prováděna novou aktivitou, na větších zařízeních je položka zobrazena v pravém panelu aplikace. Pod popisem této položky je zobrazeno tlačítko „Zobrazit v prohlížeči“. Viz obr. 21. Toto tlačítko umožní zobrazení originálního článku pomocí internetového prohlížeče. Je-li v zařízení pouze jeden internetový prohlížeč, dojde ihned k zobrazení tohoto článku v tomto prohlížeči. Pokud je však v zařízení nainstalováno více internetových prohlížečů, je uživateli nejdříve zobrazeno dialogové okno, které umožní výběr, ve kterém prohlížeči má být daná položka zobrazena. Viz obr. 22. Pro tuto činnost je již vyžadováno aktivní připojení k internetu a zobrazování článků je prováděno on-line.



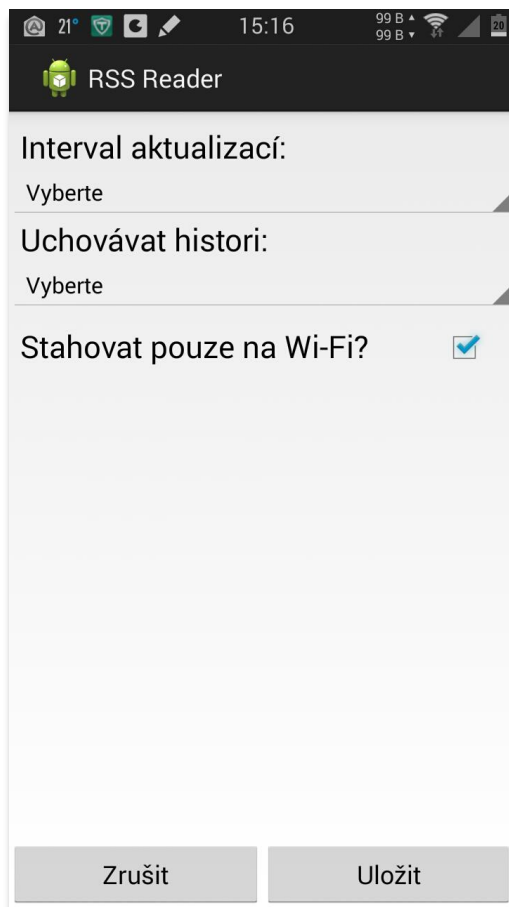
Obr. 21 Detail položky



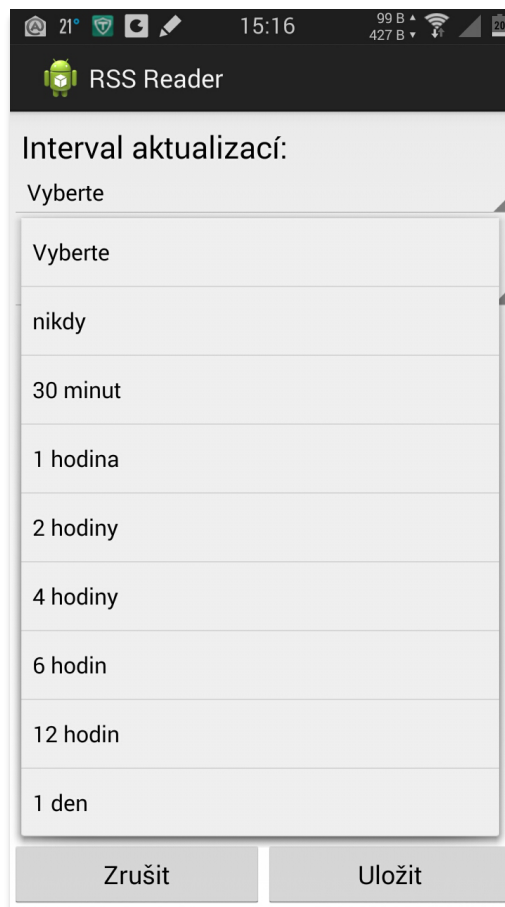
Obr. 22 Výběr prohlížeče

## 4.7 Nastavení

Nastavení aplikace je vyvoláno po stisknutí ikony nastavení v panelu akcí. Aplikace umožňuje nastavit interval automatických aktualizací, dobu, po kterou mají být uchovávány články a nastavit stahování pouze prostřednictvím připojení k síti Wi-Fi. Viz obr.23. Interval aktualizací a dobu ukládání článků lze nastavit v předdefinovaných intervalech pomocí rozbalovacího seznamu. Viz obr.24. Nastavení stahování pouze na Wi-Fi je zprostředkováno pomocí zaškrtačacího pole. Hodnoty nastavení jsou uloženy až po stisknutí tlačítka „Uložit“. O uložení hodnot je uživatel informován oznámením typu toast.



Obr. 23 Obrazovka nastavení



Obr. 24 Rozevřený seznam

## ZÁVĚR

Hlavním cílem této práce bylo vytvořit aplikaci pro mobilní platformu, která bude číst informace z kanálů RSS a zpřístupňovat jejich obsah k off-line užití. Dílčím cílem pak byla implementace stahování těchto dat na pozadí aplikace, automatická správa kanálů a vytvoření uživatelské dokumentace.

Všechny body zadání byly splněny a aplikace byla navíc rozšířena o některé funkce, které uživateli zvýší komfort při užívání této aplikace. Mezi tyto funkce patří především automatické spuštění služby, která zabezpečí stahování zdrojů i po restartu zařízení, automatické mazání starších článků, po uživatelem definované době, přehledná správa kanálů pomocí výsuvného navigačního panelu nebo systém upozornění na nové články.

Aplikace byla vyvíjena dle současného trendu způsobu vytváření aplikací pro mobilní operační systém Android. Bylo využito prvků, které jsou dostupné pro nejnovější verze tohoto systému a fragmentů, které umožňují přizpůsobení vzhledu aplikace velikosti zobrazovací plochy. Současně však byly použity zpětně kompatibilní knihovny, které umožňují instalaci a správnou funkci této aplikace i na zařízeních se staršími verzemi systému. Aplikaci je tak možno spustit na zařízeních s verzemi Android 2.3 a vyšší, což činí tuto aplikaci použitelnou v naprosté většině zařízení, které využívají tuto platformu.

Testování a ladění aplikace probíhalo jak v emulovaném prostředí, tak na fyzických zařízeních. Emulovány byly verze systému 2.3.3 a 4.1.2. Fyzická zařízení byla reprezentována telefony Samsung Galaxy Note II s custom ROM ve verzi 4.4.2, Sony Xperia U s oficiálním firmware ve verzi 4.0.4 a tabletem Asus Transformer Pad s custom ROM ve verzi 4.0.3. Finální verze aplikace na těchto konfiguracích běžela bez problémů.

Jediný problém se vyskytl při čtení některých RSS feedů ve formátu Atom. I když nebyla tato funkčnost ze zadání bakalářské práce vyžadována, snažil jsem se vytvořit kompaktní aplikaci, která by tento formát podporovala. Z tohoto důvodu jsem se rozhodl použít ke zpracování stažených zdrojů knihovny `horrorss.jar`, kterou je možné použít pod licencí GNU Lesser GPL a která podporuje jak formáty RSS, tak formáty Atom. Čtení některých zdrojů ve formátu Atom se však pomocí této knihovny nezdařilo. Položky zdrojů, které neobsahují potřebné údaje nebo zdroje, které zmíněná knihovna nezpracuje, nejsou proto ukládány.

Pokud by byla aplikace nadále vyvíjena, bylo by dle mého názoru vhodné doladit pod odborným dohledem její vzhled, vytvořit kontextové menu pro editaci uložených zdrojů (aktuálně lze zdroje pouze přidávat a mazat), umožnit uživateli označovat oblíbené články, popřípadě vyvinout vlastní parser, který by byl schopen čít zdroje všech formátů a ošetřoval by případy čtení nesprávně publikovaných feedů. Vývoj tohoto parseru, který by ošetřoval všechny zmíněné situace, by však musel být prováděn dlouhodobě.

**SEZNAM POUŽITÉ LITERATURY**

- [1] Android Developers[online]. [vid. 2014-05-26]. Dostupné z: <http://developer.android.com>
- [2] GRANT, Allen. Android 4: Průvodce programováním mobilních aplikací. Brno: COMPUTER PRESS, 2013. ISBN 978-80-251-3782-6
- [3] SCHILDT, Herbert. JAVA 7: Výukový kurz. Brno: COMPUTER PRESS, 2012. ISBN 978-80-251-3748-2
- [4] MEIER, Reto. Profesional Android 4 Application Development. New York City: John Wiley & Sons, Inc. 2012. ISBN 978-1-118-10227-5. Dostupné také z: [http://lecturer.eepis-its.edu/~yuliana/Android/Buku/professional\\_android\\_4\\_application\\_development.pdf](http://lecturer.eepis-its.edu/~yuliana/Android/Buku/professional_android_4_application_development.pdf)
- [5] KOSEK, Jiří. XML pro každého: Podrobný průvodce. Praha: Grada Publishing, 2000. ISBN 80-7169-860-1. Dostupné také z: <http://www.kosek.cz/xml/xmlprokazdeho.pdf>
- [6] BERKMAN CENTER FOR INTERNET & SOCIETY AT HARVARD UNIVERSITY[online]. [vid. 2014-05-26]. Dostupné z: <http://cyber.law.harvard.edu>
- [7] Jak vytvořit RSS kanál v PHP: Co je to RSS. Wwww.samuraj-cz.com [online]. 2006, 2008-10-12 [vid. 2014-05-26]. Dostupné z: <http://www.samuraj-cz.com/clanek/jak-vytvorit-rss-kanal-v-php>
- [8] Android[online]. [vid. 2014-05-26]. Dostupné z: <http://android.com>
- [9] Android: A visual history. THE VERGE [online]. 2011-12-7 [vid. 2014-05-26]. Dostupné z: <http://www.theverge.com/2011/12/7/2585779/android-history>
- [10] Using the Emulator. Android Developers[online]. [cit. 2014-05-26]. Dostupné z: <http://developer.android.com/tools/devices/emulator.html>
- [11] Context. Android Developers[online]. [cit. 2014-05-26]. Dostupné z: <http://developer.android.com/reference/android/content/Context.html>
- [12] Context. Android Developers[online]. [cit. 2014-05-26]. Dostupné z: <http://developer.android.com/reference/android/content/Context.html>
- [13] Saving File. Android Developers[online]. [vid. 2014-05-26]. Dostupné z: <http://developer.android.com/training/basics/data-storage/files.htm>

- [14] Horrorss[online]. [vid. 2014-05-26]. Dostupné z:  
<https://code.google.com/p/horrorss/>

## SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

- ADT Zásuvný modul pro vývojové prostředí, určený k vývoji Android aplikací. Z anglického Android Development Tool.
- API Rozhraní pro programování aplikací. Z anglického Application Programming Interface.
- CDMA Přenos dat pomocí více signálů jediného média. Z anglického Code Division Multiple Access.
- CPU Mikroprocesor. Z anglického Central Processing Unit.
- GNU Licence pro svobodný software.
- GPL Všeobecně veřejná licence. Z anglického General Public Licence.
- GSM Systém pro mobilní komunikaci. Z francouzského Groupe Spécial Mobile.
- HDR Systém umožňující zvětšit dynamický rozsah expozice fotografie mezi světlým a tmavým bodem. Z anglického High Dynamic Range.
- HTML Jazyk pro hypertext. Z anglického HyperText Markup Language.
- ISO Mezinárodní organizace pro standardizaci. Z anglického International Organization for Standardization.
- JIT Metoda překladač pro urychlení běhu kódu. Z anglického Just In Time.
- LCD Obrazovka z tekutých krystalů. Z anglického Liquid Crystal Display.
- MIME Označení typu média.
- MMS Multimediální zpráva. Z anglického Multimedia Messaging Service.
- MMU Jednotka správy paměti. Z anglického Memory-Management Unit.
- NFC Bezdrátová technologie pro přenos dat na krátkou vzdálenost. Z anglického Near Field Communication.
- OTA Systém aktualizací softwaru pomocí bezdrátové sítě. Z anglického Over The Air.
- P2P Přímá komunikace uživatel – uživatel. Z anglického Peer to Peer.
- QWERTY Označení plnohodnotné klávesnice.

---

RAM	Paměť s přímým přístupem. Z anglického Random-Access Memory.
ROM	Typ paměti, která je naprogramována výrobcem zařízení a která uchovává data nezávisle na napájení. Pro účely této práce software, který je na tento typ paměti nahrán. Z anglického Read-Only Memory.
RSS	Formát pro výměnu obsahu založený na XML. Z anglického Really Simple Syndication.
SD	Typ paměťové karty, kterou využívají přenosná zařízení. Z anglického Secure Digital.
SDK	Sada vývojových nástrojů. Z anglického Software Development Kit.
SIM	Typ paměťové karty, která je určena k identifikaci účastníka v mobilní síti. Z anglického Subscriber Identity Module.
SIP	Protokol pro přenos signalizace v internetové telefonii. Z anglického Session Initiation Protocol.
SMS	Textová zpráva. Z anglického Short Message Service.
SQL	Standardizovaný dotazovací jazyk. Z anglického Structured Query Language.
UI	Uživatelské rozhraní. Z anglického User Interface.
URI	Identifikátor zdroje. Z anglického Uniform Resource Identifier.
URL	Řetězec znaků specifikující umístění zdroje na Internetu. Z anglického Uniform Resource Locator.
VM	Softwarově emulované virtuální zařízení. Z anglického Virtual Machine.
XML	Rozšiřitelný značkovací jazyk. Z anglického Extensible Markup Language.

**SEZNAM OBRÁZKŮ**

Obr. 1 Ikona RSS.....	12
Obr. 2 Zastoupení mobilních operačních systému na trhu s mobilními telefony.....	14
Obr. 3 Podíl jednotlivých verzí systému Android na trhu ke dni 1.4.2014.....	19
Obr. 4 Architektura operačního systému Android.....	19
Obr. 5 Android emulátor.....	22
Obr. 6 Android Virtual Device Manager.....	23
Obr. 7 Životní cyklus aktivity.....	24
Obr. 8 Ukázka deklarace intent filtru v souboru AndroidManifest.xml.....	26
Obr. 9 Životní cyklus služby.....	27
Obr. 10 Hierarchie UI.....	32
Obr. 11 První spuštění aplikace.....	36
Obr. 12 Přidání zdroje a hlášení v případě chybného zadání.....	37
Obr. 13 Upozornění na nové články v základním a rozšířeném tvaru.....	38
Obr. 14 Dialogové okno progressBar.....	39
Obr. 15 Upozornění typu toast.....	39
Obr. 16 Hlavní okno aplikace spuštěné na telefonu se 4,5 palcovou obrazovkou.....	40
Obr. 17 Hlavní okno aplikace spuštěné na tabletu s 10 palcovou obrazovkou.....	40
Obr. 18 Navigační panel.....	41
Obr. 19 Seznam vyfiltrovaného zdroje.....	41
Obr. 20 Kontextové menu smazat.....	42
Obr. 21 Detail položky.....	43
Obr. 22 Výběr prohlížeče.....	43
Obr. 23 Obrazovka nastavení.....	44
Obr. 24 Rozevřený seznam.....	44

## SEZNAM PŘÍLOH

PŘÍLOHA P I: CD-ROM s aplikací a zdrojovými kódy

