

**Univerzita Pardubice**

**Fakulta ekonomicko-správní**

**Návrh aplikace pro mobilní OS Android ve vztahu k veřejné správě**

**Jan Zamazal**

**Bakalářská práce  
2013**

PROSTOR PRO ZADÁVACÍ LIST

## **PROHLÁŠENÍ**

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 30. 6. 2013

Jan Zamazal

## **PODĚKOVÁNÍ:**

Tímto bych rád poděkoval svému vedoucímu práce Ing. Janu Panušovi, Ph.D. za jeho odbornou pomoc, cenné rady a poskytnuté materiály, které mi pomohly při zpracování bakalářské práce. Dále bych rád poděkoval členům mé rodiny za jejich podporu při studiích.

## **ANOTACE**

*Tato práce bude sloužit studentům, kteří se zajímají o operační systém Android, a pro porozumění na jakých principech lze pro něj vytvářet aplikace. Studenti se zde seznámí s obecnými charakteristikami systému, přístupy a zásadami využitelnými při psaní zdrojového kódu aplikací a nakonec s názornou ukázkou konkrétní aplikace, využitelnou ve veřejné správě.*

## **KLÍČOVÁ SLOVA**

*Operační systém, Android, aplikace, Linux, smartphone, programování*

## **TITLE**

*Designing an application for OS Android related to public administration*

## **ANNOTATION**

*This work will serve students who are interested in the operating system Android, and understanding of the principles upon which it is possible to create applications. Students will learn about the general characteristics of the system, approaches and principles usable in writing source code and finally the demonstration of specific application, usable in public administration.*

## **KEYWORDS**

*Operation system, Android, application, Linux, Smartphone, programming*

# OBSAH

ÚVOD .....	9
<b>1 POPIS OPERAČNÍHO SYSTÉMU ANDROID .....</b>	<b>10</b>
1.1 HISTORIE .....	10
1.2 OPEN HANDSET ALIANCE .....	11
1.3 VERZE OS ANDROID .....	11
1.3.1 <i>Android 1.0 Angel Cake</i> .....	12
1.3.2 <i>Android 1.1 Battenberg</i> .....	12
1.3.3 <i>Android 1.5 Cupcake</i> .....	12
1.3.4 <i>Android 1.6 Donut</i> .....	12
1.3.5 <i>Android 2.0/2.1 Eclair</i> .....	12
1.3.6 <i>Android 2.2 Froyo</i> .....	13
1.3.7 <i>Android 2.3 Gingerbread</i> .....	13
1.3.8 <i>Android 3.0/3.1/3.2 Honeycomb</i> .....	13
1.3.9 <i>Android 4.0 Ice Cream Sandwich</i> .....	13
1.3.10 <i>Android 4.1/4.2 Jelly Bean</i> .....	14
1.3.11 <i>Podíl verzí na trhu</i> .....	14
1.1 LOGO .....	15
1.2 LICENCE.....	16
1.3 INTERFACE.....	16
1.4 ARCHITEKTURA .....	17
1.4.1 <i>Linux kernel</i> .....	18
1.4.2 <i>Libraries (knihovny)</i> .....	19
1.4.3 <i>Android Runtime</i> .....	19
1.4.4 <i>Dalvik Virtual Machine</i> .....	19
1.4.5 <i>Aplication Framework</i> .....	19
1.4.6 <i>Aplikace</i> .....	20
1.5 GOOGLEPLAY .....	20
<b>2 VÝVOJ APLIKACÍ.....</b>	<b>21</b>
2.1 VÝVOJOVÉ PROSTŘEDÍ .....	21
2.1.1 <i>Emulátor</i> .....	21
2.1.2 <i>Odlíšnost Android SDK</i> .....	21
2.2 PROCESY A VLÁKNA.....	22
2.2.1 <i>Vlákno</i> .....	24
2.2.2 <i>Pracovní vlákno</i> .....	24
2.2.3 <i>AsyncTask</i> .....	24
2.3 KOMPONENTY APLIKACE .....	25
2.3.1 <i>Aktivity</i> .....	25
2.3.2 <i>Dodavatelé obsahu (Content Providers)</i> .....	28
2.3.3 <i>Záměry (Intents)</i> .....	28
2.3.4 <i>Služby (Services)</i> .....	28
2.3.5 <i>Broadcast receivers</i> .....	28
2.3.6 <i>Android Manifest</i> .....	29
2.4 STRUKTURA PROJEKTU.....	31
2.4.1 <i>Zdroje</i> .....	31
2.4.2 <i>Přístup ke zdrojům</i> .....	32
2.4.3 <i>Přizpůsobení zdrojů</i> .....	32
2.5 UŽIVATELSKÉ ROZHRAŇÍ .....	32
2.5.1 <i>Rozložení (Layout)</i> .....	34
2.5.2 <i>Ovládací prvky</i> .....	35
2.5.3 <i>Události</i> .....	35
2.5.4 <i>Menu</i> .....	36
2.5.5 <i>Action Bar</i> .....	37
2.5.6 <i>Dialog</i> .....	37
2.5.7 <i>Notification</i> .....	37
2.5.8 <i>Toasts</i> .....	38
2.5.9 <i>Styly a témata</i> .....	38
2.6 UKLÁDÁNÍ DAT .....	39

2.6.1	<i>SharedPreferences</i> .....	39
2.6.2	<i>Soubory v paměti</i> .....	40
2.6.3	<i>Interní paměť</i> .....	40
2.6.4	<i>Databáze</i> .....	41
2.7	MÉDIA .....	41
2.7.1	<i>Podporované formáty</i> .....	41
2.7.2	<i>MediaPlayer</i> .....	42
2.8	POLOHA A MAPY .....	42
2.9	SENZORY .....	42
<b>3</b>	<b>PRAKTICKÝ NÁVRH A VÝVOJ APLIKACE</b> .....	<b>44</b>
3.1	POPIS APLIKACE „NOVA PRACE“ .....	44
3.2	FUNKCIONALITA APLIKACE .....	44
3.3	INTERFACE APLIKACE .....	45
3.4	ANDROIDMANIFEST.XML PROJEKTU .....	48
3.5	SOUČÁSTI PROJEKTU .....	49
3.5.1	<i>ActionBarSherlock</i> .....	49
3.5.2	<i>Databáze</i> .....	49
3.6	ZHODNOCENÍ PROJEKTU .....	50
	<b>ZÁVĚR</b> .....	<b>51</b>
	<b>POUŽITÁ LITERATURA</b> .....	<b>52</b>
	<b>SEZNAM PŘÍLOH</b> .....	<b>54</b>

## SEZNAM ILUSTRACÍ

Obrázek 1: Zastoupení jednotlivých verzí OS Android .....	15
Obrázek 2: Logo OS Android.....	16
Obrázek 3: Architektura OS Android.....	18
Obrázek 4: Android Market x GooglePlay logo.....	20
Obrázek 5: Životní cyklus aktivity .....	27
Obrázek 6: Hierarchická struktura souboru AndroidManifest.xml .....	30
Obrázek 7 - Hierarchická struktura grafického UI .....	33
Obrázek 8: Use-case diagram aplikace NovaPrace     Zdroj: Vlastní zpracování .....	44
Obrázek 10: Aktivita JobsActivity     Zdroj: Vlastní zpracování .....	45
Obrázek 11: Aktivita DetailActivity     Zdroj: Vlastní zpracování.....	46
Obrázek 12: Zobrazení menu na verzi Android 2.1 Eclair     Zdroj: Vlastní zpracování ....	46
Obrázek 13: Dialog "O aplikaci"     Zdroj: Vlastní zpracování .....	47
Obrázek 14: Aktivita SettingsActivity     Zdroj: Vlastní zpracování.....	47



## ÚVOD

Mobilní telefon je zařízení, které v posledních letech zažilo obrovský rozmach napříč moderní společností a jeho nejsilnější stránka, tedy možnost okamžité komunikace s kýmkoliv v dosahu vysílačů signálu, neodmyslitelnou součástí našeho každodenního žití. Nejen, že díky němu máme vždy v dosahu naše blízké, ať už rodinu nebo přátelé, umožňuje také podnikatelům efektivněji řídit své firmy a v neposlední řadě urychluje komunikaci i ve veřejné správě.

Trh s mobilními telefony postupem času vzrostl a výrobci, aby si udrželi či vylepšili postavení na trhu, začali integrovat do telefonů nové funkce. První monochromatické displeje dnes již nahradily širokouhlé dotykové displeje s barevnou hloubkou až 24 bitů, integrované objektivy s rozlišením až 41 Mpix nahradily klasické fotoaparáty. Přehrávání hudby a videa v digitálních formátech, možnost připojení k internetu, GPS modul, USB, Bluetooth a Wi-Fi konektivita, senzorické snímače a mnoho dalších moderních technologií dali za vznik nynější podobě mobilního telefonu označovaného jako tzv. smartphone (chytrý telefon), který svojí architekturou připomíná spíše osobní počítač.

Jako každý osobní počítač i smartphone potřebuje ke správě svého hardware operační systém, který poskytne uživateli rozhraní pro komunikaci se všemi jeho komponenty. V současnosti nejrozšířenějším systémem je Android, vyvíjený sdružením Open Handset Alliance, který podle analytické společnosti IDC (International Data Corporation) dosáhl v roce 2012 tří čtvrtinový podíl na trhu s prodanými smartphony a zanechal tak daleko za sebou iOS výhradně dodávaný s telefony iPhone od společnosti Apple, BlackBerry od společnosti RIM, který ovšem v Evropě nebyl moc rozšířen, Symbian od společnosti Nokia a v neposlední řadě také Microsoft s jeho operačním systémem Windows Mobile.

Cílem této práce je seznámit čtenáře s operačním systémem Android, poukázat na jeho specifika a pokusit se objasnit proč je právě tento systém dominantním na trhu. Dále pak přiblížit problematiku programování aplikací, ukázat nástroje a možnosti dostupné vývojářům. V poslední části této práce se pokusím vyvinout jednoduchou aplikaci, která, jak název práce napovídá, najde využití právě ve veřejné správě.

# 1 POPIS OPERAČNÍHO SYSTÉMU ANDROID

## 1.1 Historie<sup>1 2</sup>

Počátek operačního systému Android se datuje k roku 2003. V říjnu tohoto roku založili Andy Rubin, Rich Miner, Nick Sears a Chris White ve městě Palo Alto v Kalifornii společnost Android Inc., jejímž záměrem bylo vyvíjet, jak řekl sám Andy Rubin: „*smarter mobile devices that are more aware of its owner's location and preferences*“ (chytřejší mobilní přístroje, které si jsou více vědomy pozice majitele a jeho preferencí). Jejich prvním projektem bylo vytvořit pokročilý operační systém pro digitální fotoaparáty, ale trh s nimi nebyl dostatečně velký a proto přeorientovali svoji snahu do operačního systému pro mobilní telefony, který by konkuroval Symbianu a Windows Mobile.

Dne 17. srpna roku 2005 byla společnost odkoupena IT gigantem Google Inc., který z ní vytvořil svojí dceřinou společností se zachováním všech jejich klíčových zaměstnanců včetně Andy Rubina, kterého dosadil do čela společnosti. Od té doby nebylo veřejně známo, na čem se ve společnosti pracuje, ale mnozí usoudili, že Google tímto krokem plánuje vstup na trh mobilních telefonů. Dalším milníkem byl 5. listopad roku 2007, kdy došlo ke zformování uskupení pod názvem Open Handset Alliance v čele s Google, jehož smyslem byl vývoj otevřeného standardu pro mobilní zařízení. Ještě v ten samý den byl veřejnosti představen první Android 1.0, založený na linuxovém jádře 2.6, který se dočkal pouze vlažné reakce. O rok později, 22. října 2008, byl na trh uveden první smartphone s operačním systémem Android a to HTC Dream, znám také pod názvem T-Mobile G1. Na český trh se tento telefon dostal až počátkem roku 2009.

Nokia a Microsoft, doposud neohrožení, nového hráče na trhu ze začátku velmi podceňovali, ale myšlenka open-source operačního systému zažehla jiskru a Android se brzy začal probíjovat až na vrchol. Analytická společnost Canalys připisovala Androidu ve druhém kvartálu roku 2009 2,8% podíl na světovém trhu, o rok a půl později tedy ve čtvrtém kvartálu roku 2010 už podíl činil 33%, čímž se stal Android nejprodávanějším operačním systémem pro smartphony. Podle společnosti Gartner ve třetím kvartálu roku 2011 měl Android více jak polovinu trhu (52,5%) a o rok na to,

---

<sup>1</sup> BEZROUKOV, Nikolai. *Android History*. Softpanorama [online]. 2013 [cit. 2013-08-07]. Dostupné z: [http://www.softpanorama.org/Commercial\\_linuxes/Android/history.shtml/](http://www.softpanorama.org/Commercial_linuxes/Android/history.shtml/)

<sup>2</sup> MARVAN, Filip. *Mobilní operační systém Android*. DIIT.CZ [online]. 2011 [cit. 2013-08-07]. Dostupné z: <http://diit.cz/clanek/mobilni-operacni-system-android>

výzkum, prováděný společností IDC, přikládá Androidu tržní podíl ve výši 75%, čímž naprosto ovládnul trh.<sup>3</sup>

## 1.2 Open Handset Alliance<sup>4</sup>

OHA je konsorcium sdílejících společnou vizi o konkurenceschopném operačním systému pro mobilní zařízení a zaštiťující vývoj operačního systému Android. Bylo zaleženo 5. listopadu 2007 34 společnostmi. V současnosti má 84 členských společností, které lze podle charakteru rozdělit do 5 kategorií.

- **Mobilní operátoři** (například v ČR známi Vodafone, T-Mobile, Telefónica, ...)
- **Výrobci mobilních zařízení** (ASUSTeK Computer Inc., Samsung Electronics, Sony Ericsson, ...)
- **Výrobci polovodičů** (Broadcom Corporation, Intel Corporation, NVIDIA Corporation, ...)
- **Softwarové společnosti** (Google Inc., eBay Inc., NXP Software, ...)
- **Obchodní společnosti** (Accenture, L&T Infotech, Wipro Technologies, ...)

## 1.3 Verze OS Android<sup>5 6</sup>

Od první verze vydané v roce 2008 vyšla řada dalších aktualizací a nových verzí systému. Následující text stručně charakterizuje pokrok ve funkcích integrovaných do systému. Zajímavostí jsou kódová označení jednotlivých verzí systému začínající vždy dalším písmenem abecedy v řadě a nesoucí název cukrovinkových pochutin.

---

<sup>3</sup> Android ovládl trh. Jeho podíl vystoupal na 75 %, poslední z velkých firem je Microsoft s Windows. DSL.CZ [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://www.dsl.cz/clanek/2754-android-ovladl-trh-jeho-podil-vystoupal-na-75-posledni-z-velkych-firem-je-microsoft-s-windows>

<sup>4</sup> Open Handset Alliance. Open Handset Alliance [online]. 2013 [cit. 2013-08-07]. Dostupné z: [http://www.openhandsetalliance.com/oha\\_members.html](http://www.openhandsetalliance.com/oha_members.html)

<sup>5</sup> MARVAN, Filip. Mobilní operační systém Android. DIIT.CZ [online]. 2011 [cit. 2013-08-07]. Dostupné z: <http://diit.cz/clanek/vyvoj-androidu>

<sup>6</sup> COMPLETE HISTORY OF ANDROID. ANSAR MUHAMMED BASHIR [online]. 2011 [cit. 2013-08-07]. Dostupné z: <http://eceansar.blogspot.cz/2011/07/complete-history-of-android-successthe.html>

### **1.3.1 Android 1.0 Angel Cake**

- Datum vydání: září 2008.

Prvním z rodiny OS Android přinesl uživateli do telefonu integraci Google cloudových služeb, možnost surfování na internetu s plnou podporou HTML a XHTML, přístup do Android Marketu ke stahování nových aplikací, multitasking a podporu IM (Instant Messaging), Wi-fi a Bluetooth.

### **1.3.2 Android 1.1 Battenberg**

- Datum vydání: únor 2009.

První vylepšení systému na sebe nenechalo dlouho čekat, ale nepřineslo žádné převratné změny spíše jen pár oprav a vylepšení výkonu.

### **1.3.3 Android 1.5 Cupcake**

- Datum vydání: duben 2009.
- Linux kernel: 2.6.27.

Verze 1.5 už přinesla nějaká zásadní vylepšení, jako bylo grafické upravení vzhledu většiny základních grafických aplikací, nová softwarová klávesnice, ale také podpora formátu médií MPEG-4 a 3GP, možnost uploadování na servery YouTube a Picaso a vylepšení Bluetooth.

### **1.3.4 Android 1.6 Donut**

- Datum vydání: září 2009.
- Linux kernel: 2.6.29.

Nově přidaná podpora technologií CDMA/EV-DO, 802.1x a VPN, dále přidána aplikace Quick Search Box na vyhledávání přímo z domovské plochy. Také vylepšuje aplikaci Android Market a zůstřuje možnost hlasového vyhledávání.

### **1.3.5 Android 2.0/2.1 Eclair**

- Datum vydání: říjen 2009.
- Linux kernel: 2.6.29.

Poslední vydaná verze v roce 2009 přinesla podporu pro Microsoft Exchange, Bluetooth 2.1, nový internetový prohlížeč s podporou pro HTML5, optimalizaci na zrychlení systému,

pár změn v uživatelském prostředí včetně animace pozadí plochy a podporu světelné diody při focení v šeru.

### **1.3.6 Android 2.2 Froyo**

- Datum vydání: květen 2010.
- Linux kernel: 2.6.32.

Verze Froyo obsahuje několik dalších očekávaných vylepšení. Došlo k dalšímu zrychlení systému díky kompilátoru JIT (Just-in-time), vylepšení správy paměti RAM, možnost instalace aplikací na paměťovou kartu a podpora OpenGL ES 2.0. Dále přináší možnost vytvořit z telefonu Wifi hotspot nebo sdílet připojení k internetu s PC přes USB. Přibyla také možnost doinstalovat do prohlížeče Adobe Flash 10.1.

### **1.3.7 Android 2.3 Gingerbread**

- Datum vydání: prosinec 2010.
- Linux kernel: 2.6.35.

Nejzajímavější změnou v této verzi je podpora NFC (Near Field Communication) technologie, které umožňuje například mobilní platby. Další novinkou je podpora formátu WebM pro HTML5 video, vylepšená QWERTY klávesnice a přechod ze souborového systému YAFFS na EXT4. Gingerbread je v současnosti nejrozšířenější verzí.

### **1.3.8 Android 3.0/3.1/3.2 Honeycomb**

- Datum vydání: únor 2011.
- Linux kernel: 2.6.36.

Honeycomb je první verzí optimalizovanou pro tablety. Nalezneme zde kompletně předělané uživatelské rozhraní, upravený multitasking, přístup ke Google eBooks a podporu videohovorů přes Google Talk.

### **1.3.9 Android 4.0 Ice Cream Sandwich<sup>7</sup>**

- Datum vydání: říjen 2011.
- Linux kernel: 3.0.1.

---

<sup>7</sup> JEŽEK, David. Google uvedl Android 4.0. DIIT.CZ [online]. 2011 [cit. 2013-08-07]. Dostupné z: [http://diit.cz/clanek/google-uvvedl-android-40#utm\\_source=readmore&utm\\_medium=textlink&utm\\_campaign=readmore](http://diit.cz/clanek/google-uvvedl-android-40#utm_source=readmore&utm_medium=textlink&utm_campaign=readmore)

Verze Ice Cream Sandwich odbourává potřebu hardware tlačítek, vylepšuje uživatelské rozhraní ve snaze zjednodušit uživateli práci s mnoha prvky systému. Obsahuje pokročilou technologii NFC, nazývanou Android Beam, kterou lze jednoduše sdílet data pouhým přiložením telefonů k sobě. Systém je optimalizován pro displeje s rozlišením 1280x720. Dále je vylepšena technologie rozpoznávání hlasu při psaní zpráv a přidáno rozpoznávání obličeje při odemykání systému.

### **1.3.10 Android 4.1/4.2 Jelly Bean<sup>8</sup>**

- Datum vydání: červenec 2012.
- Linux kernel: 3.4.

V systému jsou v této verzi předělané uživatelské účty, nyní lze přepínat mezi uživateli a každý má uložené svoje aplikace a nastavení. Vylepšení se dočkala aplikace fotoaparátu, která implementovala technologii PhotoSphere 360° fotografie.

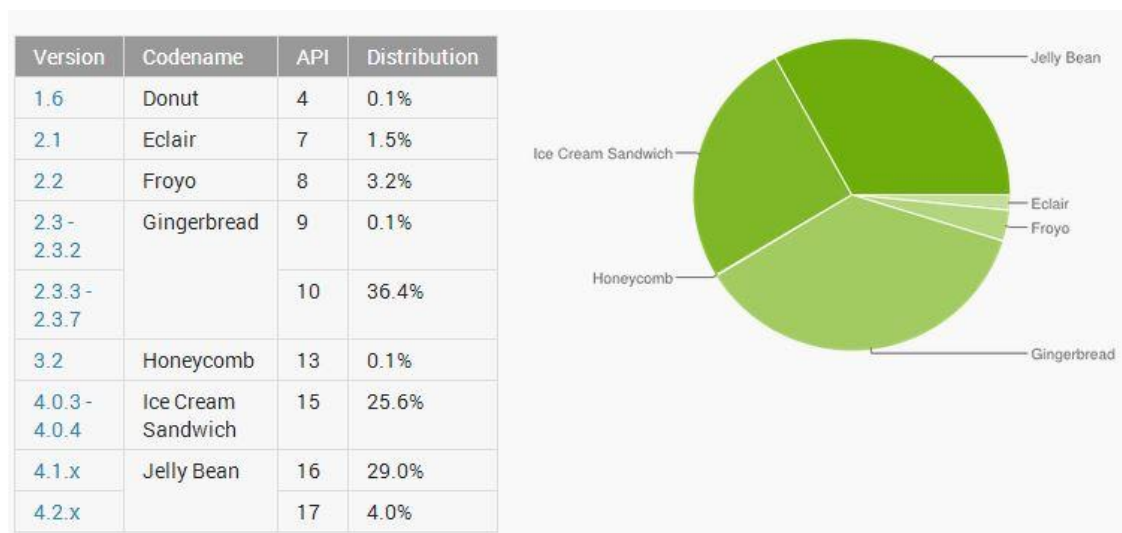
### **1.3.11 Podíl verzí na trhu<sup>9</sup>**

Následující Obrázek 1 zobrazuje, jak velké zastoupení mají jednotlivé verze na trhu. Data byla naměřena během 14 denní doby končící 3. června 2013 podle návštěvnosti Google Play Store (dříve znám jako Android Market). Výsledky menší než 0,1% ve výsledcích nejsou zahrnuty.

---

<sup>8</sup> KŮŽEL, Filip a Radek JANATA. Vyzkoušeli jsme Android 4.2 Jelly Bean: novinky, dojmy. *Mobilmania.cz* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://www.mobilmania.cz/clanky/vyzkouseli-jsme-android-42-jelly-bean-novinky-dojmy/sc-3-a-1322229/default.aspx>

<sup>9</sup> Dashboards. *Developer.android.com* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/about/dashboards/index.html>



**Obrázek 1: Zastoupení jednotlivých verzí OS Android<sup>10</sup>**

Nejpoužívanější Android systémy současnosti jsou tedy verze 2.3.3 – 2.3.7 označované Gingerbread. Významné procento mají však i starší verze, což musí vývojář brát v potaz při psaní své aplikace.

## 1.1 Logo<sup>11</sup>

Zelený robot, viz následující Obrázek 2, světlo světa spatřil v ten samý den, co byl představen veřejnosti první systém Android. Ze spousty designérských návrhů nakonec zvítězil ten nejjednodušší od Iriny Blokové, která pracovala pro marketingové oddělení přímo pro Google. Původně byl nazýván Bugdroid, dnes nese název robot Android. Jeho barva je v RGB označení #A4C639, u tisku se pak označuje jako PMS 376C. Google uvolnil robota Androida pod licencí Creative Commons Attribution, což znamená, že s ním lze libovolně nakládat pod podmínkou uvedení Google jako autora konceptu. Znění licence je dostupné na adrese <http://creativecommons.org/licenses/by/3.0/cz/>.

<sup>10</sup> Dashboards. Developer.android.com [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/about/dashboards/index.html>

<sup>11</sup> Jak vznikl maskot Androida?. Mobilmania.cz [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://www.mobilmania.cz/bleskovky/jak-vznikl-maskot-androida-puvodni-navrhy-byly-desive/sc-4-a-1322689>



**Obrázek 2: Logo OS Android<sup>12</sup>**

## **1.2 Licence<sup>13</sup>**

Na samotný operační systém Android se vztahují dvě licence. První z nich je GNU General Public Licence version 2, pod kterou je distribuováno linuxové jádro systému. Ve zkratce tato licence garantuje volně šiřitelný a modifikovatelný kód, ale veškeré změny tohoto kódu musejí být dále dostupné pod touto licencí. Plné znění této licence nalezneme na webu <http://www.gnu.org/licenses/gpl.html> a její český překlad na webu <http://www.gnugpl.cz/>.

Druhou licencí vztahující se na zbytek operačního systému je Apache License version 2.0, která také označuje volně šiřitelný a modifikovatelný kód, ale není podmínkou modifikace tohoto kódu distribuovat dále pod touto licencí. Znění licence je dostupné na webu <http://www.apache.org/licenses/LICENSE-2.0.html>.

Distribuce Androidu pod open-source licencemi beze sporu sehrála důležitou roli při nástupu na trh, výrobci telefonů tak mohli nasadit Android na svá zařízení a přitom si chránit soukromí svých technologií.

## **1.3 Interface<sup>14</sup>**

Uživatelské rozhraní OS Android je založeno na několika plochách po vzoru Linuxu, které si uživatel může přizpůsobit pomocí widgetů, dále se zde nachází stavový panel (stav baterie, síla signálu, odesílání dat, dostupná wifi, apod.), manažer upozornění

---

<sup>12</sup> *Jak vznikl maskot Androida?. Mobilmania.cz [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://www.mobilmania.cz/bleskovky/jak-vznikl-maskot-androida-puvodni-navrhy-byly-desive/sc-4-a-1322689>*

<sup>13</sup>

<sup>14</sup> *BEZROUKOV, Nikolai. Android. Softpanorama [online]. 2013 [cit. 2013-08-07]. Dostupné z: [http://www.softpanorama.org/Commercial\\_linuxes/Android/](http://www.softpanorama.org/Commercial_linuxes/Android/)*



na zobrazování důležitých informací spuštěných služeb (např. příchozí sms), nabídkové menu aplikací a poté samostatné aktivity aplikací. Ovládání je založeno na dotykové manipulaci s objekty. Většina modelů smartphonů s OS Android je vybavena třemi hardwarovými tlačítky, jedná se o „menu“, „domů“ a „zpět“. Některé modely mají ještě čtvrté tlačítko „vyhledávání“. Tlačítkem „domů“ se uživatel přesměruje přímo na domovskou plochu. Tlačítko „menu“ vyvolává nabídku možností v závislosti na situaci, ve které se systém nachází. Pokud je spuštěna aplikace, vyvolá nabídku možností té spuštěné aplikace. Tlačítko „zpět“ je určeno pro navigaci dozadu v zásobníku aktivit. V některých novějších typech mobilních zařízení jsou hardwarová tlačítka již odstraněna. Dalšími hardware ovládacími prvky mohou být gyroskop, měřič zrychlení nebo bezdotykový senzor, které jsou některými aplikacemi využívány.

#### **1.4 Architektura<sup>15 16 17 18</sup>**

Android je koncipován do několika vrstev, z nichž každá využívá služeb té bezprostředně pod ní a naopak poskytuje služby té bezprostředně nad ní. Na následujícím obrázku jsou zakresleny vrstvy od nejnižší, kterou je jádro systému, až po nejvyšší, kterou tvoří konečné aplikace, se kterými přijde běžný uživatel do styku. Vývojář aplikací pro Android by měl být ovšem obeznámen s celým schématem. Pro úplnost je dobré ještě zmínit Bootloader neboli zavaděč, který není přímou součástí systému, jedná se o samostatný program, který se stará o načtení jádra do operační paměti. Kromě této jeho hlavní funkce pak slouží například pro přístup k ROM, kde je operační systém fyzicky umístěn a umožňuje nahrání nové verze v případě poškození systému.

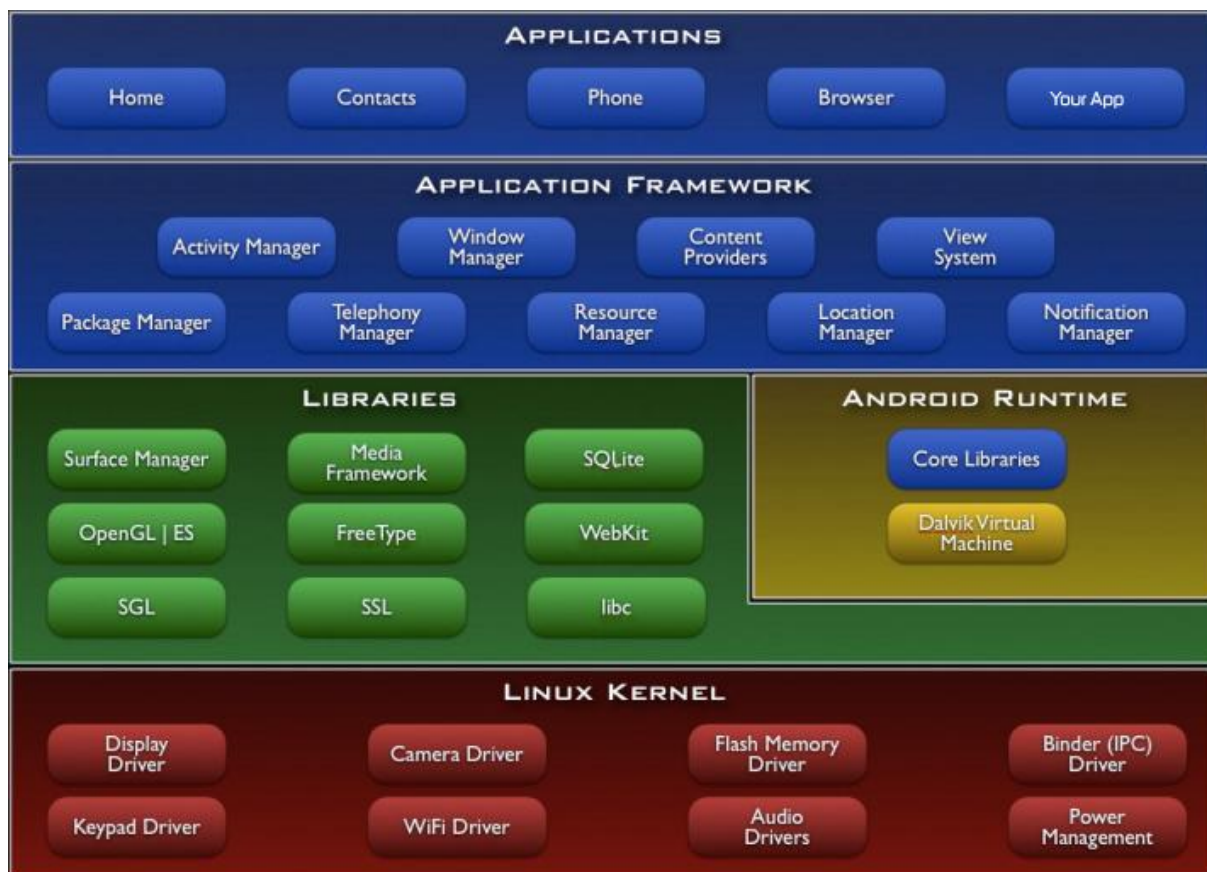
---

<sup>15</sup> ČERNÝ, Michal. *Operační systém Android. Metodický portál RVP.CZ [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://clanky.rvp.cz/clanek/o/g/15449/OPERACNI-SYSTEM-ANDROID.html/>*

<sup>16</sup> *Android Architecture – The Key Concepts of Android OS. Android-App-Market.com [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://www.android-app-market.com/android-architecture.html>*

<sup>17</sup> *Architektura. AndroidWiki [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://androidwiki.cz/Architektura>*

<sup>18</sup> *Professional Android 2 application development. Indianapolis: Wiley, 20101. ISBN 978-0-470-56552-0.*



Obrázek 3: Architektura OS Android<sup>19</sup>

### 1.4.1 Linux kernel

Základní vrstva operačního systému Android je linuxové jádro. Většina verzí stojí na jádru 2.6.x, od verze Androidu 4.0 Ice Cream Sandwich již využívají linuxového jádra 3.x. Jádro systému zodpovídá za zprostředkování fyzických hardwarových prvků další vrstvě. Skládá se z tzv. driverů neboli ovladačů jednotlivých hardware komponent. Obstarává správu procesů, operační paměti a souborů. Linuxové jádro lze snadno sestavit na jakémkoliv zařízení a díky open-source licenci lze podle potřeb jakkoliv modifikovat, což bylo pravděpodobně hlavním důvodem pro použití v operačním systému Android. Upravené jádro nepodporuje plný standard GNU knihoven, například neobsahuje podporu grafického rozhraní X Window System, používanou v desktopových linuxových distribucích.

<sup>19</sup> *Android Architecture – The Key Concepts of Android OS. Android-App-Market.com [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://www.android-app-market.com/android-architecture.html>*

## 1.4.2 Libraries (knihovny)

Další vrstvou navazující na jádro systému je vrstva nativních knihoven. Jsou psány v programovacím jazyce C/C++. Slouží ke zprostředkování základních funkčních vlastností systému. Nalezneme zde knihovny jako WebKit, která slouží pro vykreslování obsahu internetového prohlížeče, OpenGL knihovnu pro práci s 3D grafikou, SGL pro práci s 2D grafikou, Media Framework pro práci se soubory médií, SQLite pro práci s databází a mnoho dalších.

## 1.4.3 Android Runtime

Jedná se o vrstvu zajišťující běh aplikací. Skládá se z základních Java knihoven a virtuálního stroje nazvaný Dalvik Virtual Machine. Java knihovny v Android systému jsou svým obsahem velice podobné Java Standard Edition, ale nenajdeme zde například AWT a Swing knihovny pro grafické uživatelské rozhraní, které byly nahrazeny grafickými knihovnami View.

## 1.4.4 Dalvik Virtual Machine<sup>20</sup>

Dalvik Virtual Machine je srdcem systému Android, stará se totiž o běh aplikací. Funguje podobně jako Java Virtual Machine s tím rozdílem, že aplikace přeložené do Java bytecode překládá do svého vlastního „dexcode“ (soubory s příponou .dex), který poté spouští. Za vývojem DVM stojí Dan Bornstein z Google a důvodem jeho vzniku byla skutečnost že JVM není distribuován pod otevřenou licenci. Tím odpadá nutnost platit společnosti Oracle (vlastník práv k JVM) licenční poplatky, navíc je DVM optimalizován přímo pro prostředí s nízkou úrovní výpočetních zdrojů jako je právě smartphone. DVM plně podporuje multitasking, každá spuštěná aplikace vytváří svoji vlastní instanci.

## 1.4.5 Application Framework

Z pohledu vývojáře aplikací je tato vrstva nejdůležitější. Poskytuje přímý přístup ke službám, které mohou být využity při psaní konkrétní aplikace. Aplikační rámec je soubor knihoven psaných v jazyce Java, které dohromady tvoří API (**Application Programming Interface**). Najdeme zde bloky aplikačního rámce jako například View system, který poskytuje nástroje pro sestavení grafického uživatelského rozhraní aplikace (tlačítka, textová pole apod.), Activity manager, který se stará o životní cyklus aplikací, Content

---

<sup>20</sup> MURPHY, Mark. *Beginning Android*. [s.l.]: Apress, Inc., 2009. 361 s. ISBN 978-1-4302-2420-4

Providers, který uchovává data ke sdílení mezi aplikacemi, Telephony manager, který spravuje příchozí a odchozí hovory a další.

### 1.4.6 Aplikace

Samotné aplikace tvoří nevrchnější vrstvu celého systému. Přes aplikace uživatel využívá všech funkcí systému. Lze je rozdělit na aplikace instalované v základu systému jako jsou například adresář kontaktů, sms klient, internetový prohlížeč a aplikace třetích stran, které si uživatel instaluje dle svých potřeb a nároků.

## 1.5 GooglePlay<sup>21 22</sup>

GooglePlay je cloudová služba společnosti Google, která v březnu 2012 nahradila stávající službu Android Market, sloužící jako online obchod s aplikacemi, hudbou, knihami a filmy. Oproti Android Marketu nepřinesla služba GooglePlay žádné závratné změny, jedná se spíše o marketingový tah proti konkurenčním službám Applu. Cílem je poskytovat všechen multimediální obsah na každém zařízení s Androidem, na které se uživatel přihlásí. Není tedy nutností synchronizovat veškerá zařízení, aby měl uživatel dostupný veškerý zakoupený obsah na jakémkoliv zařízení. Jedinou podmínkou pro využívání služby je mít založen Google e-mail účet (Gmail). Služba GooglePlay však není v současnosti dostupná všude v plném rozsahu, v ČR je dostupná pouze část a to Google Play Store, ostatní multimediální obsah momentálně u nás není dostupný.



**Obrázek 4: Android Market x GooglePlay logo<sup>23</sup>**

<sup>21</sup> O službě Google Play. Support Google [online]. 2013 [cit. 2013-08-07]. Dostupné z: [https://support.google.com/googleplay/answer/2490014?hl=cs&p=play\\_faq&rd=1](https://support.google.com/googleplay/answer/2490014?hl=cs&p=play_faq&rd=1)

<sup>22</sup> Android Market končí, přichází Google Play: co je nového?. Mobilenet.cz [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://mobilenet.cz/clanky/android-market-konci-prichazi-integrovaný-google-play-co-je-noveho-8741>

## 2 VÝVOJ APLIKACÍ

### 2.1 Vývojové prostředí<sup>24</sup>

Základem pro vývoj aplikací pro systém Android je mít v počítači nainstalovány balíky Java SE SDK a Android SDK, protože jejich knihovny jsou základem, který v psaní aplikace vývojář využívá. Poté už jen stačí nainstalovat vhodné vývojové prostředí. Nejpoužívanějším a také oficiálně podporovaným prostředím pro vývoj aplikací na Android je Eclipse s pluginem ADT (Android Development Tools). Programátor si však může zvolit i jiná řešení, například dobře známé prostředí NetBeans, musí ovšem počítat s tím, že instalace všech potřebných nástrojů bude o něco obtížnější. ADT plugin pro prostředí Eclipse obsahuje veškeré nástroje a jeho instalace je maximálně snadná.

#### 2.1.1 Emulátor<sup>25</sup>

Jedním z důležitých nástrojů pluginu ADT je Android Emulator. Jedná se o virtuální mobilní zařízení, které běží na počítači. Emulátor dovoluje vývojářům testovat svoje aplikace bez nahrávání aplikace na fyzické zařízení. Při spouštění je nutné nastavit několik parametrů specifikujících virtuální systém spouštěný v emulátoru. Nainstalovaný systém obsahuje základní sadu aplikací.

#### 2.1.2 Odlišnost Android SDK<sup>26</sup>

Oproti standardnímu balíčku Java SDK, který nabízí řadu programátorských knihoven, na které jsou Java programátoři zvyklí, můžeme najít v Android SDK podporu pro funkce specifické pro chytré telefony.

---

<sup>23</sup> *Android Market končí, přichází Google Play: co je nového?. Mobilenet.cz [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://mobilenet.cz/clanky/android-market-konci-prichazi-integrovaný-google-play-co-je-noveho-8741/android-market-google-play.jpg>*

<sup>24</sup> *Instalace SDK. AndroidWiki [online]. 2012 [cit. 2013-08-07]. Dostupné z: [http://androidwiki.cz/Instalace\\_SDK](http://androidwiki.cz/Instalace_SDK)*

<sup>25</sup> *Android Emulator. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/tools/help/emulator.html>*

<sup>26</sup> *MURPHY, Mark L.. Android 2 : průvodce programováním mobilních aplikací. Brno: Computer Press, 2011. ISBN 978-80-251-3194-7*

- **Telefonní služby:** Jelikož Android je především systém pro telefony, může vyvíjená aplikace integrovat funkce jako posílání a přijímání SMS zpráv a uskutečňování telefonních hovorů.
- **GPS:** Global positioning system má v sobě zabudovaný většina chytrých telefonů, informace o aktuální poloze může být využita v aplikacích navigací či bezpečnostních aplikacích například při odcizení přístroje.
- **Multimédia:** Skupina knihoven zajišťující možnosti záznamu, přehrávání a upravování audio a video nahrávek.
- **Sít':** Obecně je systém Android koncipován pro stále připojení k síti pomocí různých technologií. Je možné toto připojení využít v aplikaci ať už jen surovými Java knihovnamy nebo využitím celého WebKit jádra.

## 2.2 Procesy a vlákna<sup>27</sup>

Pokud uživatel spustí nějakou komponentu aplikace a zároveň tato aplikace není již spuštěna na pozadí, systém vytvoří nový linuxový proces v jednom vlákne. Toto vlákno se nazývá hlavním vlákem a všechny komponenty aplikace pak v tomto vlákne běží. Nicméně vývojář aplikace může vytvořit pro jakoukoliv komponentu aplikace vlastní procesní vlákno. V souboru *AndroidManifest.xml* může být definováno ve kterém procesu má komponenta běžet.

Systém Android je navrhnut tak, aby z důvodu nedostatku operační paměti mohl některé procesy ukončit a uvolnit tak prostředky pro procesy s vyšší prioritou. Při rozhodování, který proces zabít, se tedy systém řídí podle důležitosti procesu pro uživatele. Například je mnohem vhodnější zabít proces, který je zastavený na pozadí, než proces, který je momentálně uživatelem využíván. Procesy teda mají hierarchické uspořádání z hlediska uživatelského významu. Při uvolňování paměti jsou tedy zabíjeny procesy s nejnižší prioritou, dokud uvolněná paměť nestačí pro další účely. Následující seznam prezentuje prioritní úrovně procesů od nejvíce důležitých po ty nejméně důležité.

### 1. Procesy v popředí

- Aktivita, se kterou právě uživatel manipuluje
- Služba, která je svázána s aktivitou, se kterou je právě manipulováno uživatelem

---

<sup>27</sup> *Processes and Threads. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/components/processes-and-threads.html>*

- Služba, která pracuje v popředí (byla zavolána metodou *startForeground()*)
- Služba, která je právě ve fázi svého cyklu *onCreate()*, *onStart()* nebo *onDestroy()*

## 2. Viditelné procesy

- Aktivita, která není přímo v popředí, ale je stále viditelná (nachází se ve stavu *onPause()*)
- Služba, která je svázána s viditelnou aktivitou

## 3. Proces služby

Proces provádějící nějakou službu, která byla spuštěna metodou *startService()* a zároveň nespadá do předchozího výčtu vyšších priorit služeb. Služby, ačkoliv nejsou pro uživatele viditelné, většinou vytvářejí v pozadí záslužnou činnost, například stahování nebo přehrávání hudby.

## 4. Proces na pozadí

Procesy, na kterých běží nějaká aktivita, která není přímo viditelná pro uživatele (aktivita je ve stavu *onStop()*). Zpravidla na pozadí běží takovýchto procesů hodně, systém má vytvořen LRU (last recently used) seznam těchto procesů, aby proces, který byl naposled viditelný uživateli, byl také posledním, který bude zabit pro uvolnění prostředků.

## 5. Prázdné procesy

Procesy, které nespravují žádné aktivní komponenty aplikace.

Android hodnotí priority procesů nejvyšší možnou cestou, například je-li v procesu běžící nějaká služba a viditelná aktivita, systém ohodnotí prioritu jako viditelnou aktivitu a ne jako proces služby.

Pokud na nějaký proces A (například služba), slouží jinému procesu B, nemůže být priorita procesu A nižší než priorita procesu B.

Z důvodu, že proces služby má vyšší prioritu než proces aktivity na pozadí, je silně doporučeno aktivitě, která spouští dlouhodobou operaci, aby pro tuto operaci vytvořila nový proces služby.

### 2.2.1 Vlákno

Když je spuštěna nová aplikace, systém vytvoří procesní vlákno, nazývané také hlavní vlákno. Toto vlákno je velice důležité, protože spravuje veškeré prvky uživatelského rozhraní a interakci uživatele s nimi. Někdy se také toto vlákno nazývá UI vlákno. Systém netvoří pro každou další komponentu aplikace nové vlákno, vše probíhá v tomto jednom hlavním. Pokud je v tom vláknu spuštěna dlouhodobější operace, může uživatelské rozhraní vykazovat špatný výkon při interakci s uživatelem nebo dokonce zamrznout úplně. Pokud je UI zablokováno déle jak 5 vteřin, objeví se ANR dialog (application not responding = aplikace nereaguje). Zpravidla se toto stává při dlouhých síťových operacích nebo složitých databázových dotazech. Existují dvě zásady jak se vyvarovat problémů s odezvou uživatelského rozhraní:

- Neblokovat UI vlákno
- Nezasahovat do uživatelského rozhraní z jiného než UI vlákna

### 2.2.2 Pracovní vlákno

Pokud tedy nějaká operace přímo nesouvisí s manipulací s prvky uživatelského rozhraní, je vhodné pro tuto činnost vytvořit vlastní pracovní vlákno. Vytvoření pracovního vlákna lze jednoduše vytvořením nového objektu třídy *Thread* a v kódu metody *run()* zapsat potřebné operace, které se provedou po zavolání metody *start()*. Pokud je třeba výsledek vlákna zobrazit v UI, je možné toto provést s pomocí hlavního UI vlákna. K tomuto účelu složí metody *Activity.runOnUiThread(Runnable)* nebo *View.post(Runnable)*.

V případě že složitost takovýchto operací na pozadí narůstá, je vhodné do pracovního vlákna zakomponovat objekt *Handler*, který zpracovává zprávy z hlavního vlákna. Pravděpodobně nejvhodnějším řešením je však využití třídy *AsyncTask*.

### 2.2.3 AsyncTask<sup>28</sup>

Instance této třídy je vhodná pro asynchronní práci s uživatelským rozhraním. Obsahuje metody, které rozdělují operace mezi hlavní UI vlákno a pracovní vlákno, které sama vytvoří. Její hlavní metody jsou:

---

<sup>28</sup> Vícevláknové zpracování. *AndroidWiki [online]. 2012 [cit. 2013-08-07]. Dostupné z: [http://androidwiki.cz/V%C3%ADcevl%C3%A1knov%C3%A9\\_zpracov%C3%A1n%C3%AD](http://androidwiki.cz/V%C3%ADcevl%C3%A1knov%C3%A9_zpracov%C3%A1n%C3%AD)*



- ***onPreExecute()*** – kód v této metodě se provádí v UI vlákne, může například zmrazit některé UI prvky
- ***doInBackground()*** – tato metoda vytvoří pracovní vlákno, ve kterém provede dlouhodobější operaci
- ***onPostExecute()*** – do této metody je doručen výsledek pracovního vlákna, který je poté může být implementován do uživatelského rozhraní pomocí UI vlákna
- ***onProgressUpdate()*** – metoda pro zobrazení například částečných mezivýsledků

Využívání *AsyncTask* má tedy nesporně velké výhody a to především v jednodušším a přehlednějším kódu a také ochraně UI proti ztrátě odezvy.

## 2.3 Komponenty aplikace<sup>29</sup>

Každá aplikace se skládá ze stavebních bloků, komponent. Každý z nich je samostatnou entitou s rozdílným životním cyklem, který definuje, jak je komponenta vytvořena a kdy zaniká.

### 2.3.1 Aktivity<sup>30 31</sup>

Jedná se o základní stavební prvky aplikace, lze je chápat jako analogii oken na stolním počítači. Aktivitě je při vytvoření přiřazen vzhled obsahující grafické ovládací prvky navržené pro přímou interakci s uživatelem (uživatelské rozhraní). Během svého životního cyklu se proces aktivity může nacházet ve třech stavech (resumed, paused, stopped), při každé změně stavu se volá příslušná metoda. Z důvodu omezených zdrojů může systém proces aktivity ve stavech paused a stopped zmrazit nebo úplně ukončit, aby uvolnil prostředky pro běh dalších procesů. Následující obrázek popisuje životní cyklus procesu aplikace.

---

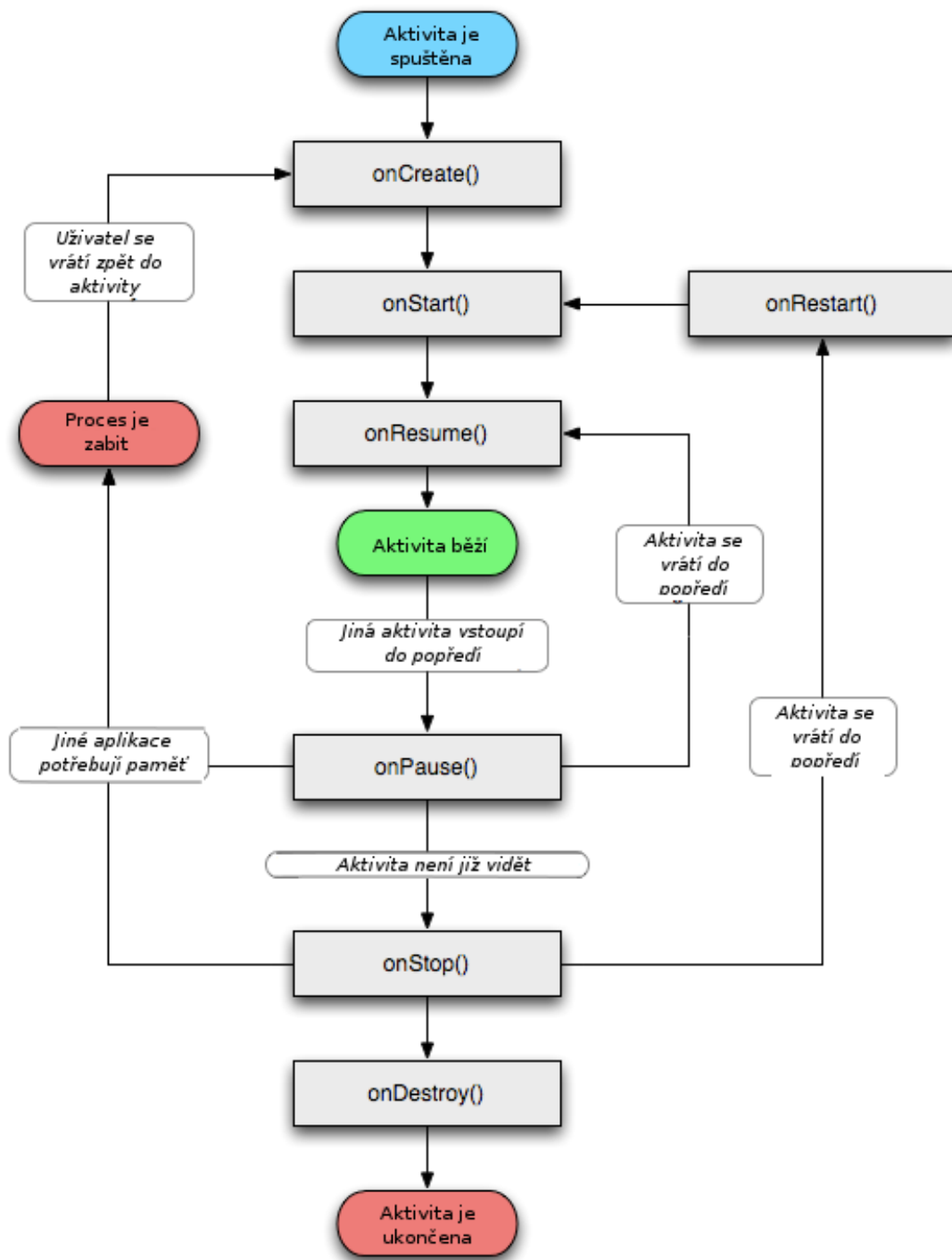
<sup>29</sup> MURPHY, Mark L.. *Android 2 : průvodce programováním mobilních aplikací*. Brno: Computer Press, 2011. ISBN 978-80-251-3194-7

<sup>30</sup> *Activities*. *Android Developers* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/components/activities.html>

<sup>31</sup> *Aktivity a intenty*. *AndroidWiki* [online]. 2012 [cit. 2013-08-07]. Dostupné z: [http://androidwiki.cz/Aktivity\\_a\\_intenty](http://androidwiki.cz/Aktivity_a_intenty)

### 2.3.1.1. Události procesu aktivity

- ***onCreate()*** – metoda volaná při vytvoření aktivity, doporučuje se zde inicializovat vzhled a nastavit posluchače objektů, této metodě může být předán parametr typu *Bundle* obsahující nějaká data (zpravidla uložená data z předchozí ukončené aktivity)
- ***onStart()*** – metoda volaná při zobrazení uživateli
- ***onRestart()*** – pokud se uživatel vrátí zpět do aktivity
- ***onPause()*** – pokud se uživatel přepne mimo tuto aktivitu a ta je tím pozastavena, doporučuje se zde ukládat data z aktivity, protože systém může v tomto stavu aktivitu ukončit
- ***onResume()*** – když se z pozastaveného stavu vrátí do běžícího stavu
- ***onStop()*** – pokud aktivita zmizí z přepínače úloh, pravděpodobně již nebude obnovena
- ***onDestroy()*** – je volána pokud je aktivita zrušena aplikací samotnou (například metodou *finish()*)



Obrázek 5: Životní cyklus aktivity<sup>32</sup>

<sup>32</sup> *Aktivity a intenty. AndroidWiki [online]. 2012 [cit. 2013-08-07]. Dostupné z: [http://androidwiki.cz/Aktivita\\_a\\_intenty](http://androidwiki.cz/Aktivita_a_intenty)*

### 2.3.2 Dodavatelé obsahu (Content Providers)<sup>33</sup>

Je to rozhraní pro abstrakci jakýchkoliv datových struktur uložených v zařízení, dostupných všem aplikacím v systému. Vývojový model směřuje vývojáře, aby zpřístupňovali data i jiným aplikacím. Toto je dosaženo právě díky dodavatelům obsahu.

### 2.3.3 Záměry (Intents)<sup>34</sup>

Jsou to systémové zprávy, na jejichž základě aktivita předává data a spouští jinou aktivitu nebo službu či broadcast receiver. Díky záměrům lze propojovat funkčnost více aplikací mezi sebou. Má podobu pasivní datové struktury, která nese data určená pro daný komponent.

### 2.3.4 Služby (Services)<sup>35</sup>

Služby jsou komponenty s dlouhodobou životností, pracující na pozadí systému. Služba může pracovat dvěma způsoby. První způsob je „spuštěná služba“, která dále svojí životností nezávisí na komponentu, který ji spustil. Obvykle provádí na pozadí činnost jako aktualizace, nahrávání dat na síť, apod., po dokončení své úlohy se sama ukončí. Druhý způsob je „svázaná služba“, jeden nebo více komponentů se naváže na službu, která potom poskytuje komponentům klient-server rozhraní. V tomto případě se služba ukončí ve stavu, kdy na ní nejsou navázány žádné komponenty.

### 2.3.5 Broadcast receivers<sup>36</sup>

Tato komponenta slouží na rychlou reakci na systémové události. Nedisponuje grafickým rozhraním, jejím účelem je pouze informovat ostatní komponenty. Například může sloužit k ohlášení nízkého stavu baterie, příchozí SMS zprávy, apod.

---

<sup>33</sup> Content providery. AndroidWiki [online]. 2012 [cit. 2013-08-07]. Dostupné z: [http://androidwiki.cz/Content\\_providery](http://androidwiki.cz/Content_providery)

<sup>34</sup> Intents and Intent Filters. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/components/intents-filters.html>

<sup>35</sup> Services. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/components/services.html>

<sup>36</sup> Broadcast receivers. AndroidWiki [online]. 2012 [cit. 2013-08-07]. Dostupné z: [http://androidwiki.cz/Broadcast\\_receivery](http://androidwiki.cz/Broadcast_receivery)

### 2.3.6 Android Manifest<sup>37</sup>

Každá aplikace musí povinně obsahovat soubor *AndroidManifest.xml* (přesně takto pojmenovaný). Manifest prezentuje důležité informace o aplikaci systému, před tím než je aplikace spuštěna. Nese unikátní název balíčku aplikace, popisuje veškeré komponenty aplikace, deklaruje práva aplikace pro přístup ke chráněným službám a definuje minimální potřebné API pro běh aplikace. Je psán ve značkovacím jazyce XML. Na obrázku je zobrazena celá struktura souboru manifestu.

---

<sup>37</sup> *The AndroidManifest.xml File. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/manifest/manifest-intro.html>*

```

<?xml version="1.0" encoding="utf-8"?>

<manifest>

    <uses-permission />
    <permission />
    <permission-tree />
    <permission-group />
    <instrumentation />
    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />
    <compatible-screens />
    <supports-gl-texture />

    <application>

        <activity>
            <intent-filter>
                <action />
                <category />
                <data />
            </intent-filter>
            <meta-data />
        </activity>

        <activity-alias>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </activity-alias>

        <service>
            <intent-filter> . . . </intent-filter>
            <meta-data/>
        </service>

        <receiver>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </receiver>

        <provider>
            <grant-uri-permission />
            <meta-data />
            <path-permission />
        </provider>

        <uses-library />

    </application>

</manifest>

```

**Obrázek 6: Hierarchická struktura souboru AndroidManifest.xml<sup>38</sup>**

<sup>38</sup> *The AndroidManifest.xml File. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/manifest/manifest-intro.html>*

## 2.4 Struktura projektu<sup>39</sup>

Při vytváření nového projektu vývojáře přehledně naviguje vývojové prostředí, kde je třeba vyplnit například název projektu, minimální požadovaná úroveň API a několik dalších kroků. Po dokončení úvodní sekvence kroků se vygeneruje struktura kořenového adresáře projektu specifická pro Android aplikace, která obsahuje následující:

- ***AndroidManifest.xml***: soubor popisující aplikaci a její komponenty (viz kapitola 2.6.6)
- ***Build.xml***: ant skript pro kompilaci aplikace a její instalaci
- ***Default.properties/local.properties***: soubory nastavení pro překladač Ant
- ***Assets/***: složka obsahující binární soubory, které chce vývojář přidat k aplikaci
- ***Bin/***: složka uchovávající zkompilovanou aplikaci
- ***Gen/***: složka vygenerovaných zdrojových kódů překladačnými nástroji
- ***Libs/***: složka, kde se nacházejí Java knihovny třetích stran, které aplikace využívá
- ***Src/***: složka, ve které se nachází zdrojový kód aplikace
- ***Res/***: složka uchovávající zdrojové prostředky grafického uživatelského rozhraní (pohledy aktivit, ikony, obrázky, zvukové stopy)
- ***Tests/***: složka, kde se nachází oddělený Android projekt pro testování

### 2.4.1 Zdroje<sup>40</sup>

Samotné zdroje lze rozlišovat na Assets a Resources, jsou to soubory připojené k projektu, které neobsahují java zdrojový kód. V případě Assets se jedná pouze o binární soubory ne často využívané. Ovšem v případě Resources se jedná o podstatnou část projektu, která zahrnuje následující podsložky:

- ***Res/drawable/***: obsahuje obrázky (png, jpeg, ...)
- ***Res/anim/***: předdefinované animace

---

<sup>39</sup> MURPHY, Mark L.. *Android 2 : průvodce programováním mobilních aplikací*. Brno: Computer Press, 2011. ISBN 978-80-251-3194-7

<sup>40</sup> Zdroje. *AndroidWiki* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://androidwiki.cz/Zdroje>

- **Res/color/**: barevná paleta nadefinovaných barev
- **Res/layout/**: takto složka obsahuje předdefinované návrhy uživatelského rozhraní psaného v XML
- **Res/menu/**: pro specifikace menu založené XML
- **Res/raw/**: složka pro ostatní typy souborů
- **Res/values/**: složka pro hodnoty proměnných řetězců, polí, stylů a dalších
- **Res/xml/**: pro ostatní XML návrhy

## 2.4.2 Přístup ke zdrojům

Při první kompilaci aplikace kompilátor vygeneruje soubor *R.java*, který obsahuje veškeré konstanty nacházející se v adresáři *res/*, při psaní zdrojového kódu se poté k těmto konstantám prostřednictvím tohoto souboru jednoduše přistupuje přes ID, které se skládá z typu zdroje a názvu zdroje. Například zápis ve zdrojovém kódu *R.string.hello* odkazuje na řetězec umístěný v */res/values/string.xml*. Druhý způsob přístupu ke zdrojům je používán v XML souborech a odkaz na ten samý řetězec by měl podobu *@string/hello*.

## 2.4.3 Přizpůsobení zdrojů<sup>41</sup>

Různá mobilní zařízení z pravidla mívají rozdílné parametry zobrazení, zejména pak rozlišení displeje a systémový jazyk. V adresáři zdrojů lze vytvořit složky, které budou obsahovat soubory pro alternativní zobrazení. V praxi pak taková složka je opatřena příponou charakteristickou pro dané zobrazení, například složka *values-cz* by obsahovala řetězce pro český jazyk. Dále se často využívá alternativní složka *layout-land* pro zobrazení při otočeném telefonu delší hranou dolů nebo složka *drawable-hdpi* pro telefony s vysokým rozlišením displeje.

## 2.5 Uživatelské rozhraní<sup>42</sup>

Veškeré elementy uživatelského rozhraní, jsou potomky třídy *View*. Objekty, které jsou potomkem této třídy, jednoduše vykreslují něco na plochu displeje, s čím uživatel poté

---

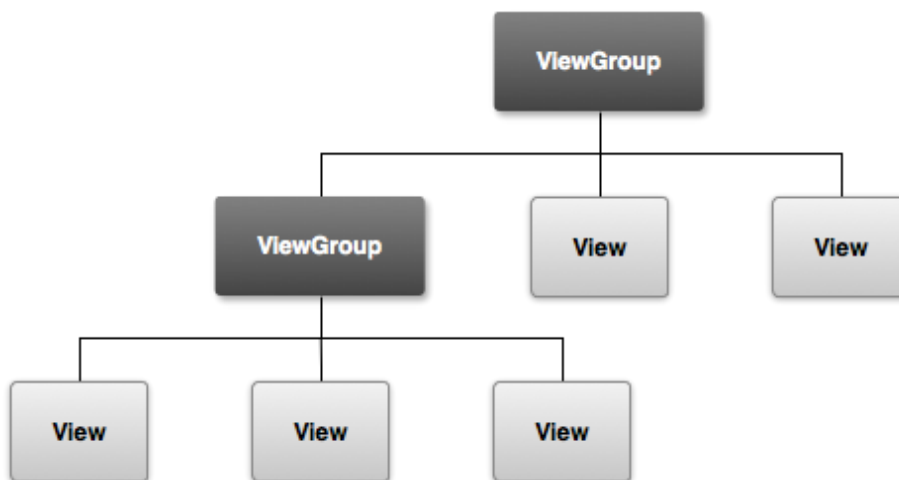
<sup>41</sup> Zdroje. *AndroidWiki* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://androidwiki.cz/Zdroje>

<sup>42</sup> *UI Overview. Android Developers* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/overview.html>



může manipulovat. Speciální podtřídou je třída *ViewGroup*, jejíž potomci mají za úkol především držet ostatní *View* objekty pohromadě a tvořit tak celkové uživatelské rozhraní.

Samotné uživatelské rozhraní je definované hierarchickou strukturou *ViewGroup* a *View* objektů, kde *ViewGroup* je neviditelný kontejner, který organizuje zobrazení a *View* představují jednotlivé prvky, které vykreslují uživateli vstupní ovládací prvky, grafická zobrazení a další. Tato struktura může být libovolně složitá, avšak jednoduchost znamená rychlejší běh aplikace.



Obrázek 7 - Hierarchická struktura grafického UI<sup>43</sup>

Při deklaraci rozhraní je možné iniciovat *View* objekty a vytvořit hierarchickou strukturu přímo ve zdrojovém kódu aplikace, ale nejjednodušší a také nejefektivnější cesta vede přes definování rozložení prvků do XML souboru. V XML jsou veškeré elementy pojmenovány po třídách objektů, které reprezentují, a každý prvek lze dle potřeby předdefinovat řadou vlastností. Při načtení rozložení během chodu aplikace se veškeré XML elementy převedou na objekty, se kterými lze dále v kódu provádět libovolné změny. Nejdůležitějším atributem jednotlivých prvků v XML návrhu je atribut *id*, přes který se přistupuje k danému prvku ve zdrojovém kódu aplikace. Dále jsou velice užitečné atributy *layout\_width*, *layout\_height* a *orientation*, které slouží k definování velikosti objektu a orientaci na šířku nebo na výšku. Samotná výška a šířka může být deklarována dvěma způsoby, relativní a absolutní cestou. Relativní cesta spočívá v nastavení šířky a výšky buď hodnotou *fill\_parent*, kde objekt se snaží být tak velký jako jeho rodičovský prvek anebo hodnotou *wrap\_content*, kde je objekt pouze tak velký jako jeho obsah.

<sup>43</sup> *UI Overview. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/overview.html>*

Absolutní cesta spočívá v přiřazení konkrétní hodnoty šířky a výšky objektu. Slouží k tomu několik druhů jednotek:

- *in* (inches, palce) – odpovídá 2,54cm
- *mm* (milimetr)
- *pt* (bod) – odpovídá 1/72 palce, přibližně 0,035 cm
- *px* (pixely) – jeden pixel odpovídá jednomu bodu na obrazovce, jejich počet je dán rozlišením displeje
- *dp* – nezávisí na počtu pixelů, počítá s hustotou 160dpi (bodů na palec)
- *sp* – podobně jako dp, ale mění se s velikostí písma

### 2.5.1 Rozložení (Layout)<sup>44</sup>

Typickým potomkem třídy *ViewGroup* je objekt rozložení, který definuje rozpoložení ostatních prvků obsažených uvnitř. V současnosti se nejvíce využívá rozložení *LinearLayout*, *RelativeLayout*, *ListView* a *GridView*. Dále existují starší a již nevyužívané *TableLayout* a *FrameLayout*.

#### 2.5.1.1. **LinearLayout**<sup>45</sup>

Lineární rozložení řadí obsažené objekty do řady za sebou podle nastaveného parametru *orientation* horizontálně nebo vertikálně. Nastavení pozic objektů lze ovlivnit přiřazením váhy k jednotlivým objektům (parametr *layout\_weight*), použitím okrajů, nebo centrováním (parametr *gravity*) vpravo, vlevo nebo doprostřed.

#### 2.5.1.2. **RelativeLayout**<sup>46</sup>

Jednotlivé objekty v relativním rozložení nejsou seřazeny za sebou, jejich pozice závisí na nastavení každého z nich. Mohou být přiřazeny k jakémukoliv dalšímu objektu uvnitř nebo umístěny vzhledem k celému rozložení.

---

<sup>44</sup> *Layouty*. *AndroidWiki* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://androidwiki.cz/Layouty>

<sup>45</sup> *Linear Layout*. *Android Developers* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/layout/linear.html>

<sup>46</sup> *Relative Layout*. *Android Developers* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/layout/relative.html>

### 2.5.1.3. **ListView**

*ListView* je rozložení zobrazující rolující seznam položek. Pro zobrazení se využívá třídy *Adapter*, která jednotlivé položky získává z pole nebo databáze. Každá položka pak má definováno určité vnitřní rozložení.

### 2.5.1.4. **GridView**

Zde se jedná o rozložení, které funguje stejně jako *ListView* s tím rozdílem, že výsledné zobrazení má podobu dvojdimenzionální tabulky.

## 2.5.2 **Ovládací prvky**<sup>47</sup>

Pro interakci s aplikací slouží uživateli řada prvků. Veškeré prvky mohou být definovány při tvorbě rozložení s využitím XML návrhu, nebo mohou být přidávány a měněny v průběhu chodu aplikace. Mezi standardně používané patří:

- ***TextView*** – obyčejné zobrazení textu
- ***ImageView*** – zobrazení obrázku
- ***EditText*** – pole pro vkládání textu uživatelem
- ***Button*** – tlačítko
- ***Checkbox*** – zaškrtačovací tlačítko
- ***RadioButton*** – pro výběr z několika nabízených možností
- ***ToggleButton*** – přepínač zapnuto/vypnuto
- ***Spinner*** – nabídka pro výběr z nabízených možností
- ***Picker*** – nabídka pro výběr času či data

## 2.5.3 **Události**<sup>48</sup>

Existuje více způsobů jak reagovat na události spuštěné uživatelem. Samotná třída *View* k tomu poskytuje prostředky a to ve formě veřejných metod se zpětnou reakcí. Například metoda *onTouchEvent()* je volána v případě dotyku na daný objekt. Toto řešení však

---

<sup>47</sup> *Input Controls. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/controls.html>*

<sup>48</sup> *Input Events. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/ui-events.html>*

zahrnuje rozšíření třídy a přepsání metody. Druhým a snadnějším způsobem je přidáním posluchače (*Listener*) s jednou metodou, která je spuštěná, pokud posluchač zareaguje na danou událost. Nejpoužívanějšími objekty třídy *Listener* jsou:

- ***OnClickListener*** – reaguje na kliknutí
- ***OnLongClickListener*** – reaguje na dlouhé kliknutí
- ***OnFocusChangeListener*** – reaguje na změnu zaměření
- ***OnKeyListener*** – reaguje na stisk klávesy definované parametrem *KeyEvent*
- ***OnTouchListener*** – reaguje na provedení dotyku, který je definován parametrem *MotionEvent*
- ***OnCreateContextMenuListener*** – reaguje na vyvolání kontextového menu

## 2.5.4 Menu<sup>49</sup>

Podobně jako na v aplikacích na stolních počítačích i aplikace psané na Android mohou obsahovat nabídku možností neboli menu. Existují dva typy menu, prvním je menu aplikace nazývané také menu voleb, druhým typem je kontextové menu, obsahující nabídku možností ke konkrétnímu prvku UI.

### 2.5.4.1. Menu voleb

Menu voleb je zobrazeno po stisku hardwarového tlačítka pro menu, případně softwarového ekvivalentu v novějších typech telefonů. Po stisku se zobrazí prvních šest voleb jako velká tlačítka, pokud se menu skládá z více možností, šesté tlačítko bude označené jako další a po jeho stisku se zobrazí zbylé volby. Ve zdrojovém kódu se menu implementuje v metodě *onCreateOptionsMenu()*, které se předá parametr s objektem třídy *Menu*.

### 2.5.4.2. Kontextové menu

Kontextové menu se vyvolává podržením prstu na konkrétním ovládacím prvku. Je nutné specifikovat, které objekty kontextové menu potřebují. V aktivitě, která tyto objekty obsahuje, se zavolá metoda *registerForContextMenu()* a předají se jí potřebné

---

<sup>49</sup> *Menus. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/menus.html>*

objekty. Poté stačí implementovat metodu *onCreateContextMenu()*, které se předá parametr s objektem třídy *ContextMenu*.

### 2.5.5 Action Bar<sup>50</sup>

*ActionBar* je implementace panelu do aplikací. Velice zpřehledňuje navigaci, zpravidla obsahuje prvky jako název aktivity, ve které se aplikace nachází a nabídku akcí, které lze dále vykonávat. Poprvé byl zaveden až ve verzi Android 3.0 (API úroveň 11), ale existují podpůrné knihovny, které jeho funkčnost přinášejí i do starších verzí, konkrétně jej lze využívat již od Android 2.1 (API úroveň 7). Přidání *ActionBaru* lze jednoduše provést zděděním z třídy *ActionBarActivity*, poté aktivita na panelu zobrazuje klasické menu voleb.

### 2.5.6 Dialog<sup>51</sup>

*Dialog* je objekt uživatelského rozhraní, který po spuštění vyvolá malé okno v popředí aktivity na které je spuštěn. Jeho účelem může být informovat uživatele o nějaké akci, nebo dotaz, zda pokračovat v činnosti či výběr nějakého parametru pro další zpracování. Systém Android nabízí několik druhů dialogových oken, všechny odvozené od třídy *Dialog*.

- ***AlertDialog*** – nejpoužívanější druh dialogu, zobrazuje titulek, může obsahovat až tři tlačítka a nějaký obsah, který může mít podobu od obyčejného textu až po list položek
- ***DatePickerDialog*** – dialog pro výběr data
- ***TimePickerDialog*** – dialog pro výběr času

Pokud prostředky připravených dialogů nejsou dostatečné, je možné vytvořit vlastní dialog rozšířením třídy *Dialog* a definováním vlastního layoutu.

### 2.5.7 Notification<sup>52</sup>

Jedná se o zprávu, která se zobrazuje mimo aplikační rozhraní v panelu oznámení, který je přístupný tahem horní lišty směrem dolů a je spravován systémem. Notifikace jsou určeny k informování uživatele o důležitých událostech či o probíhajících činnostech.

---

<sup>50</sup> *Action Bar. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/actionbar.html>*

<sup>51</sup> *Dialogs. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/dialogs.html>*

<sup>52</sup> *Notifications. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/notifiers/notifications.html>*

Typické notifikace jdou například oznámení o příchozí sms, zmeškaném hovoru či dostupných aktualizacích.

K vytvoření notifikace slouží objekt *NotificationCompat.Builder*, který po zadání parametrů a spuštění metody *build()* vrací objekt třídy *Notification*. Ten se předá objektu třídy *NotificationManager*, který je dostupný metodou *getSystemService(Context.NOTIFICATION\_SERVICE)*, a metodou *notify()* lze na něm upozornění spouštět.

V objektu třídy *Notification* je nutné definovat minimálně tři vlastnosti a to ikonu, titulek a detailní text upozornění. Pokud je třeba, aby po kliknutí na upozornění byla spuštěna nějaká aktivita, je třeba v objektu nastavit *PendingIntent* neboli odložený záměr, který příslušnou akci definuje.

### 2.5.8 Toasts<sup>53</sup>

*Toast* je jednoduchá zpětná reakce na nějakou operaci. Záměrem je informovat uživatele v podobě vyskakovacího oznámení minimálních rozměrů, toast zpravidla obsahuje pouze strohý text a objeví se pouze po krátké stanovenou dobu. Ukázkovým příkladem je například pokud uživatel má rozepsanou SMS a ukončí aplikaci, objeví se toast informující o uložení zprávy jako konceptu. Toast nijak neomezuje funkčnost a ovladatelnost právě spuštěné aktivity.

Vytvoření toastu v jeho nejjednodušší podobě lze zavoláním metody *makeText()* třídy *Toast*, kde se definují parametry, co má zobrazit a po jak dlouhou dobu, a nakonec zobrazením metodou *show()*. Standardně se toast objevuje ve spodní části obrazovky, jeho pozice se může nastavit metodou *setGravity()*. Vzhled toastu je možné přizpůsobit definováním layoutu.

### 2.5.9 Styly a témata<sup>54</sup>

Jedná se o kolekci vlastností, které definují vzhled a umístění pro prvky grafického UI. Definice každého UI prvku v XML je zdlouhavá na zápis, obzvláště jedná-li se o rozsáhlejší aplikaci. Obdobně jako se ve web-designu využívá CSS, tak také zde lze

---

<sup>53</sup> *Toasts*. *Android Developers [online]*. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/notifiers/toasts.html>

<sup>54</sup> *Styles and Themes*. *Android Developers [online]*. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/ui/themes.html>

nadefinovat jednotný styl pro objekty stejných tříd. Při psaní stylu je kořenovým elementem `<resources>`, ve kterém se nachází element `<style>` s nadefinovanými položkami atributů a jejich vlastnostmi v elementech `<item>`. Použití stylu se pak provádí odkazem `style="@style/název"`.

Pokud je styl definován pro celou aktivitu, nazývá se tématem.

## 2.6 Ukládání dat<sup>55</sup>

System Android nabízí řadu možností pro uložení dat vytvořených aplikací. Konkrétní řešení ukládání dat závisí na specifických potřebách. Je třeba zhodnotit, jak velký objem dat je třeba ukládat nebo zdali uložená data budou k dispozici i dalším aplikacím.

### 2.6.1 SharedPreferences<sup>56</sup>

Sdílené preference poskytují aplikaci úložiště pro párová data typu klíč-hodnota, což se hodí pro ukládání nastavení aplikace. Lze ukládat pouze primitivní typy dat, jako jsou *boolean*, *float*, *int*, *long* a *string*. Pro získání objektu sdílených preferencí slouží dvě metody:

- ***getPreferences()*** – slouží pro získání výchozích preferencí aplikace
- ***getSharedPreferences()*** – metoda vrací konkrétní preferenční soubor specifikovaný jménem, které se předá metodě jako argument

Pro čtení preferencí se volá metoda určená pro typ dané proměnné, kterou chceme získat. Tedy například pro získání řetězce zavoláme metodu *getString()*, pro získání celého čísla voláme *getInt()*. V metody vyžadují dva argumenty, klíč a návratovou hodnotu pokud klíč nebyl v preferencích nalezen.

Zápis do preferencí probíhá v několika krocích. Prvním krokem je zavolání metody *edit()* nad objektem preferencí, která vrací objekt třídy *SharedPreferences.Editor*. Poté z objektu této třídy voláme metody podle typu ukládaných proměnných, tedy *putString()* či *putInt()* apod, a specifikujeme v nich argumenty klíč a hodnota. Po provedení úprav v editoru potvrdíme metodou *commit()* a veškeré hodnoty se uloží do souboru preferencí.

---

<sup>55</sup> *Storage Options. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/data/data-storage.html>*

<sup>56</sup> *Preference. AndroidWiki [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://androidwiki.cz/Preference>*

## 2.6.2 Soubory v paměti<sup>57</sup>

Dalším způsobem je uložení dat do ručně vytvořených souborů v interní či externí paměti telefonu.

### 2.6.3 Interní paměť

Vytvoření a zápis do souboru probíhá v následujících krocích. Pomocí metody *openFileOutput()* s argumenty jméno souboru a mód přístupu lze získat objekt třídy *FileOutputStream*. Metodou *write()* se zapisují do souboru data a nakonec metodou *close()* se soubor uzavře.

Módy pro operaci se soubory:

- **MODE\_PRIVATE** – vytvoří soubor s názvem (pokud název souboru existuje, přepíše jej) a zpřístupní ho pouze aplikaci
- **MODE\_WORLD\_READABLE** – oproti **MODE\_PRIVATE** zpřístupní čtení souboru i ostatním aplikacím
- **MODE\_WORLD\_WRITABLE** – zpřístupní všem aplikacím navíc i zápis do souboru
- **MODE\_APPEND** – pokud soubor existuje, nepřepíše jej, pouze připojí ukládaná data na konec souboru

Pro čtení souboru se zavolá metoda *openFileInput()* se jménem souboru jako argumentem vracející objekt třídy *FileInputStream*, kde se pomocí metody *read()* čte ze souboru. Nakonec se soubor zavře opět metodou *close()*.

#### 2.6.3.1. Externí paměť

Každé Android zařízení poskytuje podporu pro sdílenou externí paměť, kterou lze využít pro ukládání souborů. Nejprve je vhodné ověřit si, zdali je paměť dostupná, a to metodou *Environment.getExternalStorageState()*, která navrácí stav primární externí paměti. Vytváření, čtení a zápis pak probíhají standardně jako do interní paměti pomocí java tříd k tomu určených. Cestu k souboru je však nutné specifikovat absolutní cestou, slouží k tomu například metoda *getExternalFileDir()*, která navrácí absolutní cestu k externí paměti (zpravidla */mnt/sdcard*).

---

<sup>57</sup> *Filesystem. AndroidWiki [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://androidwiki.cz/Filesystem>*



## 2.6.4 Databáze<sup>58</sup>

Pro potřeby ukládání složitějších datových struktur je systém Android vybaven podporou databáze SQLite. Doporučenou metodou pro vytvoření nové SQLite databáze je vytvoření samostatné podtřídy *SQLiteOpenHelper* a přepsání metody *onCreate()*, kde se implementuje kód na vytvoření jedné či více tabulek v databázi. Poté v aktivitě může být vytvořen objekt této třídy a metodou *getReadableDatabase()* nebo *getWritableDatabase()* je vrácen objekt třídy *SQLiteDatabase*, nad kterým jdou provádět klasické databázové operace jako *query()*, *insert()*, *delete()*, apod. Metoda *query()* pak vrací objekt třídy *Cursor*, který reprezentuje datovou strukturu získanou z databáze, tedy řádky a jednotlivé údaje.

## 2.7 Média<sup>59</sup>

Android disponuje podporou pro přehrávání celé řady multimediálních formátů. Při vytváření aplikace může vývojář zahrnout prvky přehrávání hudby a videí, využívat hardwarových komponent jako je kamera a mikrofon. Pokud se rozhodne připojit k aplikaci mediální soubory, využívá se k tomu složka */res/raw*. Lze také využít síťového připojení a média tak streamovat. K tomu účelu slouží třídy *MediaPlayer*, *AudioManager* a *SoundPool* pro přehrávání multimédií, *MediaRecorder* pro zachycení zvukové stopy a třída *Camera* pro ovládání integrované kamery.

### 2.7.1 Podporované formáty<sup>60</sup>

Vývojář Android aplikací by měl být obeznámen, jaké multimediální formáty může ve svém projektu využít. Následující výčet stručně charakterizuje podporovaná multimédia.

- **Síťové protokoly:** RTSP (RTP, SDP), HTTP/HTTPS progresivní stream, HTTP/HTTPS živý stream
- **Audio kodeky:** AAC LC, HE-AACv1 (AAC+), HE-AACv2 (enhanced AAC+), AAC ELD, AMR-NB, AMR-WB, FLAC, MP3, MIDI, Vorbis, PCM/WAVE
- **Video kodeky:** H.263, H264 AVC, MPEG-4 SP, VP8
- **Obrázky:** JPEG, GIF, PNG, BMP, WEBP

---

<sup>58</sup> *SQLite*. *AndroidWiki* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://androidwiki.cz/SQLite>

<sup>59</sup> *Media Playback*. *Android Developers* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/media/mediaplayer.html>

<sup>60</sup> *Supported Media Formats*. *Android Developers* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/appendix/media-formats.html>

## 2.7.2 MediaPlayer

*MediaPlayer* je hlavní třídou pro přehrávání lokálních mediálních souborů i streamování medií z URI. Pro spuštění souboru stačí vytvořit objekt třídy *MediaPlayer* a metodou *create()* iniciovat jeho obsah, kde se předá parametr v podobě odkazu na položku v adresáři */raw/res*. Poté už jen stačí zavolat metodu *start()* a soubor bude v aplikaci přehrán.

## 2.8 Poloha a mapy<sup>61</sup>

Jedna z předností mobilních zařízení s operačním systémem Android je, že téměř všechna podporují možnost zaměření polohy. Využívá se přitom hned několika technologií, zaměření může být provedeno pomocí GPS přijímače, triangulací signálu mobilní sítě nebo pomocí přístupového Wi-Fi bodu.

Zpřístupnění těchto služeb zajišťuje třída *LocationManager*, o její instanci se z aplikace žádá metodou *getSystemService(Context.LOCATION\_SERVICE)*. Díky tomuto poté může aplikace zaznamenávat současnou polohu a vytvářet záměry, které upozorňují na blízké body zájmu, pokud se zařízení dostane do určité vzdálenosti. Aby aplikace mohla využít *LocationManager* je nutné do manifestu přidat povolení *ACCESS\_FINE\_LOCATION*.

Od verze Android 2.2 je k dispozici Google Maps Android API, které je volně ke stažení na GooglePlay. Lze díky tomu naplno využívat v mobilním zařízení zobrazení mapy celého světa a využít tak naplno lokalizačních služeb. Klíčová třída je v tomto případě *MapView*, která zobrazuje mapu získanou z cloudové služby GoogleMaps. Takto získaná mapa navíc poskytuje veškeré ovládací prvky pro manipulaci.

## 2.9 Senzory<sup>62</sup>

Velká část zařízení Android disponují řadou senzorů, které mohou být využívány jako ovládací prvky aplikací nebo jen sloužit k měření hodnot a jejich následné interpretaci. Senzory se dají kategorizovat do tří skupin:

- **Pohybové senzory** – měří sílu zrychlení a rotační síly kolem tří os, skládají se z akcelerometru, gravitačního senzoru, gyroskopu a rotačních senzorů

---

<sup>61</sup> *Location and Maps. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/guide/topics/location/index.html>*

<sup>62</sup> *Sensors Overview. Android Developers [online]. 2013 [cit. 2013-08-07]. Dostupné z: [http://developer.android.com/guide/topics/sensors/sensors\\_overview.html](http://developer.android.com/guide/topics/sensors/sensors_overview.html)*

- **Senzory prostředí** – měří parametry okolí jako teplotu vzduchu, tlak a světelnou intenzitu, skládají se z barometru, fotometru a teploměru
- **Poziční senzory** – tyto senzory měří fyzickou polohu zařízení, zahrnují orientační senzor a magnetometr

Klíčovou třídou je zde *SensorManager*, o jeho instanci si aplikace musí požádat systém metodou *getSystemService(Context.SENSOR\_SERVICE)*. Tato třída disponuje metodami pro přístup k sensorům, registraci posluchačů sensorových událostí a získávání naměřených dat.

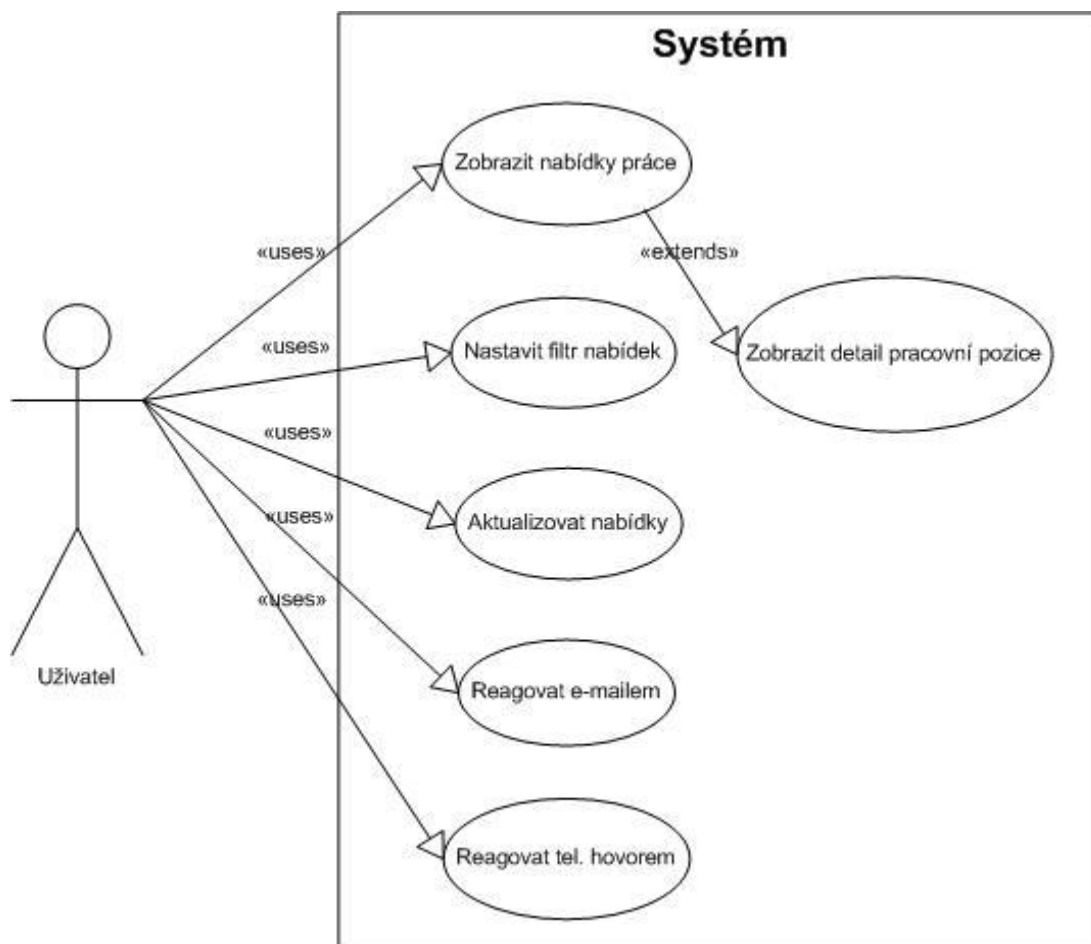
### 3 PRAKTICKÝ NÁVRH A VÝVOJ APLIKACE

#### 3.1 Popis aplikace „NovaPrace“

Jako praktický příklad ukázky vývoje aplikace jsem si připravil projekt s názvem NovaPrace. Jedná se o jednoduchou a graficky nenáročnou aplikaci, která přináší do mobilního zařízení možnost procházení nabídek pracovního trhu a poskytuje prvky pro okamžité kontaktování zaměstnavatele v případě zájmu o pracovní pozici. Ovládání aplikace je velice přehledné a intuitivní proto by neměl být problém se v aplikaci rychle zorientovat.

#### 3.2 Funkcionalita aplikace

Následující Use-case diagram přehledně zobrazuje případy užití aplikace.



Obrázek 8: Use-case diagram aplikace NovaPrace

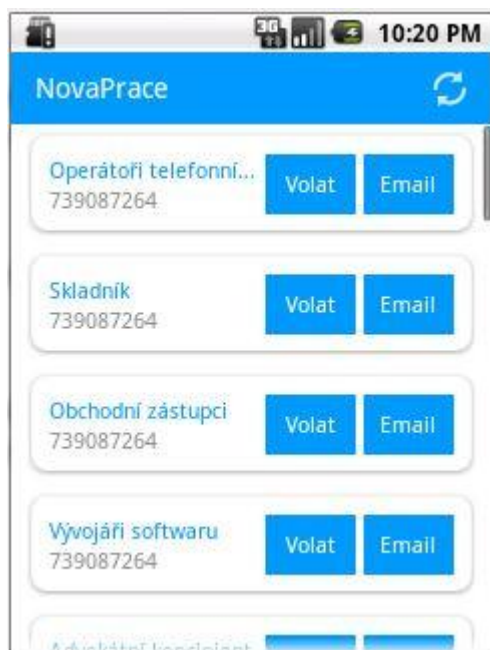
Zdroj: Vlastní zpracování

- **Zobrazit nabídky práce** – funkce zobrazit nabídky práce probíhá automaticky po zapnutí aplikace, zobrazí seznam položek pracovních nabídek

- **Zobrazit detail pracovní pozice** – tato funkce umožňuje zobrazit detailní popis pracovní nabídky po kliknutí na nabídku ze seznamu nabízených míst
- **Nastavit filtr nabídek** – funkce vyvolá aktivitu se seznamem oborů pracovní činnosti společně s checkboxy pro výběr, o které obory má uživatel zájem
- **Aktualizovat nabídky** – po stisku aktualizčního tlačítka aplikace obnoví seznam pracovních pozic
- **Reagovat e-mailem** – po stisku tlačítka na posláni e-mailu aplikace vyvolá záměr, který převezme některá z aplikací pro elektronickou poštu
- **Reagovat telefonním hovorem** – obdobně jako u e-mailu, aplikace vyvolá záměr na vytočení čísla, které je uvedeno u pracovní nabídky

### 3.3 Interface aplikace

Po zapnutí aplikace se uživateli zobrazí aktivita *JobsActivity*, v souboru *AndroidManifest.xml* je nastavena jako hlavní. V horní části zobrazení se nachází panel s názvem aplikace NovaPrace v levé části a s tlačítkem aktualizovat v pravé části panelu. Zbytek aktivity slouží pro zobrazení seznamu položek. Každá položka seznamu je tvořena vlastním layoutem a obsahuje název pozice, telefonní číslo a tlačítka pro kontaktování pomocí hovoru nebo zaslání emailu na danou nabídku.



Obrázek 9: Aktivita JobsActivity

Zdroj: Vlastní zpracování

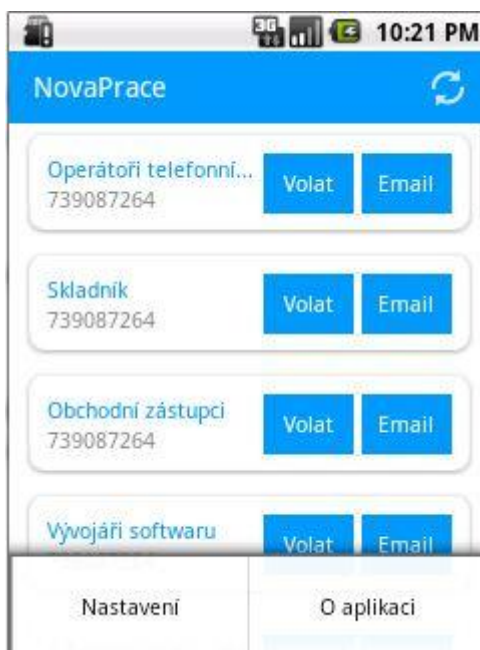
Po kliknutí na položku pracovní nabídky se zobrazí aktivita *DetailActivity*. Obsahuje taktéž panel s názvem Detail práce pro přehlednost, ve zbytku se zobrazí veškeré informace z databáze, které o této nabídce byly uvedeny.



Obrázek 10: Aktivita DetailActivity

Zdroj: Vlastní zpracování

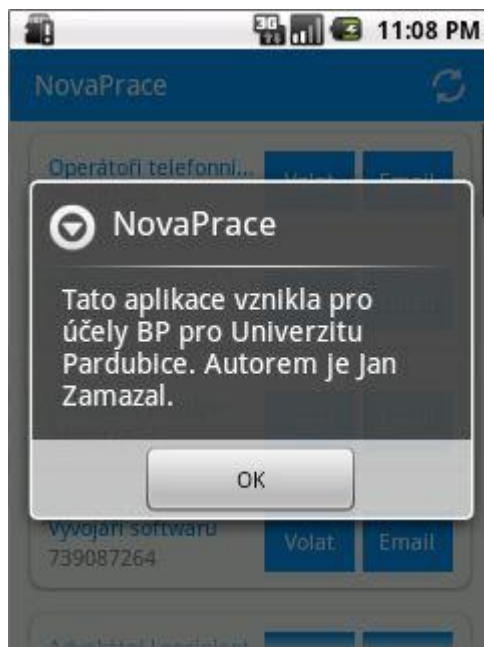
Z detailního pohledu na pracovní nabídku je možné se dostat tlačítkem zpět opět do hlavní aktivity. Další funkce jsou přístupné po stisku tlačítka menu. Zde dochází k odlišnému zobrazení na různých verzích systému Android. Ve verzi Android 2.1 Eclair (API úroveň 7) vypadá zobrazení menu takto.



Obrázek 11: Zobrazení menu na verzi Android 2.1 Eclair

Zdroj: Vlastní zpracování

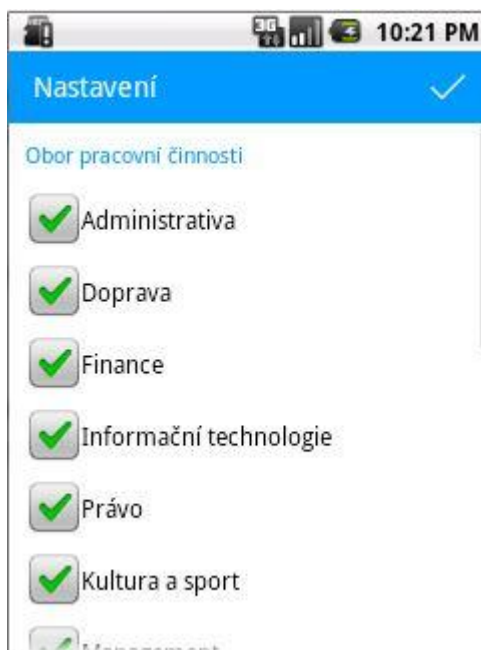
Na nejnovějších zařízeních, kde chybí hardwarová tlačítka, bude menu implementováno na panelu v podobě tří vertikálně orientovaných teček, což ovšem na funkcionalitu nemá žádný dopad. Nabídka menu obsahuje možnosti „Nastavení“ a „O aplikaci“. Pokud je zvolena volba O Aplikaci, objeví se krátký dialog, viz Obrázek 12.



Obrázek 12: Dialog "O aplikaci"

Zdroj: Vlastní zpracování

Poslední aktivitou je *SettingsActivity*, která je zobrazena při volbě Nastavení v menu.



Obrázek 13: Aktivita SettingsActivity

Zdroj: Vlastní zpracování

Aktivita obsahuje opět panel s názvem a tlačítkem pro potvrzení výběru a dále výčet kategorií se zaškrťovacími políčky pro výběr oborů pracovních činností, o které by měl uživatel zájem.

### 3.4 AndroidManifest.xml projektu

Soubor *AndroidManifest.xml* projektu NovaPrace vykazuje veškerá specifika projektu. Projekt byl vytvořen pro minimální API úroveň 7 (odpovídá verzi Android 2.1 Eclair) a je kompatibilní i s nejnovějším API úrovní 17. Při instalaci na zařízení bude aplikace žádat povolení přístupových práv k systémovým službám „INTERNET“ a „CALL\_PHONE“, tedy síťové připojení a uskutečňování hovorů. Komponenty projektu tvoří tři aktivity, přičemž u aktivity *JobsActivity* je nastaven filtr záměru určující její spuštění hned po startu aplikace. Všechny aktivity jsou definovány pouze v režimu portrét, při otočení displeje na šířku aktivita v tomto režimu zobrazení zůstane.

Takto vypadá celý manifest projektu:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cz.upce.android.novaprace"
    android:installLocation="preferExternal"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="7"
        android:targetSdkVersion="17" />

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CALL_PHONE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="cz.upce.android.novaprace.JobsActivity"
            android:label="@string/app_name"
            android:screenOrientation="portrait" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="cz.upce.android.novaprace.DetailActivity"
            android:screenOrientation="portrait" >
        </activity>
        <activity
            android:name="cz.upce.android.novaprace.SettingsActivity"
```



```
        android:screenOrientation="portrait" >
    </activity>
</application>

</manifest>
```

## 3.5 Součásti projektu

Celý projekt se skládá ze tří částí, první část tvoří aplikace NovaPrace, druhou součástí je pomocný projekt *ActionBarSherlock* a třetí část tvoří databáze údajů.

### 3.5.1 ActionBarSherlock<sup>63</sup>

*ActionBarSherlock* je rozšířením pomocných knihoven vytvořených pro použití prvku *ActionBar* i ve starších verzích systému, kde ještě nebyl podporován. Je zakomponován v projektu NovaPrace jako pomocná knihovna. V zařízeních s API úrovně 11 a výších bude použita nativní knihovna *ActionBar*, zatímco v nižších úrovních API bude prvek panelu vykreslován pomocí této podpůrné knihovny. Použití ve zdrojovém kódu projektu vypadá tak, že každá vytvořená aktivita je potomkem třídy *SherlockActivity* a interakce s panelem je prováděna pomocí metody *getSupportActionBar()*.

*ActionBarSherlock* je distribuován pod Apache License 2.0 a je dostupný na adrese <http://actionbarsherlock.com>.

### 3.5.2 Databáze

Součástí projektu je fiktivní databáze pracovních nabídek vytvořená pro demonstraci funkcionality. Databáze je umístěna na freehostingovém serveru, kde byla pro tyto účely zaregistrována doména [novaprace.wz.cz](http://novaprace.wz.cz). Přístup k databázi je přes rozhraní phpMyAdmin na adrese <https://www.webzdarma.cz/mysql/>. Databáze byla naplněna dvaceti pěti fiktivními případy.

- Přihlašovací jméno: novaprace97
- Heslo: admin

Databáze je tvořena jedinou relací Job (**id**, title, info, phoneNumber, email, category, requirements).

---

<sup>63</sup> *ActionBarSherlock* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://actionbarsherlock.com/>

### **3.6 Zhodnocení projektu**

Projekt NovaPrace přináší velice přívětivé uživatelské rozhraní při hledání nové pracovní pozice a mohl by najít využití například ve spolupráci s okresními úřady práce, kde by stačilo pro tyto účely vytvořit a spravovat databázi reálných pracovních nabídek. Stávající verze aplikace by pak prošla úpravou pro přijímání nové datové struktury z jiného serveru a mohla by být volně šiřitelná pro zájemce o novou pracovní pozici.

Aplikace je kompatibilní se všemi systémy Android od verze 2.1, což je podle kapitoly 1.3.11 99,8% mobilních zařízení, na kterých běží operační systém Android a proto by mohla být velmi dobře šiřitelná.

Prostředí Eclipse s pluginem ADT naprosto splňuje veškeré nároky na tvorbu projektů pro vývoj aplikací na mobilní operační systém Android. Emulátor, který je součástí vývojového prostředí dovoluje testovat téměř všechny funkce, některé je však nutné testovat přímo ve fyzickém zařízení.

## ZÁVĚR

Cílem práce bylo poukázat na rozvíjející se platformu Android, která každým dnem získává stále větší podíl na trhu s mobilními telefony a v poslední době také tablety. Síla Androidu spočívá především ve snaze otevřít veškeré vývojové prostředky a poskytnout tak komplexní vývojovou platformu veřejnosti. Pro tyto účely je vytvořen internetový průvodce na adrese <http://developer.android.com/>, kde je celé API detailně popsáno a názorně na příkladech demonstrováno od instalace požadovaného softwaru, přes základní znalosti o naprogramování jednoduché aplikace „Hello\_world“, až po nejobtížněji zvládnutelné komponenty, které vývojář může ve svém projektu využít. Nutno dodat, že existuje také celá řada knižních publikací, ke kterým jsou k dispozici rozsáhlé příklady zdrojových kódů a také nesčetně diskusních fór, kde se dá vždy sehnat dobrá rada.

Další silnou stránkou je propojení programovací jazyka Java se značkovacím jazykem XML, který do velké míry zjednoduší vytváření uživatelského rozhraní. Vývojář si jednoduše pomocí XML rozvrhne, jakým vzhledem bude jeho aplikace disponovat. Nejnáročnější znalostí při psaní potom zůstává zvládnutí základních principů objektového programovacího jazyku Java.

Využití aplikací si neklade příliš meze, mobilní zařízení smartphone je vybaveno mnoha hardwarovými komponenty, které se dají použít pro nejrůznější účely. Nejvyužívanějšími prvky zatím zůstávají kamera, audio/video a připojení k síťovým službám, v současnosti se ale začínají využívat i senzorické služby. Dopředu také výrazně pokročily výkonnostní komponenty, jako CPU, GPU a velikost operační paměti. Nejnovější smartphony už disponují slušným výkonem i pro graficky náročné aplikace a tím se platforma otevírá i pro herní průmysl.

Jak jsem názorně v této práci demonstroval, mohou Android aplikace najít využití i ve veřejné správě. Přesná a rychlá informace je základním kamenem pro zvyšování efektivity snad v každém odvětví. Procesy veřejné správy budou muset dříve či později konvertovat stále více do elektronické podoby a tak by využití mobilních zařízení mohlo nalézt další uplatnění při komunikaci s orgány veřejné správy.

Z historických údajů jasně vyplývá vysoká popularita a potenciál Androidu stát se dominantou mezi ostatními poskytovateli operačních systémů či komplexních platforem pro mobilní zařízení.

## POUŽITÁ LITERATURA

- [1] MURPHY, Mark. *Beginning Android*. [s.l.]: Apress, Inc., 2009. 361 s. ISBN 978-1-4302-2420-4.
- [2] MURPHY, Mark L.. *Android 2 : průvodce programováním mobilních aplikací*. Brno: Computer Press, 2011. ISBN 978-80-251-3194-7
- [3] HEROUT, Pavel. *Učebnice jazyka Java*. České Budějovice: Kopp, 2010. ISBN 978-80-7232-398-2.
- [4] *Professional Android 2 application development*. Indianapolis: Wiley, 2010. ISBN 978-0-470-56552-0.
- [5] BEZROUKOV, Nikolai. *Android. Softpanorama* [online]. 2013 [cit. 2013-08-07]. Dostupné z: [http://www.softpanorama.org/Commercial\\_linuxes/Android/](http://www.softpanorama.org/Commercial_linuxes/Android/)
- [6] ČERNÝ, Michal. *Operační systém Android. Metodický portál RVP.CZ* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://clanky.rvp.cz/clanek/o/g/15449/OPERACNI-SYSTEM-ANDROID.html/>
- [7] MARVAN, Filip. *Mobilní operační systém Android. DIIT.CZ* [online]. 2011 [cit. 2013-08-07]. Dostupné z: <http://diit.cz/clanek/mobilni-operacni-system-android>
- [8] *Android ovládl trh. Jeho podíl vystoupal na 75 %, poslední z velkých firem je Microsoft s Windows. DSL.CZ* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://www.dsl.cz/clanek/2754-android-ovladl-trh-jeho-podil-vystoupal-na-75-posledni-z-velkych-firem-je-microsoft-s-windows>
- [9] MYSLIVEČEK, David. *Krátké ohlédnutí za historií Androidu. Svět Androida* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://www.svetandroida.cz/kratke-ohlednuti-za-historii-androidu-201305>
- [10] *Open Handset Alliance. Open Handset Alliance* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://www.openhandsetalliance.com/>
- [11] *COMPLETE HISTORY OF ANDROID. ANSAR MUHAMMED BASHIR* [online]. 2011 [cit. 2013-08-07]. Dostupné z: <http://eceansar.blogspot.cz/2011/07/complete-history-of-android-successthe.html>

- [12] JEŽEK, David. Google uvedl Android 4.0. *DIIT.CZ* [online]. 2011 [cit. 2013-08-07]. Dostupné z: [http://diit.cz/clanek/google-vedl-android-40#utm\\_source=readmore&utm\\_medium=textlink&utm\\_campaign=readmore](http://diit.cz/clanek/google-vedl-android-40#utm_source=readmore&utm_medium=textlink&utm_campaign=readmore)
- [13] KŮŽEL, Filip a Radek JANATA. Vyzkoušeli jsme Android 4.2 Jelly Bean: novinky, dojmy. *Mobilmania.cz* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://www.mobilmania.cz/clanky/vyzkouseli-jsme-android-42-jelly-bean-novinky-dojmy/sc-3-a-1322229/default.aspx>
- [14] Jak vznikl maskot Androida?. *Mobilmania.cz* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://www.mobilmania.cz/bleskovky/jak-vznikl-maskot-androida-puvodni-navrhy-byly-desive/sc-4-a-1322689>
- [15] Android Architecture – The Key Concepts of Android OS. *Android-App-Market.com* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://www.android-app-market.com/android-architecture.html>
- [16] Android Market končí, přichází Google Play: co je nového?. *Mobilenet.cz* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://mobilenet.cz/clanky/android-market-konci-prichazi-integrovan-y-google-play-co-je-noveho-8741>
- [17] *O službě Google Play. Support Google* [online]. 2013 [cit. 2013-08-07]. Dostupné z: [https://support.google.com/googleplay/answer/2490014?hl=cs&p=play\\_faq&rd=1](https://support.google.com/googleplay/answer/2490014?hl=cs&p=play_faq&rd=1)
- [18] AndroidWiki. *AndroidWiki* [online]. 2012 [cit. 2013-08-07]. Dostupné z: <http://androidwiki.cz/>
- [19] Developer's Guide. *Android Developers* [online]. 2013 [cit. 2013-08-07]. Dostupné z: <http://developer.android.com/index.html>

## **SEZNAM PŘÍLOH**

Příloha A, kompletní zdrojový kód projektu NovaPrace.