

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO  
KATEDRA INFORMATIKY

## BAKALÁŘSKÁ PRÁCE

Aplikace pro vyhledávání nemovitostí pro Android

Estate search application for Android



2012

Roman Tiefenbach

## **Anotace**

*Bakalářská práce se zabývá vyhledáváním a zobrazováním dat z realitního serveru na základě uživatelem definovaných filtrů. Důraz je přitom kladen na pohodlné a přímočaré ovládání celé aplikace, která je cílená na mobilní platformu s operačním systémem Android. Aplikace využívá on-line přístupu na realitní server pomocí webových služeb. Pro snazší zadávání filtrů mobilní zařízení poskytují svoji přibližnou aktuální polohu. Vybraná data lze uchovávat off-line pro pozdější možnost opětovného prohlížení, včetně ukládání jednotlivých vyhledávacích filtrů z oblasti zájmu uživatele. Součástí práce je i popis platformy Android, jakožto i možnosti vývoje pro tuto platformu.*

Tímto děkuji vedoucímu bakalářské práce Mgr. Janu Outratovi, Ph.D., Michalu Valtovi (jednateli a projektovému manageru firmy viaCentrum s.r.o.) za umožnění vzniku této práce, technické zázemí a podporu, stejně tak jako svým kolegům Mgr. Petru Kašpárkovi, Mgr. Vladimíru Pindřurovy a Mgr. Vítu Zaoralovi, za odbornou pomoc, náměty a cenné připomínky. Děkuji své ženě Zdeňce a rodině za podporu.

# Obsah

<b>1. Úvod</b>	<b>8</b>
<b>2. Mobilní zařízení</b>	<b>9</b>
2.1. Apple iOS . . . . .	9
2.2. Windows Phone . . . . .	9
2.3. Android . . . . .	10
<b>3. Platforma Android</b>	<b>11</b>
3.1. Historie . . . . .	11
3.2. Architektura . . . . .	11
3.2.1. Jádru OS (Linux kernel) . . . . .	12
3.2.2. Běhové prostředí (Runtime) . . . . .	12
3.2.3. Knihovny (Libraries) . . . . .	13
3.2.4. Aplikační rámec (Application Framework) . . . . .	13
3.2.5. Aplikace . . . . .	14
3.3. Základní principy pro Android . . . . .	14
3.3.1. The Manifest File . . . . .	15
3.4. Komponenty pro aplikace Android . . . . .	15
3.4.1. Aktivita – Activities . . . . .	15
3.4.2. Služby – Services . . . . .	16
3.4.3. Poskytovatelé obsahu – Content Providers . . . . .	17
3.4.4. Přijímače žádostí – Broadcast Recievers . . . . .	17
3.5. Historie verzí . . . . .	17
<b>4. Vývojářské prostředky</b>	<b>18</b>
4.1. Android SDK . . . . .	18
4.2. Java (Eclipse) . . . . .	19
4.3. Adobe (Flex) . . . . .	20
4.4. Xamarin (Mono for Android) . . . . .	23
4.5. HTML5 . . . . .	23
<b>5. Vývojové prostředí, instalace a zkušenosti</b>	<b>25</b>
5.1. Příprava a instalace . . . . .	25
5.2. Zkušenosti s vývojem, emulátorem, GUI a fyzickými zařízeními . . . . .	26
5.2.1. Emulátor . . . . .	27
5.2.2. Tvorba GUI . . . . .	27
5.2.3. Připojení fyzických zařízení . . . . .	28
<b>6. Aplikace vyhledávání nemovitostí</b>	<b>29</b>
6.1. Omezení oblastí funkcí aplikace . . . . .	29
6.2. Konkurence a existující řešení . . . . .	30

<b>Závěr</b>	<b>34</b>
<b>Reference</b>	<b>35</b>
<b>A. Uživatelská příručka</b>	<b>37</b>
A.1. Instalace . . . . .	37
A.2. Scénář první – nalezení nemovitosti podle viaKódu, její zobrazení na mapě a posláni dotazu . . . . .	37
A.3. Scénář druhý – hledání nemovitostí vytvořením filtru . . . . .	38
A.4. Scénář třetí – hledání nemovitostí v okolí . . . . .	39
A.5. Scénář čtvrtý – hledání nemovitostí realitní kanceláře . . . . .	39
A.6. Scénář pátý – využívání uložených informací . . . . .	41
<b>B. Programátorská dokumentace</b>	<b>42</b>
B.1. GUI . . . . .	42
B.1.1. Screens . . . . .	42
B.1.2. Adapters . . . . .	43
B.1.3. Aplikační vrstva (AL) . . . . .	43
B.1.4. Resources . . . . .	44
B.2. Knihovna .dll . . . . .	44
B.2.1. Business vrstva (BL) . . . . .	45
B.2.2. Datová vrstva (DL) . . . . .	45
B.2.3. Helpers . . . . .	46
B.2.4. Extensions . . . . .	46
B.2.5. Webové reference . . . . .	46
<b>C. Obsah přiloženého CD</b>	<b>47</b>

## Seznam obrázků

1.	viaReality - vyhledávání nemovitostí. . . . .	8
2.	Obrázek architektury platformy Android. . . . .	12
3.	Životní cyklus aplikace. . . . .	16
4.	Android SDK manager zobrazující dostupné balíčky pro jednotlivé verze platform. . . . .	19
5.	Nástroj DDMS. . . . .	20
6.	Vývojové prostředí Flash Builder - kódová část. . . . .	21
7.	Flash Builder - debug a simulátor. . . . .	22
8.	Instalace Mono for Android. . . . .	26
9.	Visual Studio s nastavením Mono for Android. . . . .	27
10.	RealCity - seznam nemovitostí, nový filtr, oblíbené, detail nemovitosti. (Poslední dobou mi aplikace přestala zobrazovat fotografie – možná potřeba aktualizace) . . . . .	31
11.	HyperReality - seznam nemovitostí, nový filtr, historie, detail nemovitosti. . . . .	31
12.	Reality - seznam nemovitostí, nový filtr, historie, detail nemovitosti. . . . .	33
13.	dmn:sReality - seznam nemovitostí, nový filtr, hlavní obrazovka, detail nemovitosti – přelom mezi popisem a galerií (horizontální posouvání). . . . .	33
14.	Scénář 1 – zadání viaKódu, detail nalezené nemovitosti, její zobrazení na mapě, poslání dotazu k nemovitosti. . . . .	38
15.	Scénář 2 – nový filtr, zadání polohy, zadání ceny, výsledný filtr. . . . .	39
16.	Scénář 3 – hlavní obrazovka aplikace (Mé okolí), nalezené nemovitosti, detail nemovitosti (jeho spodní část s funkcemi), zobrazení podobných nemovitostí k právě prohlížené. . . . .	40
17.	Scénář 4 – hledání realitní kanceláře, detail realitní kanceláře, zobrazení jejích nemovitostí, filtrování nemovitostí konkrétní realitní kanceláře. . . . .	40
18.	Scénář 5 – seznam oblíbených, úprava seznamu naposledy navštívených, kontextová nabídka nad seznamem filtrů, vyvolání menu nad seznamem filtrů. . . . .	41
19.	Komunikace SOAP. . . . .	47

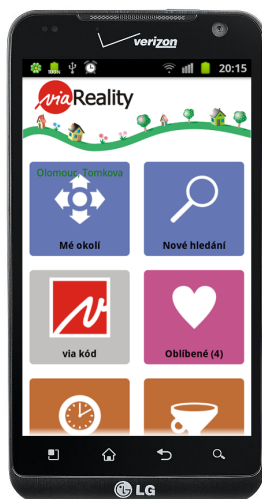
## Seznam tabulek

- |    |   |    |
|----|---|----|
| 1. | Tabulka verzí OS Android s procentuálním zastoupením. . . . . | 17 |
| 2. | Tabulka porovnávající vybranou funkčnost 5 aplikací. . . . .  | 32 |

# 1. Úvod

Každý někdy hledal nemovitost. Ať už z důvodů stěhování, koupě, investice, či jen tak pro udržení si podvědomí o cenách realit. Určitě mi dáte za pravdu, že hledat na více místech se vyplácí, ale stojí více času. Hledat v inzertních novinách nebo časopisech v dobách internetu je pomalé, i když stále má své příznivce. Hledání na internetu, když nevíte kde a jak, je problém. Pokud nejste fandou konkrétní realitní kanceláře, nebo nechcete „surfovat“ přes všechny stránky realitních kanceláří v okolí, chce to realitní portál. A to takový, který podporuje vyhledávání podle parametrů, co vás zajímají, má velkou základnu a širokou škálu nemovitostí, ať je z čeho vybírat. A pokud vás nudí opakovaně zadávat jedny a tytéž parametry vyhledávání den za dnem, obzvláště pokud máte konkrétní představu, bylo by dobré nějak tyto „filtry“ ukládat a později se k nim vracet.

Přestože většinu této funkčnosti v jisté podobě již realitní portály pro „dospělé“ počítače splňují, stále více lidí tráví na cestách spoustu času, a pokud ho nechce trávit poslechem hudby a sledováním okolí, může mimo jiné třeba brouzdat po nemovitostech. Tudíž další logický krok těchto realitních portálů, jak nabídnout nemovitosti širší komunitě lidí, je skrze mobilní zařízení, jež nabírají na oblíbenosti. Ta mají menší displeje, horší rozlišení, většinou omezené připojení na internet a hlavní limitací je baterie těchto zařízení. I z těchto důvodů jsou webové stránky pro mobilní webové klienty co nejvíce jednoduché, bez zbytečné grafiky a patřičně okrájené. Asi úplně nejlepší řešení v této situaci je aplikace v nativním kódu šitá na míru vašemu zařízení.



Obrázek 1. viaReality - vyhledávání nemovitostí.



## 2. Mobilní zařízení

Pryč jdou doby, kdy se za malé „chytré bričky neboli smartphony“ s dotykovým displejem málem platilo zlatem. Dnes jsou již dostupné téměř všem, i za rozumnou cenu, dle výběru zájemce. Na trhu je nepřeberné množství modelů, barev, technologií i designů těchto smartphonů. Obdobně je to i se softwarovým vybavením těchto zařízení. Líbí se vám tato funkčnost? Nebo tento styl? Jsem si jistý, že si vyberete.

Na trhu, co se operačních systémů (OS) pro ony „chytré“ mobilní telefony týče, podle [1] nyní převládá Android nad Symbianem, iOS, BlackBerry, MeeGo, Windows Phonem (dříve Windows Mobile) a Badou. Symbian od Nokie pomalu upadá a byl jí opuštěn ve prospěch Windows Phone. BlackBerry byl a je zaměřen vesměs pro business třídu a vyznačuje se hardwarovou klávesnicí. I z těchto důvodů, když jsme ve firmě zvažovali pro který systém se rozhodnout, pro nás byli ve hře Windows Phone, Android a iOS od Applu.

### 2.1. Apple iOS

Velice „moderní“ platforma, těžící zejména z pouhých několika typů hardware a velice striktní politikou vývoje a publikování aplikací. Má líbivý, jednoduchý a účelný design, který, nutno říct, funguje. Aplikace jsou důkladně vyzkoušeny a jinak než z online obchodu Apple Storu je oficiálně nezískáte, tudíž uživatel iPhone má jistotu, že daná aplikace bude fungovat a bude dělat to co má.

Přestože na internetových stránkách [5] je podrobný návod jak vyvíjet aplikace pod touto platformou, jednoduše vychází z toho, že vlastníte zařízení s Mac OS X. Jinak máte smůlu. Narážíte na problém registrace mobilního zařízení pro vývoj, získání certifikátů potřebných k podpisu aplikace, nemluvě o nutnosti zvládnutí Xcode nebo Objective-c. Samozřejmě existují postupy jak tvořit programy i pod jinými OS než Mac OS X, nikoliv však přímočaré. Holt je to platforma zaměřená na svoje uživatele, za cenu méně dostupného vývoje.

### 2.2. Windows Phone

Windows Phone ve spojení s novými zařízeními Nokia vypadá velice slibně, přestože nepřináší na trh nic převratného. Zatím běží jen na několika typech hardwaru. Má kompletně nový design – Metro design. Tyto neustále se překlápějící dlaždice, nahrazující widgety, nemusí všem vyhovovat, ale v uceleném, unifikovaném a osobitém designu přinášejí automaticky nové a nové informace. I styl aplikací je na doporučení unifikovaný, jednoduchý, ve dvou barevných tématech. Ale jestli chcete psát proti proudu . . .

Podle mého názoru Microsoft vyráží podobným směrem jako Apple. Jediná možnost, jak získat novou aplikaci, je z online obchodu Windows Store, kde pro nově přijímané taktéž vládou přísné podmínky, ale zaručují další bezproblémový

chod. Vše je striktně nastavené na co největší výdrž zatím nejslabší části mobilních zařízení – baterky. Vlastní systém má v tomto duchu mnohá omezení, se kterými se vývojáři musejí srovnat. Na druhou stranu o to lépe pro uživatele. Asi největší předností vývoje pro tuto platformu je, že vše prostě funguje. Vlastní emulátor je dobře provázán s vývojovým prostředím. Vše je pěkně zdokumentované a popsané.

## 2.3. Android

Android je mladý a velice úspěšný OS, který byl představen v roce 2007. Od té doby se stal nejrozšířenějším OS na trhu s mobilními zařízeními pro smartphones. Nutno podotknout, že se jedná o open-source (založený na jádru Linux) podporovaný Googlem, navržený pro běh na různém hardwaru. Má integrované aplikace přímo od Googlu, jako Mapy, Kalendář, Gmail a webový prohlížeč. Jako u předešlých platforem i zde je možnost obstarání aplikací třetích stran přes online obchod Google play.

Co se vývoje pro Android týče nabízí efektivní nástroje v Software Development Kit (SDK) spolu s vlastním emulátorem pro testování nových aplikací. Bohužel tento emulátor nepatří mezi nejrychlejší. I když vše vypadá pěkně nachystané a naservírované pro nové programátory, narazil jsem při instalaci potřebných komponent na několik problémů popsaných v kapitole číslo 5. Jako oficiálně podporované vývojové prostředí je zvoleno Eclipse od Javy.

### 3. Platforma Android

Přestože se naše firma orientuje spíše na nástroje Microsoftu a výběr platformy Windows Phone pro mobilní aplikace by byla v tomto případě logická volba, vybral jsem si platformu Android. A to hned z několika důvodů. Jakožto nového uživatele této platformy (do té doby používající Windows Mobile) mě samozřejmě zajímají její možnosti, principy a způsob vývoje aplikací. A nejlepší cestou pro získávání informací z první ruky je si to vyzkoušet na vlastní kůži. Za další Android jako OS je mladý, a i za tuto krátkou dobu si našel spoustu příznivců a zákazníků napříč jejich spektrem (pyšní se přes 300mil používaných androidích zařízení). Pravda zatím převládá mladší generace, která je lákána hlavně přívětivou cenovou politikou zařízení s tímto OS.

Podle [2], [3] nebo [6] je Android open source platforma pro mobilní zařízení, která zahrnuje OS, uživatelské rozhraní a aplikace. Důraz je kladen na běh jádra na různém hardwaru (rozlišení, chipset, velikost), minimální spotřebu, menší nároky na výkon a paměť.

Platforma poskytuje kompletní řešení nasazení OS (specifikace driverů aj.) pro mobilní operátory a výrobce zařízení, stejně jako vývojářům poskytuje efektivní nástroje pro vývoj - SDK.

Android Open Source Project (AOSP) – je veden společností Google, která se stará o údržbu a vývoj této platformy. AOSP se taktéž stará o *Android Compatibility Program*, který definuje zařízení plně kompatibilní s platformou Android.

#### 3.1. Historie

Firma Android Inc. vznikla v Kalifornii koncem 2003 a v roce 2005 ji jako neznámou odkoupil Google Inc. a udělal z ní svoji dceřinou společnost. Byla vyvinuta platforma na linuxovém jádře a když ke konci 2007 Google získal několik patentů v oblasti mobilních technologií, začalo se spekulovat o vstupu Googlu na trh chytrých mobilních telefonů s vlastním modelem.

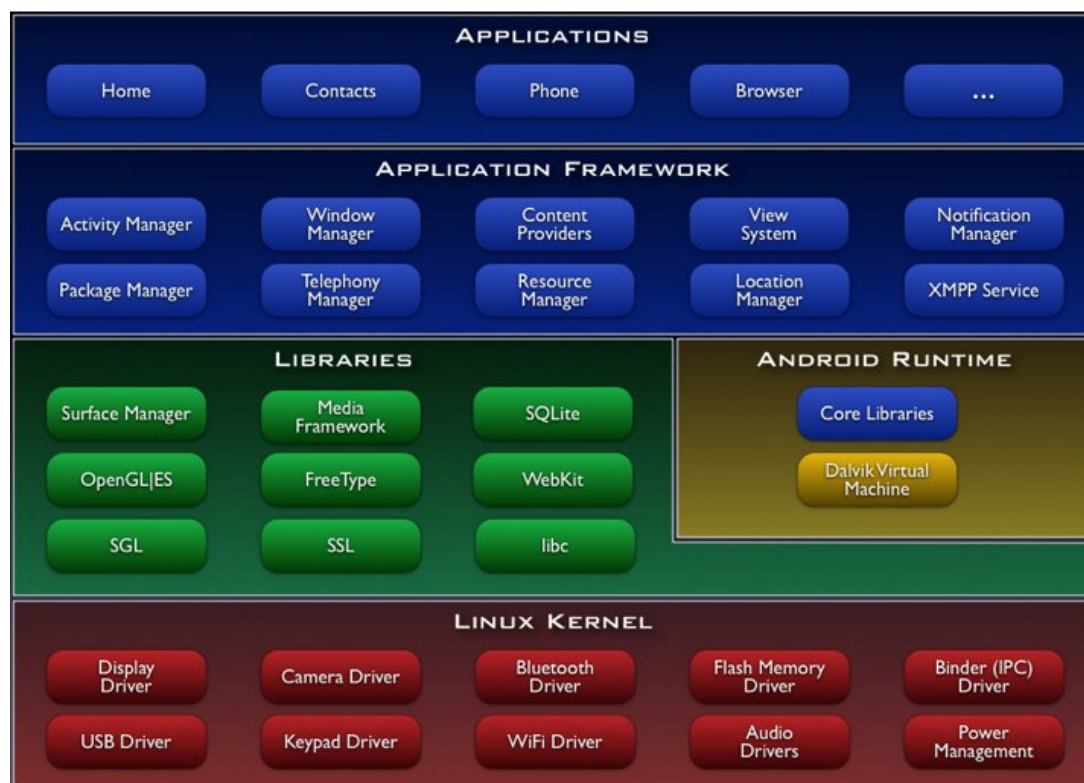
Později v 2007 bylo utvořeno konsorcium několika (nyní asi 85) společností – Open Handset Alliance, zahrnující hardwarové, softwarové a telekomunikační firmy (Google, HTC, Intel, Samsung, LG, Texas Instruments, ...). Cílem bylo vytvoření otevřeného standartu pro mobilní zařízení. Ještě téhož dne konsorcium odhalilo svůj první produkt – Android, platformu pro mobilní zařízení postavenou na jádře Linux 2.6.

Koncem roku 2008 byl v USA uveden první komerční telefon (HTC) s OS Android. V roce 2009 jich bylo již více než 20 druhů.

#### 3.2. Architektura

Architektura (znázorněna na obrázku 2.) je, jak popisuje [8] nebo [2], rozdělena do 5 vrstev z nichž každá má svůj účel a nemusí být nutně oddělena od

ostatních vrstev. Probereme je trochu podrobněji.



Obrázek 2. Obrázek architektury platformy Android.

Zdroj: <http://developer.android.com>

### 3.2.1. Jádru OS (Linux kernel)

Abstraktní vrstvu mezi hardwarem a softwarem používaném ve vyšších vrstvách tvoří upravené Linuxové jádro ve verzi 2.6, které tvoří jádro OS. Využívá se z něj mnoha vlastností (zpráva pamětí, sítí, procesů, ...) spolu z omezenou sadou GNU knihoven, zato vůbec neobsahuje grafické uživatelské rozhraní X Window System. Omezená sada knihoven činí problémy s portováním existujících Linux aplikací nebo knihoven na Android. Důvodem použití Linuxového jádra bylo jednoduché sestavení na různých zařízeních a tím zajištěna snadná přenositelnost.

### 3.2.2. Běhové prostředí (Runtime)

Aplikační virtuální stroj zvaný Dalvik (DVM – Dalvik Virtual Machine) byl speciálně vyvíjen pro Android již od 2005. Důvody vzniku nového virtuálního stroje byly licenční práva k JVM (Java Virtual Machine) a optimalizace pro mobilní zařízení zejména v oblasti poměru úspory energie a výkonu.

Je to registrově orientovaná architektura využívající základních vlastností Linuxového jádra (správa paměti, práce s vlákny). Jsou zde obsaženy základní knihovny jazyku Java, vyjma knihoven uživatelského rozhraní (AWT, Swing), které byly nahrazeny knihovnami uživatelského rozhraní Android. Byly přidány knihovny Apache pro práci se sítí.

Každá spuštěná aplikace běží na vlastním procesu s vlastní instancí DVM. Při překladu aplikace napsané pro Android se nejdříve zkompile zdrojový Java kód do Java byte kódu, ten je následně překompilován Dalvik kompilátorem a výsledný Dalvik byte kód je spuštěn v DVM.

### 3.2.3. Knihovny (Libraries)

Další vrstvou jsou knihovny, které využívají různé komponenty systému. Jejich funkce jsou vývojářům dále poskytnuty v *Application Frameworku*. Jsou psány v C/C++. Zde uvádím některé z nich.

*Media Libraries* – knihovna podporující přehrávání video a audio formátů spolu s formáty obrázků.

*LibWebCore* – knihovna webového prohlížeče.

*SQLite* – knihovna odlehčené relační databáze.

*FreeType* – knihovna na renderování vektorových i bitmapových fontů.

*OpenGL* – knihovna na vykreslování počítačové grafiky.

### 3.2.4. Aplikací rámec (Application Framework)

Application framework je nejdůležitější vrstvou pro vývojáře Android aplikací. Poskytuje stejná přístupová rozhraní, které využívají základní vestavěné aplikace. Tím se plně otevírá cesta k velkému počtu služeb jako zpřístupnění hardwaru daného zařízení, informace o poloze, spouštění služeb na pozadí, nastavování upozornění do stavové lišty, zpřístupňování dat v jiných aplikacích, zpřístupnění uživatelského rozhraní, ... Pod všemi aplikacemi je sada systémů a služeb včetně následujících.

*Sada prvků View* – prvky pro sestavení vizuální podoby aplikace. Obsahuje seznamy, tlačítka, textová pole, nebo vestavěný webový prohlížeč.

*Content provider* – „poskytovatelé obsahu“ umožňují přístup k obsahu jiných aplikací (např. seznam kontaktů).

*Resource manager* – poskytuje přístup k „nekódovým“ zdrojům jako grafika, textové řetězce, pole, popis uživatelského rozhraní dané obrazovky.

*Notification manager* – povolí všem aplikacím zobrazovat vlastní upozornění ve stavovém řádku.

*Activity manager* – řídí životní cyklus aplikací (znázorněn na obrázku 3.) a poskytuje základní prvky pro navigaci v zásobníku obrazovek.

### 3.2.5. Aplikace

Android je dodáván se základními vestavěnými aplikacemi zahrnující e-mail klienta, posílání SMS zpráv, kontakty, kalendář, mapy, webový prohlížeč a mnoho dalších. Všechny tyto aplikace jsou napsány v programovacím jazyce Java.

## 3.3. Základní principy pro Android

Aplikace pro Android jsou psány v Javě. Android SDK tools (kap. 4.1.) zkompiluje kód spolu se všemi daty a vstupními zdroji a vytvoří Android instalační balíček s příponou .apk.

Každá nainstalovaná aplikace je v bezpečně odděleném prostoru „sandboxu“ a platí pro ní:

- Android OS je více-uživatelský Linuxový systém, kde každá aplikace představuje jiného uživatele.
- Standardně je každé aplikaci přiřazeno unikátní ID uživatele (je známo pouze systému, nikoli uživateli) a pro veškeré soubory této aplikace jsou nastavena práva pro přístup pouze s tímto ID.
- Každý proces běží ve svém DVM, takže každá aplikace běží odděleně od ostatních.
- Standardně každá aplikace běží na svém procesu. Je spuštěn pokaždé, když je třeba vykonat nějakou z komponent aplikace, a naopak ukončen pokud již není třeba, nebo pokud systém potřebuje uvolnit paměť pro další aplikace.

Jde se cestou „nejmenších výsad“, kdy aplikace má přístup pouze ke komponentám, které potřebuje pro svůj běh a nic víc. To vytváří velice bezpečné prostředí, kde aplikace nemůže využívat části systému ke kterým nemá oprávnění.

Návrh systému Android umožňuje unikátní věc, kdy každá aplikace může spustit jakoukoli komponentu jiné aplikace. Například pokud chcete vyfotit obrázek pro svoji aplikaci, nemusíte si napsat aplikaci fotoaparátu, ale jednoduše využijete systémovou, nebo jakoukoli jinou, a ta vám obrázek vrátí. Nebo spuštění aplikace Mapy ve vaší aplikaci s daným místem k zobrazení taktéž využívá systémovou aplikaci.

Každá takto spuštěná komponenta aplikace je spuštěna v procesu příslušné aplikace, nikoli vaší. Má určitá práva a požaduje práva pro komunikaci, takže vaše aplikace nemůže aktivovat komponentu jiné na přímo. Lze to obejít žádostí na systém o její spuštění.

### 3.3.1. The Manifest File

Než systém spustí komponentu aplikace, musí vědět, že komponenta existuje přečtením `AndroidManifest.xml` souboru. Tento soubor obsahuje „textový popis“ aplikace, jsou zde uvedené všechny komponenty, práva, co aplikace, poskytuje, na co umí reagovat a mnoho dalšího (detailní popis je uveden [10]). Pro předcházení problémů by každá aplikace měla obsahovat:

- Minimální množinu všech povolení a oprávnění co aplikace nutně potřebuje od uživatele pro své fungování (přístup na internet, čtení kontaktů, ...).
- Uvést minimální verzi API potřebnou aplikací (pro jakou verzi byla napsána).
- Uvést hardwarové a softwarové nároky, používané nebo potřebné pro fungování aplikace (kamera, bluetooth, multitouchový displej, ...).
- A další.

## 3.4. Komponenty pro aplikace Android

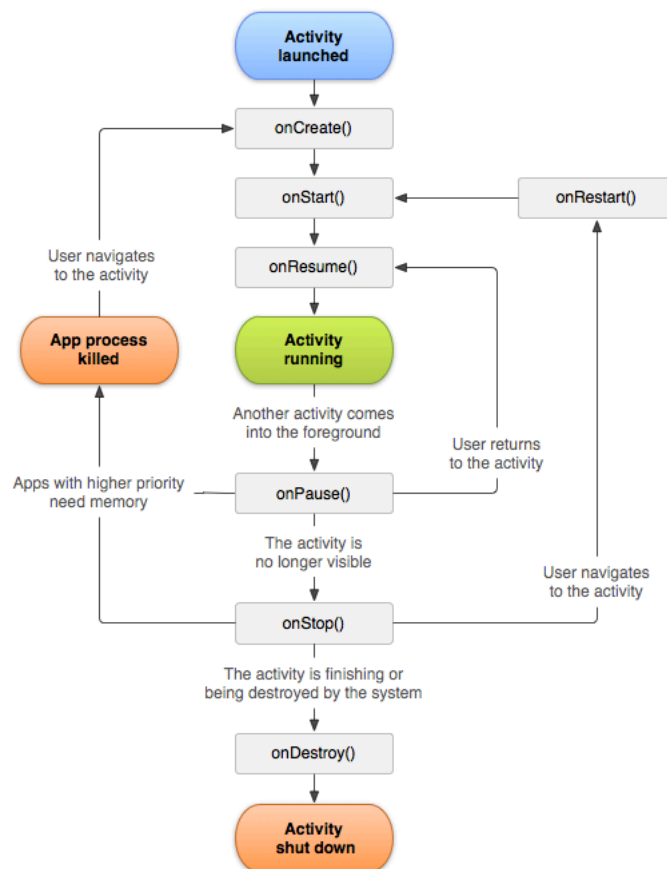
Komponenty jsou základní stavební kameny androidí aplikace ([7], [2]). Každá z nich je vstupní bod pro systém jak vstoupit do naší aplikace. Ne všechny komponenty jsou přístupovými body pro uživatele a některé závisí na ostatních, ale každá existuje jako entita a hraje specifickou roli.

Jsou 4 typy komponent pro tvorbu aplikací. Každá slouží jinému účelu a má jiný životní cyklus, který určuje jak komponenta vzniká a zaniká.

### 3.4.1. Aktivita – Activities

Aktivita odpovídá jedné obrazovce a obsahuje grafické uživatelské rozhraní pro interakci. Aplikace obvykle obsahuje více aktivit, které jsou zodpovědné za určitou činnost. Třeba u e-mailového klienta máme jednu aktivitu pro seznamem mailů, další pro otevřený detail e-mailu pro čtení a další pro vytvoření nového e-mailu. Jednotlivé aktivity lze spouštět přímo, takže když je vše správně nastaveno, lze z jiné aplikace spustit přímo zadávání nového emailu.

Zahájení aktivity je pro systém poměrně náročná činnost, kdy je třeba spustit nový proces, alokovat paměť pro objekty uživatelského rozhraní, které se rozloží do „layoutu“ a připravenou obrazovku vyvolat pro zobrazení. S tím nám pomáhá Activity manager, který se stará o vytváření, rušení a celkovou správu životního



Obrázek 3. Životní cyklus aplikace.  
Zdroj: <http://developer.android.com>

cyklu aktivity (který je znázorněn na obrázku 3.). To má výhodu pokud dojde k neošetřené výjimce, kdy se vám neukončí celá aplikace, ale jen daná aktivita a ze zásobníku otevřených aktivit se vytáhne předcházející.

### 3.4.2. Služby – Services

Služba je komponenta běžící na pozadí, která má na starosti dlouho trvající operace, nebo přístup ke vzdáleným zdrojům, kde není jistá doba odezvy. Služba neposkytuje uživatelské rozhraní. Příkladem služby může být přehrávání hudby na pozadí, zatímco uživatel používá jinou aplikaci, nebo stahuje data bez blokování interakce s uživatelem.

Služba může být spuštěna metodou startService, kdy se následně může ukončit sama, nebo být ukončena jinou komponentou. Nebo metodou bindService, kterou vyvolá komponenta (klient). Takovou službu může ukončit pouze tento klient



odhlášením se. Ke službě může být v jednu chvíli připojeno i více klientů, po jejich odhlášení se služba ukončí.

### 3.4.3. Poskytovatelé obsahu – Content Providers

Poskytovatelé obsahu je aplikační prostředí pro sdílení dat mezi aplikacemi, i jakožto sdílení dat v rámci jedné aplikace mezi aktivitami. Data mohou být uchovávána v souborech, v databázi SQLite, nebo na webu, přesto s patřičnými povoleními lze k těmto datům přistupovat pomocí jiných aplikací. Pak je možné například číst či dokonce měnit kontakty skrze vaši aplikaci.

### 3.4.4. Přijímače žádostí – Broadcast Recievers

Přijímač je komponenta, která odpovídá na systémem zasílané oznámení, pokud jsou mu určena. Oznámení zasílané systémem mohou být typu – vypnul se display, úroveň baterie je nízká, byl vyfocen obrázek, došla nová SMS. Oznámení ale mohou být vysílána i aplikacemi – byla stažena nějaká data, která další aplikace mohou použít.

Přestože přijímač nemá uživatelské rozhraní, může vytvořit upozornění do stavové lišty pro upozornění na vznik nějaké události. Obecně se jedná o „vstupní bránu“ do dalších komponent a měl by dělat co nejméně práce, jako spustit službu pro vykonání nějaké akce v závislosti na události.

## 3.5. Historie verzí

Tabulka verzí (1.) je zde uvedena pro úplnost (detailně na [2], [3] a [4]). Již existuje 7 hlavních verzí OS Android, které rozšiřují funkcionalitu a opravují chyby. Kromě verze Honeycomb jsou všechny pojmenované po zákuscích.

Verze OS	Verze API	Vydána	% zastoupení
1.5 (Cupcake)	3	30. dubna 2009	0,3%
1.6 (Donut)	4	15. září 2009	0,7%
2.0/2.1 (Eclair)	7	26. října 2009	5,5%
2.2 (Froyo)	8	20. května 2010	20,9%
2.3/2.4 (Gingerbread)	9-10	6. prosince 2010	64,4%
3.0/3.1/3.2 (Honeycomb))	11 – 13	22. února 2011	3,3%
4.0 (Ice Cream Sandwich)	14 – 15	19. října 2011	4,9%

Tabulka 1. Tabulka verzí OS Android s procentuálním zastoupením.

Zdroj: [2], [3] a [4]

## 4. Vývojářské prostředky

V této kapitole zkusím nastítnit několik možností vývoje aplikací pro Android. Určitě se nejedná o kompletní seznam možností, nebo detailní popis. Jsou to spíše pro mě (naší firmu) něčím zajímavá řešení. Zajímavá tím způsobem, že se napíše kód jednou, ale publikuje se na více platform/typů zařízení, nebo „znovupoužitelnost“ kódu, respektive jeho vnitřních částí (netýká se GUI), popřípadě možnost zpravování a vylepšování aplikace bez nutnosti publikování updatů. Vycházím pouze ze svých zkušeností, kterými jsem prošel a vyzkoušel, nebo o nich aspoň v nějaké míře uvažoval.

### 4.1. Android SDK

Android Software Development Kit (SDK) je sada nástrojů nezbytná pro vývoj aplikací pro Android [11]. A to ať už zvolíte téměř jakékoliv vývojové prostředí (výjimkou je Flex – kap. 4.3. a HTML5 – kap. 4.5.). SDK je dostupný pro všechny hlavní platformy operačních systémů a dělený na tři části (základní, doporučená a plná konfigurace). K udržení přehledu o instalovaných balíčcích a nástrojích je zde SDK Manager (na obrázku 4.) a pro verze emulátorů AVD Manager.

Základní konfigurace obsahuje nezbytné nástroje pro vývoj. *SDK Tools* (detailně popsané na [9]), kde jsou nástroje pro debugging (DDMS), správu Android Virtual Devices (AVD), Android Emulátor, analýza grafického layoutu a další.

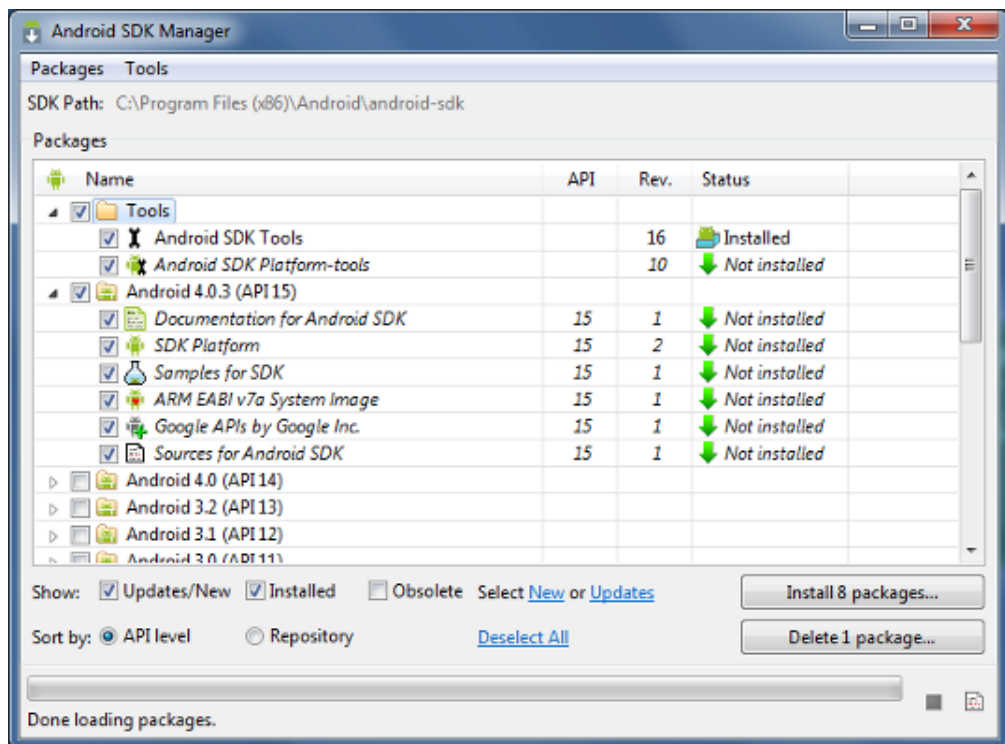
*SDK Platform-tool* obsahuje nástroje, které jsou závislé na verzi platformy a jsou aktualizovány při nové verzi SDK (třeba Android Debug Bridge – umožňuje nahrávat soubory do zařízení).

*Android SDK platforms* obsahuje verze (platformy) OS Android. Každá verze se skládá z knihoven, systémového obrazu, či skinů emulátoru. Ke kompilaci aplikace a spuštění emulátoru je potřeba alespoň jedna verze systému.

Doporučená konfigurace obsahuje *USB Driver*, který je potřeba pro Windows a pouze v případě, kdy vyvíjíte vůči reálnému zařízení. Dále zde najdeme *příklady kódů*, aktuální pro každou verzi, nebo *dokumentaci*, která je lokální kopie pro danou verzi a je využívána v prostředí Eclipse.

V plné konfiguraci pak nalezneme *Google API*, zpřístupňující rozhraní Google Maps, které je pak možno využít v aplikacích. Nebo další *SDK platformy*, což jsou doplňující balíčky třeba pro ověřování licencí aplikace, zda se nejedná o nelegální kopii (Market Licensing Package).

Jak jsem již psal, SDK obsahuje Emulátor OS Android. Umožňuje testovat aplikace vůči rozdílným konfiguracím hardware, verzím systému a dalším nastavením bez nutnosti vlastnit fyzické zařízení. Pomocí AVD Manageru si pohodlně nastavíte požadovanou konfiguraci vašeho emulátoru jako síťové připojení, SD kartu, velikost RAM, dotykový displej, ... Většina aplikací se v emulátoru chová



Obrázek 4. Android SDK manager zobrazující dostupné balíčky pro jednotlivé verze platformem.

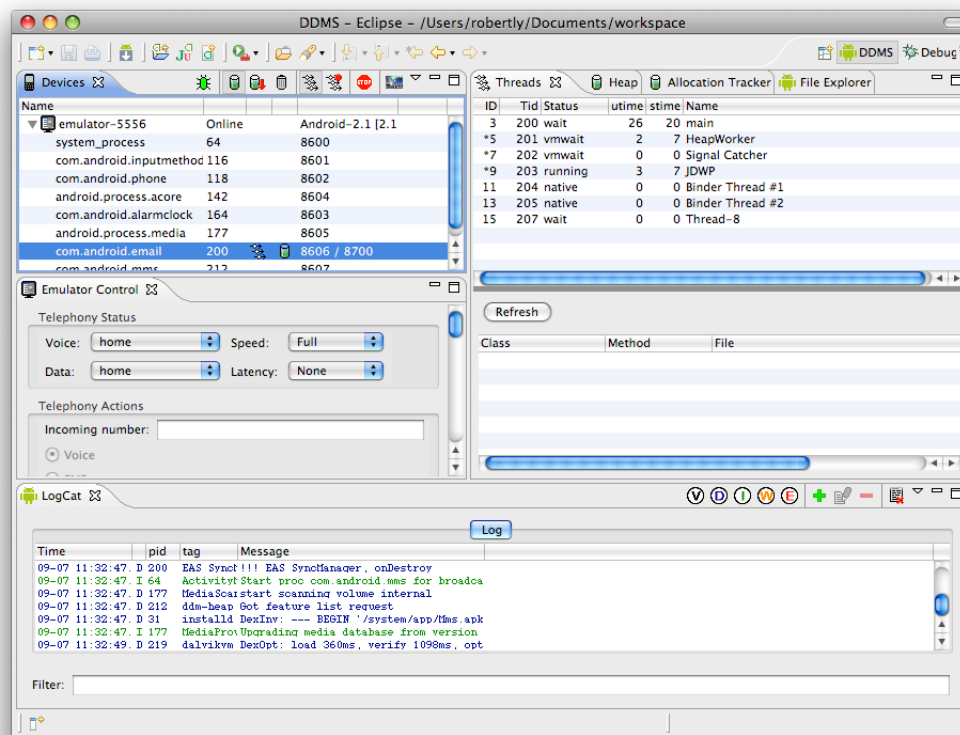
Zdroj: <http://developer.android.com/index.html>

jako na fyzickém zařízení, existují však výjimky, které se dají virtualizovat těžce nebo vůbec. Jedná se o přijímání hovorů, sms, práce s bluetooth, práce se senzory, nebo určení pozice zařízení pomocí GPS. Pro většinu zmíněných tu je nástroj DDMS (obrázek 5.), který obsahuje prvek Emulator Control.

## 4.2. Java (Eclipse)

Oficiálně podporované vývojové prostředí pro Android aplikace je Eclipse. Do něj je možné nainstalovat ADT plugin (nutná verze Eclipse 3.6.2–Helios a vyšší), který ulehčuje práci s Android projektem. Jedná se o nejpoužívanější vývojové prostředí pro Javu, už z toho důvodu, že se jedná o open source project a je variabilní díky možným rozšířením v podobě nejrůznějších pluginů.

Toto vývojové prostředí ovšem není nutností. Je možné, a plně funkční, používat jiné vývojové prostředí nebo jednoduchého textového editoru a kompilovat aplikaci pomocí příkazové řádky díky Android SDK.



Obrázek 5. Nástroj DDMS.

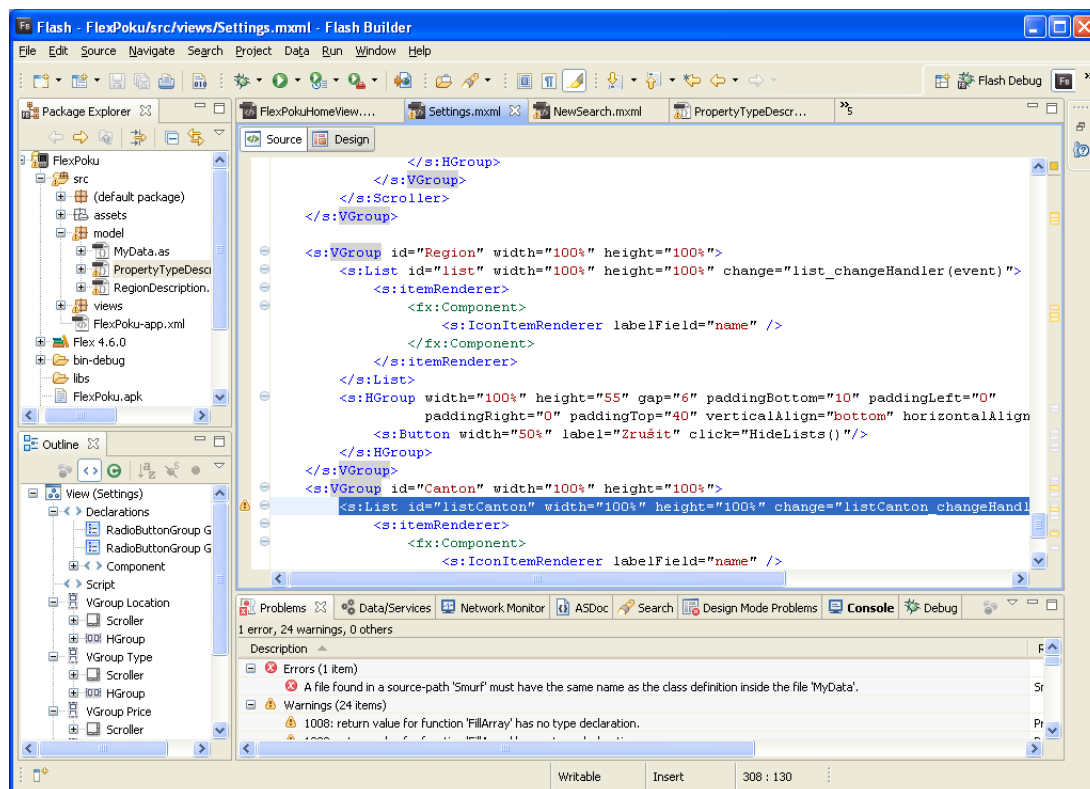
Zdroj: <http://developer.android.com/index.html>

### 4.3. Adobe (Flex)

Vytváření mobilních aplikací ve Flexu může být hodně zajímavá volba, už jen z důvodu jeho multiplatformnosti. Umožňuje tvořit aplikace pro Android, iOS a BlackBerry Tablet OS, stejně jako pro webové prohlížeče nebo desktopy ([12], [13] nebo [14]). To vše při použití stejného programovacího modelu, nástrojů a kódu. Flex je open source application framework pro vývoj aplikací. Tvoří sadu komponent a tříd spolu s kompilátory a dalšími command-line nástroji nazývaný Flex SDK. Jedná se o vytváření multiplatformních RIA (Rich Internet Application) aplikací. Aplikace pro svůj běh potřebují běhové prostředí (runtime). V současné době jsou dvě, pro běh ve webových prohlížečích se jedná o Flash Player a pro instalaci a běh na desktopu, či mobilních zařízeních, využívá Adobe AIR.

Jako placené vývojové prostředí nad Flexem lze použít Flash Builder, který značně ulehčuje tvorbu projektů ve Flexu. Vytvoří adresářovou strukturu, pomáhá s tvorbou GUI, má automatické doplňování při psaní kódu (IntelliSense u Microsoft VS), ušetří spoustu času u automaticky generovaného kódu po „projetí wizarda“ na připojení k databázi, či webové službě. Má vestavěný debugger,

umožňující sledování využití sítě, zátěže procesoru a pomocí Adobe AIR SDK lze spustit i simulátor pro běh aplikace. Tudíž není nutná instalace Android SDK, JDK a dalších knihoven, tedy pokud nechcete kontrolu na Emulátoru vůči konkrétní verzi Androidu.

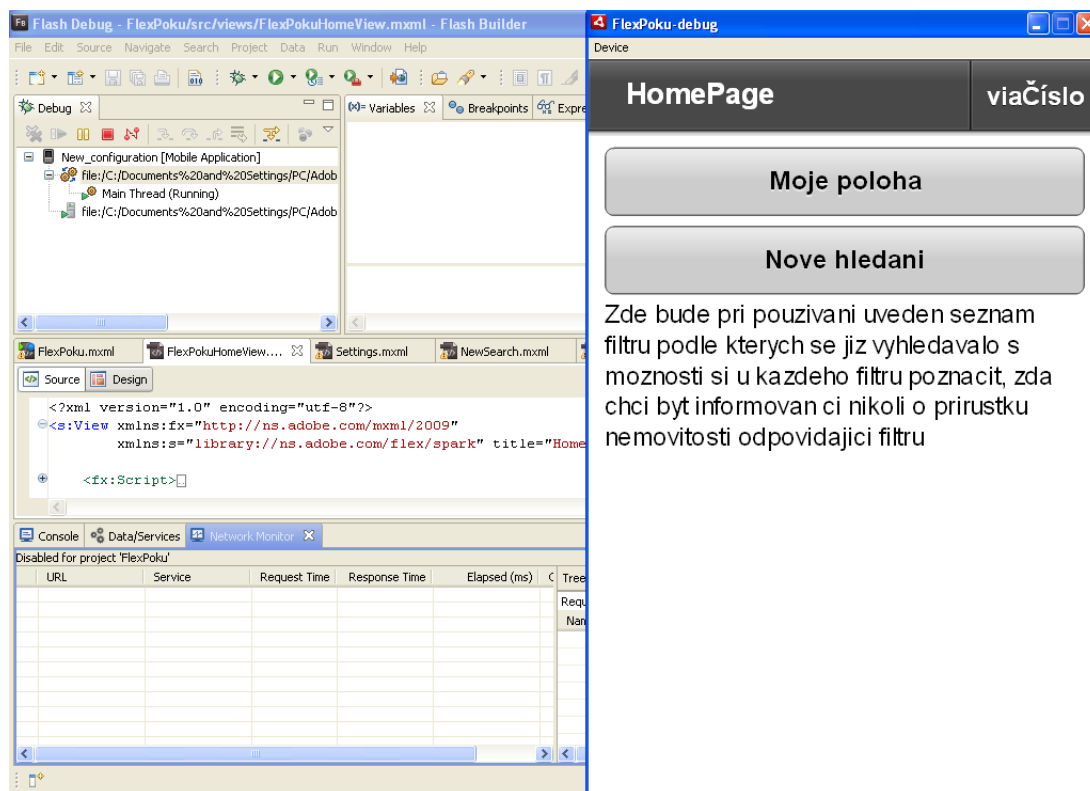


Obrázek 6. Vývojové prostředí Flash Builder - kódová část.

Zdrojový kód pak obsahuje vzhled uživatelského rozhraní, tj. typy komponent, jejich velikost a umístění na scéně, vlastnosti a funkce, podle kterých se kompiluje výsledný .swf soubor. Pro tento obsah Adobe zvolilo formát XML, respektive MXML. Má silnou sémantiku pro tvorbu uživatelského rozhraní. V kódu jsou vidět elementy jako „HBox, Label, Button, ...“. Mezi tagy *script* píšeme procedurální kód v jazyce ActionScript, který je velmi podobný javascriptu.

Ve Flexu a Flash Builderu verze 4.5 jsem si zkusil napsat prototyp aplikace pro bakalářskou práci. Práce byla poměrně rychlá a intuitivní, dokonce i napojení na webové služby bylo jednoduché a funkční, ale jednalo se pouze o prototyp bez složitých konstrukcí.

Svojí relativní jednoduchostí, se spoustou předpřipravených komponent a wizardů mě Flex / Flash Builder příjemně překvapil, jakožto i funkčností na reálném



Obrázek 7. Flash Builder - debug a simulátor.

zařízení. Přestože podpora mobilních zařízení je do Flexu / Flash Builderu přidána pouze krátce, myslím si, že tato vývojová platforma má velký potenciál, obzvláště pak pro vývojáře znající HTML/javascript, nebo obeznámené s ActionScriptem.

Pro spuštění na reálném zařízení je potřeba instalace běhového prostředí Adobe AIR, které se nechá „přibalit“ do výsledného instalačního balíčku, čímž velikost vzroste asi o 10MB před a 30MB po instalaci. Nebo se použije čistý instalační balíček a v případě, že na zařízení není Adobe AIR nainstalován je uživatel při spuštění aplikace vyzván k jeho stažení a instalaci. Druhá varianta má výhodu v případě, že máte více aplikací využívající Adobe AIR. Nevýhodou v obou případech je přítomnost kompletního běhového prostředí.

I když věčný boj s místem na našich paměťových médiích řeší spousta z nás, obzvláště pak na zařízeních levnějšího charakteru, dříve nebo později většinu z nás nějakých nadbytečných 30MB nebude dráždit. Ale přesto i kvůli stahování těchto aplikací přes internet bych do budoucna uvítal možnost přibalení jen poměrové části knihoven a běhového prostředí, které používá konkrétní aplikace (jako třeba u Mono for Android [18]).

## 4.4. Xamarin (Mono for Android)

Společnost Xamarin (<http://xamarin.com/>) byla založena na jaře 2011, aby převzala správu open source projektu Mono. V současnosti nabízí i komerční nástroje pro vývoj mobilních aplikací pro systémy iOS (MonoTouch) a Android (Mono for Android) [17].

Zatím co instalace Mono for Android je možná jak pro Mac OSX i Windows, MonoTouch jde nainstalovat pouze pod Mac OSX. Kódovou základnu vlastních „knihoven“ můžete přes drobné odchylky držet stejnou pro vývoj mobilních aplikací pod Windows Phone, iOS i Android. Toto se samozřejmě netýká uživatelského rozhraní, které je pro každou platformu jiné. Liší se také v ukládání dat do úložiště, či spolupráce s databázemi.

K vývoji aplikací následně stačí použít zmíněné Mono nebo Visual Studio a doinstalovat jeden ze systémů MonoTouch SDK nebo Mono for Android SDK, který se kompletně začlení do vývojového prostředí. Přibude možnost vytvářet projekty pro Android. Pomůže s organizací adresářové struktury, debugováním, stejně jako s automatickým doplňováním psaného kódu.

Zkompilováním vzniká instalační .apk balíček. Pokud je vývoj v „Debug módu“ potřebuje aplikace ke svému běhu runtime (v tomto případě kompletní), který se do vývojového zařízení, či emulátoru nainstaluje automaticky. Toto opatření je kvůli rychlosti, kdy není třeba toto běhové prostředí neustále kopírovat s aplikací do zařízení. Druhá varianta kdy je vývoj v „Release módu“ nastupuje pomalejší varianta tvorby instalačního .apk balíčku. Běhové prostředí je přímo přibaleno k výsledku, ale obsahuje pouze části nezbytně nutné pro běh aplikace.

Pro vývoj v Mono for Android je nezbytností instalace i Android SDK, JDK 6, automatický instalátor ještě doinstaluje GTK#, MonoDevelop a vše řádně zkontroluje.

O zkušenosti z vývoje v tomto prostředí, které jsem si nakonec zvolil, se podělím v kapitole 5.

## 4.5. HTML5

Další poměrně zajímavou „crossplatform“ možností je využití HTML5, CSS3 a javascriptu na straně klienta. Přestože HTML5 ještě není ani zdaleka standardizován, jeho vývoj je překotný, podpora stále vzrůstající a možnosti přinejmenším rozsáhlé. Možnost offline aplikace s přístupem na vlastní úložiště pro data, s podporou přístupu k hardwaru zařízení, s možnostmi vzhledu podobného nativním aplikacím je velice lákavá.

Plus přichází i s možností publikování jedné pevné verze pro uživatele někde na markety (AppStore, GooglePlay, ...) pomocí „wrapperů“ (o nich později), a protože se jedná o web aplikaci, bude se aktualizovat při každém online běhu. Což umožňuje podstatně větší pružnost updatů, nebo patchů. Na druhou stranu,

jedná se o „front-end“ část, navíc interpretovanou v prohlížeči, tudíž kód je pro všechny zájemce dobře viditelný.

Přesto jsou tu ale. Vzhledem k tomu, že se jedná o webovou stránku s touto aplikací, těžko ji najdete mezi nainstalovanými ve vašem zařízení. Ano u některých systémů lze na plochu připnout i odkaz na webovou stránku, ale to není to samé. A co se přístupnosti k hardwaru týče, není slavná. Ovšem na tyto problémy jsou zde řešení v podobě projektů jako *PhoneGap*, *Titanium* nebo *AppGyver*, *Rhodes*, *Appcelerator* za využití třeba SmartClient. Ty danou aplikaci obalí, zpřístupní hardware zařízení. Z webové aplikace je nativní i se všemi jejími výhodami s využitím komponenty „WebView“.

Je pravda, že využití bude spíše pro určitou skupinu aplikací a pro specifická řešení, a to pouze za předpokladu, že vás bude bavit neustálé odlaďování kódu pro různé prohlížeče a platformy, které mají větší či menší podporu třeba v interpretaci „gest“. Nemluvě o podpoře jednotlivých prohlížečů pro prvky HTML5 a CSS stylů.



## 5. Vývojové prostředí, instalace a zkušenosti

Dlouho jsme zvažovali platformu Flex od Adobe nebo cestu HTML5. Flex jsme nakonec vyškrtli z důvodů velkého runtime a nutnosti učit se komplet nový styl programování pro Flex. Runtime sice šel odříznout a separovat a dnes už je použitelných aplikací napsaných ve Flexu povícero, ale v době rozhodování jsem jich moc nenašel, takže i nutnost držet nainstalované dosti velké Adobe AIR kvůli jedné aplikaci mi přišlo trochu zvláštní.

Co se týče vývoje v HTML5 a javascriptu, tato cesta ještě není zavržena, spíše odsunuta. Podle mě, by mohlo jít o snadnější propagaci, udržitelnost, obzvláště pak při aplikacích pro jednoduchou prezentaci dat.

Jak už z předešlého textu vyplývá, nakonec jsme se jako firma pro finální řešení aplikace rozhodli jít cestou Visual Studio 2010 .NET (VS) a jazyka C# za využití nástroje Mono for Android (MfA) verze 4.0.6.

Především se jednalo o známé vývojové prostředí a jazyk ve kterém jsem napsal nejvíce kódu. Jsou mi známé používané konstrukce (i když při tvorbě aplikací pro android se používá spousta jiných). Dále zde byla zajímavá potenciální možnost využití již napsaného kódu pro jiné platformy (ukáže až čas). A jako firma zůstaneme věrní naším vývojovým nástrojům.

### 5.1. Příprava a instalace

Xamarin na svých stránkách má možnost stažení trial (časově neomezené) verze pro MfA. Je zde však omezená, respektive uzamčená, funkčnost. Je možné vyvíjet jen v „Debug módu“ a navíc pouze vůči emulátoru. Ale na vyzkoušení a získání představy to stačí. Po vyplnění registrace si stáhnete velikostně malý instalátor, který si po spuštění stáhne dodatečné informace a prostředky nezbytné pro kontrolu vašeho systému.

Následně vše už probíhá automaticky. Instalátor vám oznámí, co všechno nemáte a potřebujete (jedná se asi o 5 hlavních komponent). Není zde jakákoliv volba úpravy instalace. Po potvrzení všech licenčních ujednání probíhá instalace pěkně popořadě na instalátorem určené místo. Nejdříve stažení příslušné komponenty z internetu a její následná instalace a pak další. . . Díky stahování z internetu je instalace trochu delší, ale zato plně automatická.

Po nainstalování asi 3GB dat jsem plný očekávání spustil Visual Studio a vytvořil první projekt pro Android. V nabídce je několik druhů „templatů“ podle toho, zda se jedná o knihovnu, aplikaci, xml nebo tvorbu „layoutů“ (xml s automatickým doplňováním pro tvorbu GUI). Vše se tváří že běží, základní adresářová struktura je vygenerována, automatické doplňování kódu funguje.

Zkusím přeložit a ejhle. Nenalezeny komponenty Android SDK. Kontrola nastavení, instalace, adresářů, vše tak jak má být. Teprve až z hlubšího zkoumání logů vyplývá, že vnitřní odkazování nezvládlo českou lokalizaci systému XP, kde



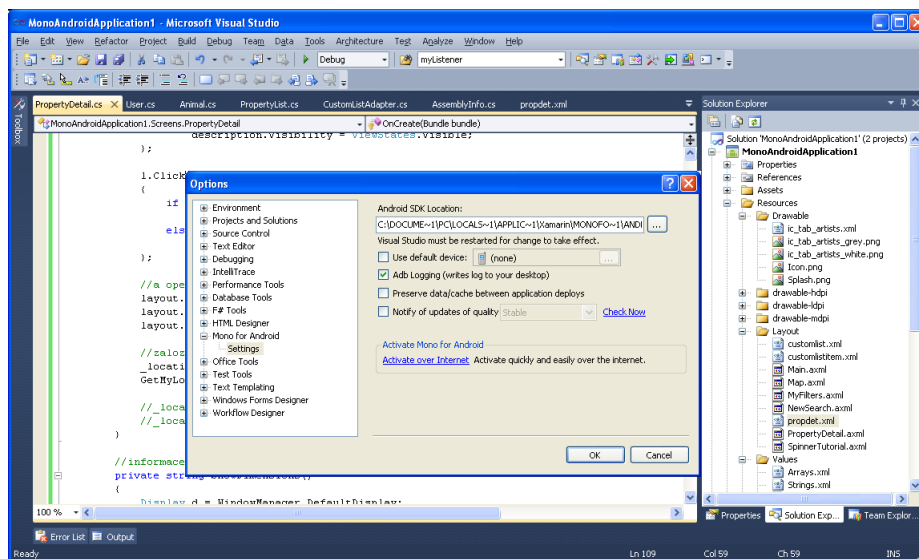
Obrázek 8. Instalace Mono for Android.  
Zdroj: <http://developer.android.com/index.html>

nějakou prapodivností jsou přeloženy do češtiny i systémové adresáře. Zkoušení přeinstalace, přesouvání a přepisování registrů ani změna systémových cest se mi nepovedla. Poslední zmíněné by mělo pomoci. Mě pomohla až anglická verze systému, kde už vše běželo jak mělo.

Další nemilé překvapení nastalo při vytváření Virtuálních verzí systémů Android pro Emulátor. Každá z nich má kolem 500MB a 5 z nich je před instalovanými. Tudíž mi velice rychle došel prostor na systémovém disku. Nepomohla ani přeinstalace Android SDK, kterou jsem sice přesunul na jiný disk, ale zmíněné Virtuální systémy pro Emulátor nikoli. Teprve až po čase stráveném hledáním jsem našel řešení, kdy se nástroje Android SDK, potažmo i MfA, odvolávají jen na konfigurační soubory na systémovém disku. Pokud v nich změníme přístupové cesty, můžeme Virtuální systémy přesunout kamkoli chceme.

## 5.2. Zkušenosti s vývojem, emulátorem, GUI a fyzickými zařízeními

Po úvodních drobnostech při zprovoznování vše funguje tak jak má. Založil jsem si pár virtuálních strojů pro Emulátor s různými verzemi a hardwarovými nastaveními (velikost a rozlišení displeje, velikost RAM, dotykový displej nebo podpora GoogleMap)



Obrázek 9. Visual Studio s nastavením Mono for Android.

### 5.2.1. Emulátor

Rychlost Emulátoru není zázračná a při představě vývoje jen vůči emulátoru, mě jímala hrůza. Existuje balíček obsažený ve verzi 2.3.3 v SDK manageru, který emulátor o něco zrychluje pro běh na x86 procesorech [19]. Emulátor občas nenaběhne, ale opravdu nejdéle mu to trvá při prvním spuštění. Další už jsou o dost rychlejší, ale stejně se nedoporučuje ho vypínat během vývoje (strávili by jste život čekáním).

Dubug ve Visual Studiu pro aplikaci běžící v Emulátoru funguje obstojně (ne vždy jsou vyhodnocené všechny lokální proměnné, obsah textových proměnných je zkracován, ...). Problémem je opět rychlost. Než se debug jen spustí, je to málem na čaj, nemluvě o skákání po příkazech, kdy přemýšlíte, zda už to spadlo či nikoli. Nicméně je to funkční a na odladění chyb, které nezachytí kompilátor – nepostradatelné.

Už z principu podstaty je Emulátor nepostradatelný. Jelikož Android běží již v několika verzích na spoustě zařízeních je bláhové si pořizovat více fyzických testovacích zařízení, už jen z toho důvodu, že u dvou různých zařízení se stejným systémem se každé může chovat jinak. Proto si vyberte nejnižší zamýšlenou podporovanou verzi, nasimulujte si několik typů zařízení a doufejte, že většině zařízení spadajících do této množiny vaše aplikace bude fungovat (pokud jste někde na něco nezapomněli).

### 5.2.2. Tvorba GUI

Nutno podotknout, že GUI pro android lze definovat čistě programově, v ex-

terním .xml souboru nebo kombinací obou – přifukováním (Inflate). Je dobrým zvykem používat xml verzi, obzvláště pro layouty, kde se nepotřebujete na jeho jednotlivé prvky odvolávat za běhu (měnit hodnotu, barvu, obrázek, ...). Je to rychlejší, přehlednější, ale občas se špatně hledají chyby. Přestože MfA ve svých .axml (normální xml soubor, ale pro popis layoutu) podporuje formu automatického doplňování, nefunguje u všech hodnot a s nápovědou se většinou odkazuje na Android. A tady nastává kámen úrazu, když tady přehodíte typ layoutu (LinearLayout na TableLayout), ale v kódu kde se na něj odkazujete, zapomenete přepsat tento typ, kompilace proběhne v pořádku, ale aplikace už padá.

Tvorba GUI pro tvorbu android aplikace je kapitola sama pro sebe [20]. V Eclipse existuje designer který vám stvoří výsledný xml kód z daných komponent, které jste si po přetahovali na plochu a po nastavovali. Někomu vyhovuje, jiný je nespokojený. Pro MfA žádný takový neexistuje. Našel jsem projekt <http://droiddraw.org/>, který má obdobnou funkcionalitu jako designer v Eclipse. V mém případě sloužil k několika testům jak skládání layoutů funguje. Po té už stejně většinou tvoříte přímo, ale občas stojí hodně času a úsilí, aby vzhled vypadal tak jak jste si vymysleli. U některých složitějších layoutů je to docela oříšek.

### 5.2.3. Připojení fyzických zařízení

Pro připojení fyzického zařízení potřebujete (kromě onoho zařízení) jeho usb ovladač. Univerzální je dodáván přímo k Android SDK. Nevím proč, ale mě nepomohl. Nicméně celkem dobře fungují programy dodávané na synchronizaci konkrétního zařízení s počítačem, které tyto drivery také instalují (v mém případě Samsung Kies, nebo HTC Sync). Samsung běžel bez problémů, HTC jsem občas musel přemlouvát.

Připojením zařízení k počítači se zbavíte závislosti na Emulátoru. Vaše testování se mnohonásobně zrychlí, což se týče i samotného debugování. Jo to opravdu příjemná změna a navíc působí i psychologicky, kdy už vidíte výsledky své práce.

## 6. Aplikace vyhledávání nemovitostí

Aplikace vyhledávání nemovitostí je cílená na široké spektrum uživatelů chytrých mobilních telefonů na platformě Android. Slouží jako další prvek přiblížení uživatelů (potencionálních zákazníků) našim klientům, pro něž zprostředkováváme nabídku.

Z toho důvodu jsem se snažil o co nejpřímější a nejjednodušší používání aplikace. Ta je primárně určena k vyhledávání podle filtrů, které musejí jít opětovně použít (nezadávat pořád to samé). Dále k zobrazení detailu nemovitosti s co nejvíce informacemi, které jsou dostupné. A následně poskytnout kontaktní informace na makléře, nebo realitní kancelář. Samozřejmostí je historie navštívených nemovitostí, nebo seznam oblíbených.

Zároveň se snažím uživatelům zpřístupnit co nejširší základnu dat, jež by je mohla zajímat. Tudíž věci jako vyhledávání vzhledem k mé pozici, vyhledávání realitních kanceláří a zjišťování počet inzerovaných nemovitostí, zobrazování podobných nemovitostí u jejich detailu, nebo prohledávání nemovitostí od konkrétní realitní kanceláře, jež aplikace umožňuje, jsou jen příjemný bonus.

Z přenášných dat jsou asi nejobjemnější data obrázků. Ta nejsou z časových důvodů přenášena přímo, ale dočítána asynchronně až když už jsou výsledky zobrazeny uživateli. Když si vezmeme scénář kdy budu hlídat konkrétní filtr každý den, tudíž se mi výsledky moc nemění, jsou převážně zobrazeny záznamy, které jsem již viděl. A proto, aby se tato data obrázků nemusela opětovně stahovat přes internet znovu a znovu (při změně filtru, či řazení), jsou jednou stažené obrázky uloženy na lokální úložiště a odkazy na ně „nakešované“. Nejméně používané odkazy jsou postupně z paměti odstraněny, aby nedošlo k vyčerpání uživatelské paměti.

Uživatelský popis aplikace, který je realizován formou scénářů užití, následuje v příloze [A.](#), programátorský popis pak v příloze [B.](#)

### 6.1. Omezení oblastí funkcí aplikace

Jelikož nejsme realitní server veřejný, týkají se nás jistá omezení ze stran našich klientů, vyplývajících ze smluv či poskytovaných dat. V poskytování detailních informací, jež by se daly zužitkovat ve funkcionalitě aplikace, jsme omezeni například nekompletní adresou (velice často je zadána pouze obec, nikoli ulice). Nemůžeme využít oblast funkcí pracující s přesnou polohou (i proto si aplikace bohatě vystačí s určováním orientační polohy ze sítí wifi a mobilních operátorů). Tudíž funkce pro zobrazování přesné polohy nemovitosti nebo celých seznamů do mapy je nepoužitelná, protože spousta nemovitostí by byla zobrazena na jednom místě – uprostřed obce. Nelze upozorňovat ani na přítomnost vybraného druhu nemovitosti ve vašem okolí, když procházíte město.

Další možná oblast funkčnosti, která je nám z technických důvodů bohužel zapovězena, je hlídání konkrétních nemovitostí nebo celých filtrů a následné upozornění uživatele na změnu. Ze zjištěných důvodů jsou nám stejná (pozměněná) data podsouvaná s jinými ID v pravidelných časových intervalech, tudíž se špatně rozhoduje zda se jedná o změnu již existujících záznamů (navíc ty byly staženy jako neplatné), nebo se jedná o záznamy nové. Neděje se tak vždy a všude. Ale uznejte, kdyby vám každý den chodily jako odpověď na filtr znovu a znovu, byť i dvě, ty samé nemovitosti. Určitě by jste takovou funkci přestali používat. To samé je u konkrétních nemovitostí, kde uživatelům s lítostí oznámíme, že danou nemovitost jim realitní kancelář stáhla, a oni ji náhle znovu najdou aktivní s nezměněnými parametry.

Než a jestli dojde ke změně výše uvedených důvodů, nemůže aplikace bohužel tyto oblasti funkcí, jakkoliv užitečných, podporovat.

## 6.2. Konkurence a existující řešení

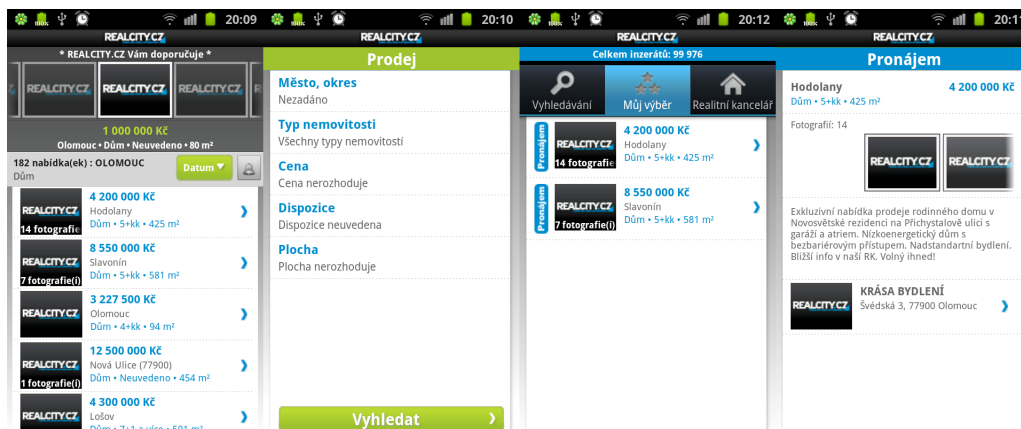
Konkurenční aplikace, pokud se dají nazvat konkurenční, jsou v Česku zatím asi tři (za nimiž stojí realitní portály) a jedna (neoficiální), co se mi povedlo najít, a nějaké zahraniční, které v našich podmínkách většinou nechodí příliš dobře.

Konkurenční jsou hlavně portály, tyto aplikace bych nazval službami potenciálním zákazníkům, protože každá tato aplikace si běží nad svojí množinou dat konkrétního portálu z které prezentuje výsledky. Tudíž pak záleží nad kterým portálem chci hledat. Pravda v hodně případech se tyto množiny částečně překrývají.

Co se aplikací týče, mají všechny pochopitelně stejné rysy a podobnou funkčnost, aspoň tu základní. Jedná se o zadání filtru, seznam nalezených, detail nemovitosti a kontakt na makléře, či realitní kancelář. Většinou bývá přítomna i historie a uložené filtry. V tabulce 2. jsem se snažil shrnout rozšiřující funkčnost. Tabulka neporovnává veškerou funkčnost aplikací, ale pouze její vybranou část z řad rozšiřujících funkcí.

**RealCity.cz** – jejich aplikace se mi z technologického hlediska líbila asi nejvíce. Spousta funkcí (oblíbené, hlídání, používání menu). Bohužel veškeré uložené věci jako oblíbené, filtry nebo realitní kanceláře mi po ukončení aplikace zmizí (myslím tím, když je aplikace odebrána z paměti, ne pouze skryta). Takže ani ono hlídání jsem nevyzkoušel. Neobsahuje náhled nemovitosti na mapě ani historii jejich zobrazení. (Obrázek č. 10.)

**HyperReality.cz** – aplikaci nelze spustit bez připojení k internetu. Hned po startu ukazují nemovitosti v okolí, tedy pokud máte zapnutou GPS, jinak nemovitosti z Prahy. Mají historii zobrazených nemovitostí, stejně jako si pamatují zadané filtry. Historie filtrů nelze mazat, jednou zadaný filtr neupravíte, musíte použít nový, navíc se ukládají všechny filtry znova (i když



Obrázek 10. RealCity - seznam nemovitostí, nový filtr, oblíbené, detail nemovitosti. (Poslední dobou mi aplikace přestala zobrazovat fotografie – možná potřeba aktualizace)



Obrázek 11. HyperReality - seznam nemovitostí, nový filtr, historie, detail nemovitosti.

už v seznamu existují). Kontaktní údaje si člověk musí napsat na papír, aby je mohl použít. Ukládání do oblíbených aplikace nenabízí. Jednoduchý funkční základ s drobnými omezeními. (Obrázek č. 11.)

**Reality.cz** – aplikace velice podobná té předchozí. Fungují nemovitosti v okolí i bez nutnosti zapínání GPS (jen síť). Zde už nelze mazat ani historie navštívených. Zato umožňuje volat a psát emaily z kontaktních údajů po stisku tlačítka. Má jednoduchý elegantní a funkční design. (Obrázek č. 12.)

**dmn:sreality.cz** – aplikace nad portálem sReality (nejsou pod ní podepsaní). Umožňují bohatý výběr nastavení hledání (pro mě trochu nepřehledný). Umožňují ukládání každého vytvořeného filtru (ne automaticky). Nemají nemovitosti v okolí, stejně jako historii, nebo možnost ukládání do oblíbených. Obsahuje spoustu nastavení, líbivých animací a souběžných obrázků. Designově velice zdařilá. (Obrázek č. 13.)

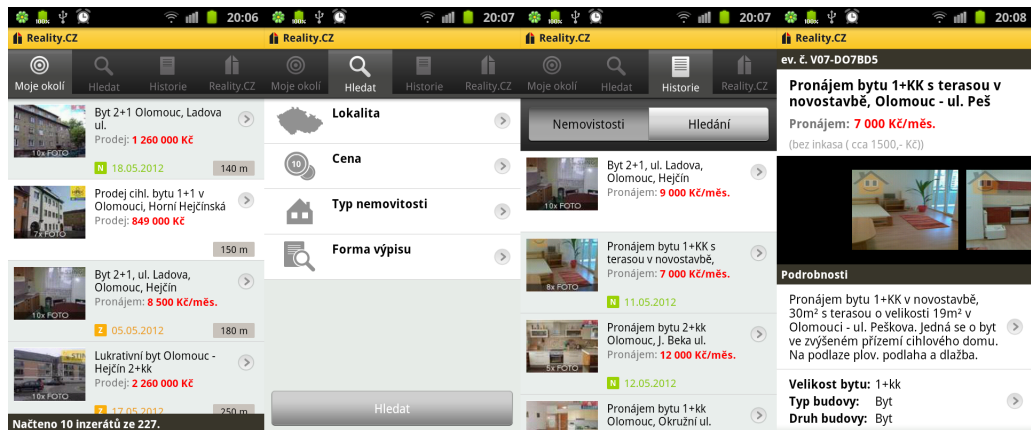
FUNKČNOST	1	2	3	4	5
Načítání něčeho po startu		X	X		
Nabízení top nemovitostí	X				
Úprava filtru ze seznamu	X			X	
Historie filtrů	X*	X	X	X	X*
Úprava filtru z historie	X		X	X	X
Historie navštívených		X	X	X	
Oblíbené	X*			X	
Mazání historie		X		X	
Mazání filtrů	X			X	X
Podobné nemovitosti				X	
Vyhledávání RK	X			X	
Vyhledávání v jedné RK	X			X	
Mapa		X	X	X	X

Tabulka 2. Tabulka porovnávající vybranou funkčnost 5 aplikací.

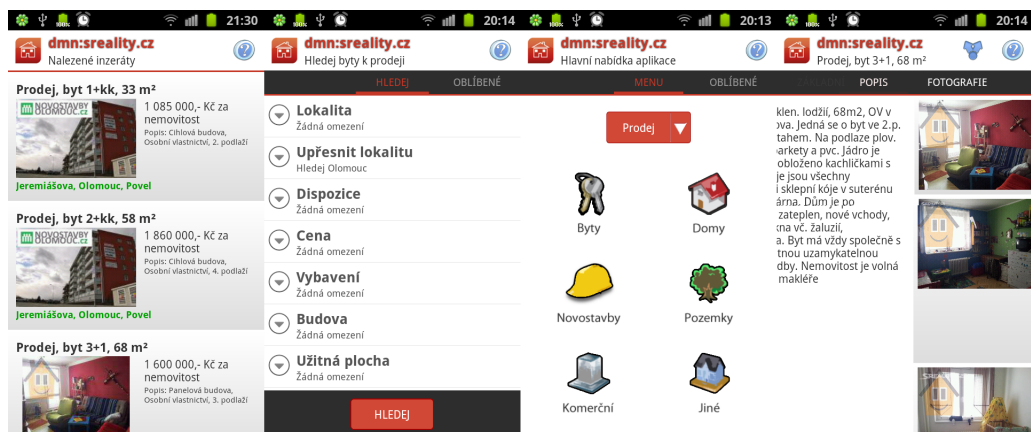
Zleva: *RealCity*, *HyperReality.cz*, *Reality.cz*, *viaReality*, *dmn:sreality.cz*

\*: *RealCity* -- po zavření aplikace mizí; *dmn:sreality.cz* -- musí se ukládat





Obrázek 12. Reality - seznam nemovitostí, nový filtr, historie, detail nemovitosti.



Obrázek 13. dmn:sReality - seznam nemovitostí, nový filtr, hlavní obrazovka, detail nemovitosti – přelom mezi popisem a galerií (horizontální posouvání).

## Závěr

Cílem této bakalářské práce bylo prozkoumat platformu Android, zjistit pro ni možnosti vývoje aplikací a následně navrhnout a realizovat aplikaci pro vyhledávání realit.

Aplikace samotná má splňovat standardy uživatelského rozhraní pro Android aplikace, spolu s přehledností a jednoduchostí ovládání. Uživatel má mít možnost využívat hledání ve svém okolí (za použití gps), jakožto i zadávání filtrů na vyhledání konkrétních typů nemovitosti třeba podle místa nebo ceny. Zadané filtry mají jít znovu jednoduše aktivovat při opětovném použití aplikace. Všechny navštívené nemovitosti se ukládají do historie, která je realizována jako fronta omezena počtem 50 posledních. Každou navštívenou nemovitost lze označit jako oblíbenou, čímž se stane přístupná i offline, včetně obrázků a informací k dané nemovitosti. Pokud jsou informace uvedeny, obsahuje nemovitost adresu, cenu, textový popis, tabulkový výčet nejdůležitějších faktů a informace o makléři a realitní kanceláři prodávající tuto nemovitost s možností jejich kontaktování.

V teoretické části bakalářské práce jsem se věnoval platformě Android, její historii, architektuře a základním komponentám, které využívají aplikace nad touto platformou.

Další část jsem věnoval vývojovým nástrojům použitelných pro vývoj aplikací. Nejedná se o ucelený výčet nebo detailní popis všech dostupných nástrojů, jsou zde uvedené nástroje se zajímavým potenciálem pro vývoj na různé platformy mobilních zařízení.

Do budoucího života aplikace se plánuje rozšíření aplikace o prvky hlídání a upozorňování na změny nemovitostí v rámci uloženého filtru, nebo konkrétní nemovitosti. Propojení aplikace na užší bázi s webem viaReality pro uživatele, tak aby měly přístupné oblíbené nemovitosti jak v mobilu, tak na webu. Vylepšení některých vnitřních nastavení plus uživatelské nastavení aplikace. Nebo zlepšit práci nad mapami. Rádi by jsme zpřístupnili funkčnost, kdy po načtení „qr kódu“ z katalogu, bude uživatel přesměrován na konkrétní detail nemovitosti v aplikaci (popř. po obdržení odkazu mailem).

## Reference

- [1] Wikipedia (en) – *Mobilní operační systémy*  
([http://en.wikipedia.org/wiki/Mobile\\_operating\\_system](http://en.wikipedia.org/wiki/Mobile_operating_system))
- [2] Wikipedia (cz) – *Android - operační systém*  
([http://cs.wikipedia.org/wiki/Android\\_\(operační\\_systém\)](http://cs.wikipedia.org/wiki/Android_(operační_systém)))
- [3] Wikipedia (en) – *Android - operační systém*  
([http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)))
- [4] Wikipedia (en) – *Android - historie verzí*  
([http://en.wikipedia.org/wiki/Android\\_version\\_history](http://en.wikipedia.org/wiki/Android_version_history))
- [5] Apple developer – *iOS*  
(<https://developer.apple.com/devcenter/ios/index.action>)
- [6] Android (en) – *Android - about*  
(<http://source.android.com/about/index.html>)
- [7] Android (en) – *Application Fundamentals*  
(<http://developer.android.com/guide/topics/fundamentals.html>)
- [8] Android (en) – *What is Android?*  
(<http://developer.android.com/guide/basics/what-is-android.html>)
- [9] Android (en) – *SDK Tools*  
(<http://developer.android.com/guide/developing/tools/index.html>)
- [10] Android (en) – *AndroidManifestFile*  
(<http://developer.android.com/guide/topics/manifest/manifest-intro.html>)
- [11] Android (en) – *SDK*  
(<http://developer.android.com/sdk/installing.html>)
- [12] Adobe (en) – *Flex SDK*  
(<http://www.adobe.com/products/flex.html>)
- [13] Interval (cz) – *Co je a co není Flex*  
(<http://interval.cz/clanky/adobe-flex-co-je-a-co-neni/>)

- [14] Adobe (cz) – *Flex*  
([http://cs.wikipedia.org/wiki/Adobe\\_Flex](http://cs.wikipedia.org/wiki/Adobe_Flex))
- [15] Wikipedia (en) – *HTML5*  
(<http://en.wikipedia.org/wiki/HTML5>)
- [16] HTML5 (cz) – *Vše co potřebujete vědět*  
(<http://www.html5.cz/>)
- [17] Xamarin (en) – *Mono for Android - Introduction*  
([http://docs.xamarin.com/android/getting\\_started/Introduction\\_to\\_Mono\\_for\\_Android](http://docs.xamarin.com/android/getting_started/Introduction_to_Mono_for_Android))
- [18] Xamarin (en) – *Mono for Android - Package size*  
([http://docs.xamarin.com/android/tutorials/Application\\_Package\\_Sizes](http://docs.xamarin.com/android/tutorials/Application_Package_Sizes))
- [19] Android (en) – *Native x86 emulator support*  
(<http://androidcommunity.com/android-sdk-tool-updated-with-native-x86-emulator-support-20120321/>)
- [20] Android (en) – *Android App Developers GUI Kits, Icons, Fonts and Tools*  
(<http://speckyboy.com/2010/05/10/android-app-developers-gui-kits-icons-fonts-and-tools/?modify=0>)
- [21] Android (en) – *Providing resources*  
(<http://developer.android.com/guide/topics/resources/providing-resources.html>)
- [22] Soap (cz) – *Kosek – Inteligentní podpora navigace na WWW s využitím XML*  
(<http://www.kosek.cz/diplomka/html/websluzby.html>)
- [23] Mark L. Murphy *Android 2 – Průvodce programováním mobilních aplikací* COMPUTER PRESS, 2011.
- [24] Wallace B. McClure, Nathan Blevins, John J. Croft, IV, Jonathan Dick, Chris Hardy – *Professional Android Programming with Mono for Android and .Net/C#* John Wiley and Sons.

## A. Uživatelská příručka

Popisuje požadavky na aplikaci, její instalaci a popis obsluhy. Popis obsluhy bude řešen jako popis nejčastějších „případů užití“.

Podle [3] verzi Androidu 2.0 a 2.1 stále používá 5% uživatelů. Toto číslo je malé a pořád se zmenšuje, proto je aplikace tvořena a testována na zařízeních s verzí 2.2 (Froyo 20,9%) a vyšších (nejvýznamnější Gingerbread – 2.3.x s 64,4%), není však zatím určena pro tablet. Funkčnost na verzích nižších a pro tablet se nevyklučuje, přestože zobrazení nemusí odpovídat standardu pro danou verzi systému.

Ke správnému fungování je zapotřebí dotykový mobilní telefon s OS Android. Jsou podporovány všechny velikosti obrazovek i rozlišení. K funkci aplikace je využívána orientační poloha GPS získávaná z wifi sítí a sítí mobilních operátorů. Tudíž funkce GPS v telefonu není vyžadována.

### A.1. Instalace

V přiloženém CD nosiči (v budoucnu i na GooglePlay i na serveru viaReality.cz) je instalační balíček s příponou *.apk*. Tento si stáhněte do mobilního zařízení (na kartu či do hlavní paměti).

Protože aplikace ještě není veřejně dostupná a „podepsaná“, před instalací je potřeba povolit položku v *Nastavení - Aplikace - Neznámé zdroje*. Nyní stažený soubor najděte a spusťte.

Nyní vás systém vyzve k potvrzení požadovaných práv aplikací (v tomto případě připojení k internetu a orientační poloha GPS). Po potvrzení a nainstalování máte možnost aplikaci otevřít. (Obrázek č. 1.)

### A.2. Scénář první – nalezení nemovitosti podle viaKódu, její zobrazení na mapě a poslání dotazu

Hned první okno aplikace, kromě úvodu, slouží jako rozcestník. Najděte dlaždici s názvem *viaKód*. Zobrazí se textové pole, kam zadáte viaKód (lze ho získat v katalogu nebo na stránkách viaReality.cz). Po potvrzení začne aplikace vyhledávat.

Pokud nejste připojeni k internetu jste vyzváni tak učinit. Pokud jste viaKód zadali špatně, nebo daná nemovitost v systému již není, aplikace Vás na tuto skutečnost upozorní. Jinak se Vám zobrazí detail hledané nemovitosti.

Můžete vidět název, adresu a cenu nemovitosti. Níže pak fotogalerii, kterou si kliknutím zvětšíte. Následuje slovní popis nemovitosti, který se v plném rozsahu zobrazí po kliknutí. Pro strukturované informace je zde políčko níže, které opět po kliku zobrazí jejich plný rozsah.

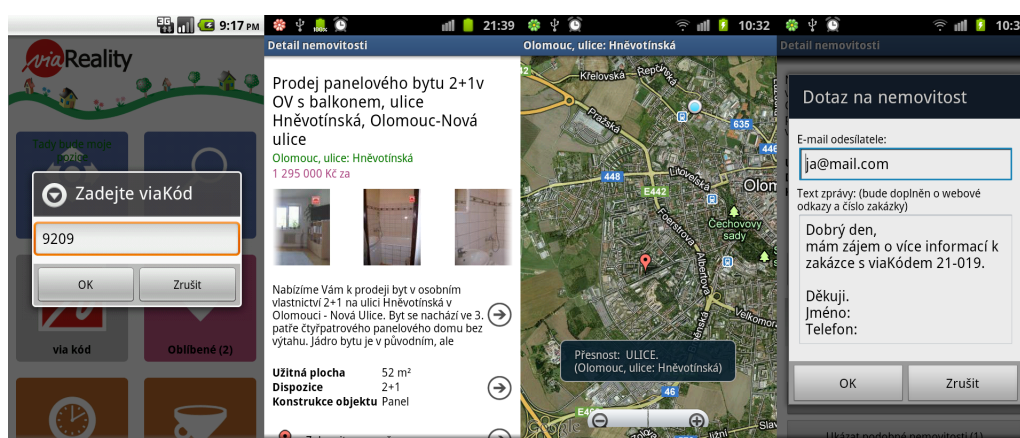
Níže na stránce je tlačítko náhledu v mapě, kdy se po kliku zobrazí mapa. Ta se „vystředí“ na adresu uvedenou nahoře a po kliknutí na „pin“ ve středu

mapy se vypíše přesnost zobrazení adresy. Modrým „puntíkem“ je označena vaše poloha.

Zpět na detailu kliknutím na tlačítko *Dotaz na nemovitost* můžete poslat dotaz na makléře ohledně dané nemovitosti. Stačí vyplnit váš email a v případě, že nemáte konkrétní dotaz, poslat před vyplněný formulář a čekat na odpověď.

Jestli Vás zajímá makléř nebo realitní kancelář pro tuto nemovitost, klikněte na políčko makléře. Zde máte uvedené kontaktní údaje makléře a realitní kanceláře s adresou, možností zobrazení jejich nemovitostí a webovou adresu.

Zpět na detailu jsou ve spodní části stránky funkční tlačítka pro podobné nemovitosti, uložení do oblíbených, poslat email s odkazem a sdílet.



Obrázek 14. Scénář 1 – zadání viaKódu, detail nalezené nemovitosti, její zobrazení na mapě, poslání dotazu k nemovitosti.

### A.3. Scénář druhý – hledání nemovitostí vytvořením filtru

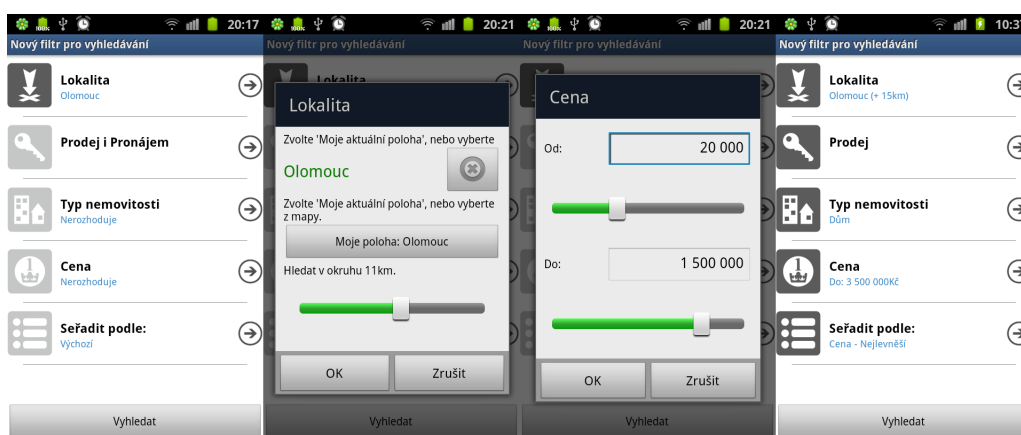
Příklad filtru může být „prodej domu v Olomouci a okolí do 15km s cenou do 3,5mil korun řazeno od nejnižší ceny“.

Pokud chceme hledat podle filtru, je ho třeba vytvořit. Na hlavní obrazovce klikněte na *Nové hledání*. Vyberte *Lokalita*. Pokud se aplikaci povedlo již zjistit vaši aktuální polohu, je před vyplněna. V tom případě zmáčkněte křížek vpravo a zadejte Olomouc. Jste-li připojeni k internetu našeptávač by měl napovědět „obec Olomouc“. (Pokud jste z Olomouce a byla zjištěna Vaše poloha – lze kliknout na tlačítko *Moje poloha* – lze použít i to.) Nastavte *Hledat v okruhu* na 15km. A potvrďte *OK*.

U druhé položky nastavte *Prodej*. U položky *Typ nemovitosti* zaškrtněte *Dům*. U *Cena* si posuvníkem nastavte cenu *Do*, v tomto případě zadáte částku ručně

na 3500000 a potvrdíte. Jako poslední nastavte *Seřadit podle* hodnotu *Cena - Nejlevnější*. (Možnost řadit dle vzdálenosti je přístupná pouze pokud je v *Lokalita* nastaveno *Hledat v okruhu* větší než nula.) A vyhledat.

Po ověření zda jste připojeni k internetu se zobrazí seznam nalezených nemovitostí řazených podle ceny. V záhlaví je uveden filtr, pod ním dvě tlačítka. Jedno pro úpravu filtru a druhé pro změnu řazení nemovitostí. Posunem seznamem dolů se načítají další nemovitosti automaticky. Po kliknutí na řádek se zobrazí detail nemovitosti (viz. kapitola A.2.).



Obrázek 15. Scénář 2 – nový filtr, zadání polohy, zadání ceny, výsledný filtr.

#### A.4. Scénář třetí – hledání nemovitostí v okolí

Pro situace, kdy Vás zajímají nemovitosti v okolí, je na hlavní obrazovce tlačítko *Mé okolí*, které ve své horní části zobrazuje Vaši aktuální polohu (v případě že už je známa). Po kliknutí na toto tlačítko se Vám (po ověření připojení k internetu) zobrazí seznam všech nemovitostí ve Vašem okolí řazený podle vzdálenosti.

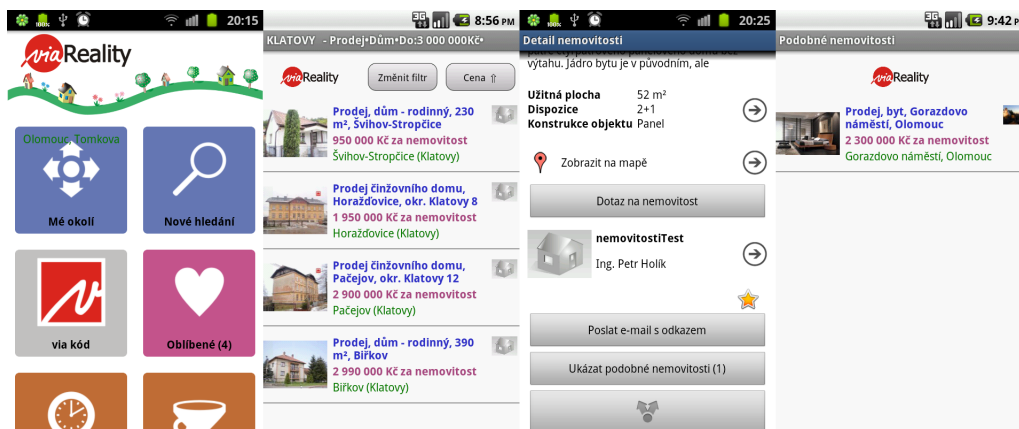
Při volbě tlačítka nahoře *Změnit filtr* se dostáváte do nastavení filtru, kde si můžete filtrování nemovitostí zúžit podle parametrů (viz kapitola A.3.).

#### A.5. Scénář čtvrtý – hledání nemovitostí realitní kanceláře

Pokud Vás zajímají nemovitosti konkrétní realitní kanceláře, využijte tlačítka *Realitní kanceláře* a do vrchního políčka zadejte název realitní kanceláře nebo obec/město, kde se nalézá. Výsledky hledání vidíte po kliknutí na symbol „lupy“.

Po kliknutí na řádek s požadovanou realitní kanceláří se zobrazí její detail (kromě údaje o makléři je podobný jako v kapitole A.2. – *Detail makléře*). Na de-





Obrázek 16. Scénář 3 – hlavní obrazovka aplikace (Mé okolí), nalezené nemovitosti, detail nemovitosti (jeho spodní část s funkcemi), zobrazení podobných nemovitostí k právě prohlíženému.

tailu je vidět název, adresa, kontaktní údaje, poloha na mapě a odkaz na webové stránky.

Číslo v závorce u tlačítka *Nemovitosti realitní kanceláře* znamená počet inzzerujících nemovitostí na našem serveru. Po kliknutí se zobrazí seznam těchto nemovitostí, který přes tlačítko *Změnit filtr* můžete zúžit filtrování nemovitostí této realitní kanceláře podle parametrů (viz kapitola A.3.).

Skutečnost, že hledáte jen nad nemovitostmi konkrétní realitní kanceláře, poznáte tak, že v záhlaví stránky je uveden její název před výčtem filtru. Nebo na každém řádku seznamu je zobrazeno stejné logo realitní kanceláře.



Obrázek 17. Scénář 4 – hledání realitní kanceláře, detail realitní kanceláře, zobrazení jejích nemovitostí, filtrování nemovitostí konkrétní realitní kanceláře.

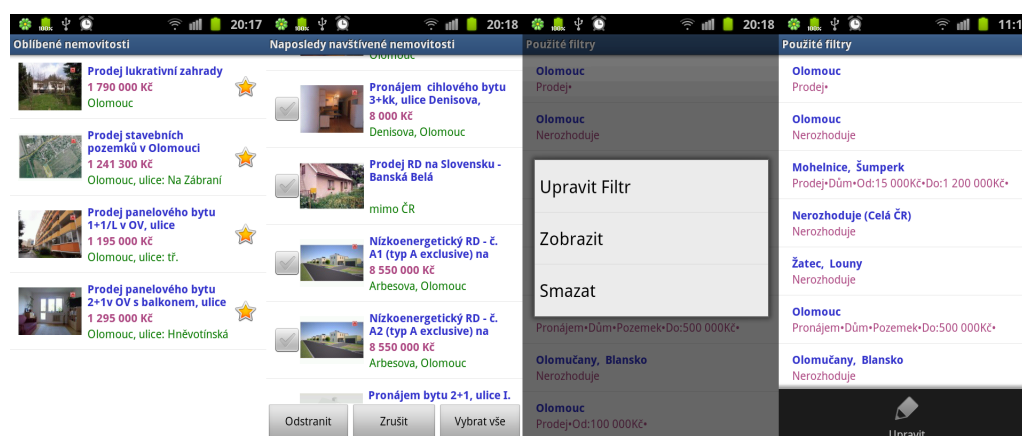


## A.6. Scénář pátý – využívání uložených informací

*Historie* – v historii je posledních padesát zobrazených nemovitostí (jejich detailu). V historii je možné mazat pomocí kontextové nabídky při podržení prstu nad konkrétním řádkem, nebo pomocí *menu – upravit*. Zaškrtnutím požadovaných řádků a kliknutím na smazat.

*Filtry* – zde je seznam použitých filtrů. Kliknutím na filtr je tento vykonán a výsledek zobrazen. Pokud chcete filtr změnit, můžete tak učinit v kontextové nabídce podržením prstu nad konkrétním filtrem, nebo v jeho výsledcích kliknout na *Změnit filtr*. Ostatní chování je totožné s nabídkou *Historie*.

*Oblíbené* – pod tímto tlačítkem je schován seznam Vašich oblíbených nemovitostí, které jste tak na stránce detailu nemovitosti (dole) označili. Tento seznam má výhodu, že jsou zde jen Vámi vybrané nemovitosti, a pokud nejste připojeni k internetu, jsou vám jednotlivé detaily nemovitostí nabídnuty offline, včetně obrázků. Pokud jste připojeni a zobrazíte detail, nemovitost se načte opět z internetu a aktualizuje se. Práce nad tímto seznamem je totožná se seznamem *Historie*.



Obrázek 18. Scénář 5 – seznam oblíbených, úprava seznamu naposledy navštívených, kontextová nabídka nad seznamem filtrů, vyvolání menu nad seznamem filtrů.

## B. Programátorská dokumentace

Aplikace byla vyvíjena s pomocí nástroje *Microsoft Visual Studio 2010* v jazyce C# v rozšíření *Mono for Android 4.0.6* a cílena na Android API verze 8 (Android verze 2.2 – Froyo). Při vývoji byla využívána Android SDK spolu s emulátory k příslušným verzím.

Aplikace je rozdělena do dvou samostatných projektů. První tvoří samotnou vizuální prezentaci, uživatelské rozhraní a interakci s uživatelem. Druhá je samostatná knihovna starající se o komunikaci se serverem, zpracováním dat a jejich přípravou pro zobrazení, jejich ukládáním na lokální úložiště, zpracováním a „logováním“ chyb.

Snažil jsem se o klasické pojetí s vytvořením třívrstvé architektury, kdy jsou odděleny vrstvy prezenční, aplikační a datové s ohledem na znovu použitelnost komponenty v případě implementace pro jiný mobilní OS.

Komunikace se serverem je zajištěna pomocí webových služeb (více kapitola [B.2.5.](#)).

### B.1. GUI

Grafické uživatelské rozhraní aplikace je tvořeno z několika „obrazovek“, které jsou na sobě téměř nezávislé. Existuje mezi nimi datová výměna, případně otvírání nových obrazovek a čekání na výsledek z nich. Každá obrazovka má své vlákno uživatelského rozhraní, když nastane problém se zobrazením (špatná data, neošetřená výjimka), je tato obrazovka zavřena a potažmo aplikace ukončena s chybou. Ale na zásobníků obrazovek můžou být další předešlé obrazovky aplikace a tady je čistě na aplikaci jak se zachová. Pokud má k dispozici všechna potřebná data (musí dojít k jejich opětovnému načtení), je aplikace schopná se z chyby zotavit a pokračovat dále.

#### B.1.1. Screens

V adresáři *Screens* jsou uloženy všechny obrazovky aplikace, tedy jejich kódová část, které všechny dědí z třídy *Activity*. Pro každou je popsána její inicializace, co se má dělat při otočení zařízení, když zařízení přejde do režimu spánku, či probuzení. Definují se zde veškeré služby, které jsou na stránce využívány (např. kontrola připojení na internet, získávání GPS polohy zařízení, ...).

Níže následuje výpis nejzajímavějších z nich.

*SplashActivity* – úvodní obrazovka slouží k inicializaci aplikace, spuštění nového procesu a nastavení potřebných věcí pro její bezproblémový chod. Zde načítám veškerá data o uživateli (emaily, velikost obrazovky nebo umístění zařízení), data o jeho předešlé aktivitě (historie, používané filtry, oblíbené) a následně spouštím hlavní obrazovku aplikace.

*MainPic* – je hlavní obrazovka aplikace. Souží jako rozcestník ke všem důležitým částem aplikace.

*NewSearch* – slouží k zadávání nového filtru, patří k nejsložitějším obrazovkám aplikace. V základu vypisuje nastavení filtru, při kliku otvírá jednoúrovňová dialogová okna sloužící pro jeho jednotlivá dílčí nastavení.

*Busy* – má funkci rozcestníku pro stahovaná data zajišťující jejich kontrolu a následné přeposlání pro zobrazení. Je zodpovědná za kontrolu připojení a případné vypisování chybových stavů s tímto spojených.

*Map* – je obrazovka vykreslující mapu spolu s adresou nemovitostí, realitních kanceláří a aktuální polohu zařízení.

*PropertyList* – slouží jako hlavní seznam aplikace zobrazující nalezené nemovitosti v okolí, podle filtru, podobné nemovitosti a nemovitosti realitních kanceláří.

*PropertyDetail*, *PropertyDetailInfo*, *BrokerDetail*, *EADetail*

Obrazovky sloužící pro výpis detailních informací daného subjektu.

kolekce tříd s přívlastkem *My*

Jedná se o soubor obrazovek poskytujících vizuální podporu pro uložená data. Najdeme zde seznam *Historie*, který zobrazuje formou fronty padesát posledních zobrazených nemovitostí, nebo seznam použitých filtrů. Další je seznam *Oblíbených*. Seznamy se vizuálně ani funkčností příliš neliší. Rozdíl představuje pouze obrazovka *MyEA*, která slouží k vyhledávání realitních kanceláří.

### B.1.2. Adapters

Třídy adaptérů pro jednotlivé seznamy. Kontejnery obsahující data a zajišťující přístup k nim spolu s jejich vykreslováním na stránce. Kvůli šetření paměti, zejména u zobrazování složitých *Layoutů* jednoho prvku, systém používá *ConvertView*. Ve finále to znamená, že okno má graficky vytvořeno pouze několik řádků (co se vleze na obrazovku) a mění pouze zobrazovaná data.

### B.1.3. Aplikační vrstva (AL)

V adresáři *AL* jsou obsaženy třídy různých „pomocníků“. Jsou tu třídy pro obsluhu obrázků, zjišťování polohy, ověřování konektivity, posílání emailů (dotaz, posílání linku), pro tvorbu kontextového a normálního menu a další pomocné.

Některé rozšiřují funkčnost základní třídy *Activity* a je z nich dále děděno jiné jsou samostatné. Zmínil bych třídy pro obsluhu obrázků, které jsou částečně převzaté a upravené (zdroj: <http://samples.xamarin.com/android> – MWC2012)

z volně dostupných a šiřitelných zdrojů. U všech takto použitých tříd jsou komentáře v původním znění spolu s uvedením autora a licenčních podmínek. Jedná se zejména o koncepci stahování, hashování a ukládání, následné znovu používání obrázků (adresář *ImageLoader*).

V aplikaci jsou zpřístupněny dva způsoby posílání mailů z aplikace. Jedna vede přes komunikační server kdy se vyplňuje minimum dat. Jedná se o způsob vyplnění jednoduchého formuláře, většinou předdefinovaného, takže stačí jen odeslat. Je zde kvůli jednodušší obsluze a zadáte si konkrétní mail, který běžně používáte. Nebo způsob druhý, za využití sdílení, kdy si zvolíte nějakého mobilního e-mailového klienta (většinou GMail), který je svázaný s telefonem.

#### B.1.4. Resources

Zdroje aplikace. Jedná se o adresář, kde v další adresářové struktuře jsou podpůrná data pro aplikaci. Ta se dají shrnout do dvou kategorií. Jedná se buďto o grafiku (veškeré ikony, záslepky, loga, obrázky) nebo o soubory typu xml.

Grafika je celkem jasná. Je zde hlavní adresář zvaný *Drawable* kde je uložena na jedné hromadě. Pokud vyvíjíte vůči větší škále zařízení, jež disponují různými displeji (velikostí, rozlišením), je dobré uvádět alternativní zdroje (v mém případě různé velikosti obrázků). Celková struktura je popsána v [21]. Pomocí tohoto systému jednu aplikaci lze spustit na různých zařízeních s různými texty, jazyky, obrázky, rozložení prvků na obrazovce při zachování stejné funkčnosti.

V souborech typu xml jsou data používaná v aplikaci, ale definována mimo (ne v kódu). To zpřehledňuje definování třeba rozložení prvků na obrazovce, textových řetězců pro aplikaci, definice barev, polí, témat, stylů, ...

Druhý nejdůležitější adresář je *Layout*. tady jsou ne-programově definována rozložení prvků pro všechny obrazovky. Můžou být komplet, nebo jen jejich části (kdy si definují jeden řádek seznamu, nebo strukturu nadpisu stránky).

Z dalších může obsahovat *Menu* pro definici kontextových a normálních menu na jednotlivých obrazovkách, nebo adresář *Values*, který obsahuje právě textové řetězce, pro různé jazyky, styly, barvy.

## B.2. Knihovna .dll

Knihovna je téměř odříznuta od Androidího „frontendu“. Má na starost komunikaci se serverem, zpracování dat, jejich ukládání na lokální úložiště, kde se opět využívají funkce příslušné jen Androidu.

Asi nejdůležitější tři třídy jsou *UserManager*, *UserSearchManager* a *ServiceCommunicator*, které obsahují většinu dat sloužící aplikaci.

*UserManager* – je třída pro nastavení uživatele, jsou zde drženy informace o velikosti displeje zařízení, emaily používané pro před vyplňování formulářů, nebo pamatování si aktuální polohy zařízení. Údaj o velikosti displeje

se používá při stahování jakýchkoli obrázků, kdy jsou stahovány obrázky konkrétních rozměrů pro konkrétní zařízení. Takže obrázky na zařízení s menším displejem jsou menší, nikoliv pouze zmenšené zařízení. Aby nebyl systém neustále dotazován na polohu, tato se kontroluje jen na dvou místech aplikace, jinde už se používají pouze výsledky této služby. Třída je vytvářena jako „jedináček“ – *Singleton*, tak aby nemohlo existovat více instancí najednou.

*UserSearchManager* – tento manager se stará kompletně o vyhledávací data, včetně dat výsledků. Jsou zde uložena veškerá stažená data; detaily prohlížených nemovitostí, realitních kanceláří a makléřů, ... Aby uživatel nebyl omezován čistě na lineární použití aplikace (jeden filtr, jedny výsledky, jeden detail nemovitosti), jsou data ukládána v seznamech korespondujícími se zásobníkem otevřených obrazovek na zařízení. Tudíž místo vytvoření „kruhu“ *PropertyList* – *PropertyDetail* – *BrokerDetail* – *PropertyList* (seznam nemovitostí odpovídající konkrétní realitní kanceláři) – *PropertyDetail* – *PropertyList* (v tomto případě zobrazení podobných nemovitostí) – atd., kdy by se zobrazovaná data postupně přemazávala, vytvořit spíše *spirálu* tak, že při použití tlačítka zpět aplikace korektně zobrazuje data i cestou zpět. Třída je vytvářena jako „jedináček“ – *Singleton*, tak aby nemohlo existovat více instancí najednou.

*ServiceCommunicator* – je jednoduchá třída zajišťující komunikaci se serverem na bázi webových služeb protokolu SOAP. Jejich popis je uveden v kapitole [B.2.5](#). Třída je vytvářena jako „jedináček“ – *Singleton*, tak aby nemohlo existovat více instancí najednou.

Nacházejí se zde i *Constants*, kde jsou drženy používané konstanty pro celou aplikaci.

### **B.2.1. Business vrstva (BL)**

V business vrstvě jsou definice partial tříd rozšiřující objekty přenášené při komunikaci mezi serverem a zařízením. Tyto objekty se dále používají i při ukládání, kdy se serializují, a ve formátu xml jsou uloženy na lokální úložiště.

### **B.2.2. Datová vrstva (DL)**

Datová vrstva se stará o uchovávání objektů v dobrém stavu na lokálním úložišti. Třída je vytvářena jako *Singleton*. Každá aplikace má svůj vlastní prostor, kam se jiné nedostanou. Na tento prostor nepotřebuje aplikace žádná speciální práva. Může si v něm vytvářet libovolnou adresářovou strukturu. K ukládání jsou použity serializované objekty do binárního xml formátu (kvůli velikosti). Při opětovném načítání jsou vytvořeny komplet objekty.

### B.2.3. Helpers

Tady se nachází třída zodpovědná za logování veškerých chyb aplikace a generování chybových hlášení, které jsou následně odeslané na server. Jsou zde zaznamenány i chyby z neošetřených výjimek, ke kterým by samozřejmě docházet nemělo, ale stát se může. Jak jsem již psal v B.1. aplikace by se měla zotavit z většiny neošetřených chyb.

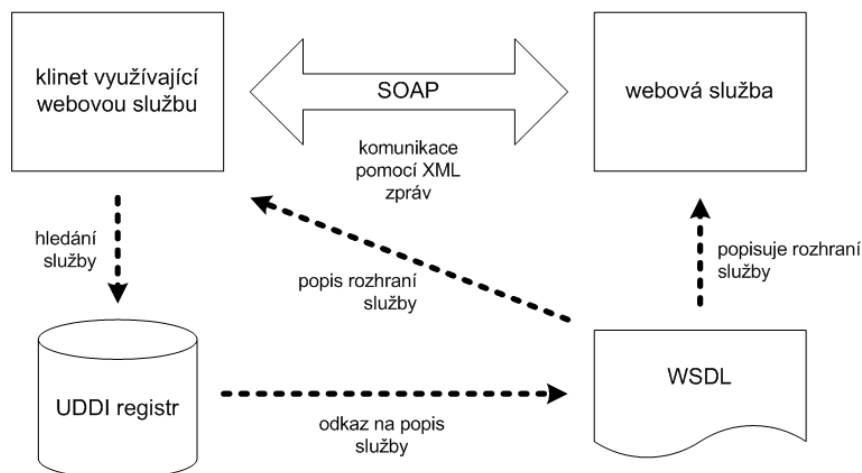
### B.2.4. Extensions

V tomto adresáři jsou třídy rozšiřující objekty a slouží pro správné uživatelské formátování adresy, čísel, atd.

### B.2.5. Webové reference

Aplikace využívají ke komunikaci služeb založených na protokolu *SOAP* – (*Simple Object Access Protocol*) – schéma komunikace na obrázku č. 19.. Přesto, že se nejedná o protokol s minimálním přenosem dat (přenáší se XML), dobře posloužil. Protože většinu špinavé práce udělali automatické generátory kódu, kdy na straně serveru mám vytvořeny objekty nachystané jako odpověď na dotaz. Systém tyto objekty sám převede na xml a výsledek pošle přes *http protokol*. Na straně aplikace je toto xml automaticky zpracováno a jsou vytvořeny stejné objekty jako byly poslány. Tudíž odpadá nutnost jednak si poslané xml „parserovat“ ručně a jednak neustálé doplňování nových položek na dvou místech (klient, server).

Aplikace v současnosti využívá dvou služeb. Jedna kompletně zajišťuje provoz a komunikaci pro získávání dat hledání nemovitostí a druhá se stará o našeptávač k zadávání polohy.



Obrázek 19. Komunikace SOAP.

Zdroj: <http://www.kosek.cz/diplomka/html/websluzby.html>

## C. Obsah příloženého CD

V samotném závěru práce je uveden stručný popis obsahu příloženého CD/DVD, tj. závazné adresářové struktury, důležitých souborů apod.

**bin/**

Instalátor \*.APK aplikace na mobilní zařízení s OS Android s dotykovou obrazovkou a verzí systému 2.2 (Froyo) a vyšší.

**doc/**

Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PřF pro diplomové práce, včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování PDF souboru dokumentace (v ZIP archivu), tj. zdrojový text dokumentace, vložené obrázky, apod.

**src/**

Kompletní zdrojové texty aplikace se všemi potřebnými (převzatými) zdrojovými texty, knihovnami a dalšími soubory pro bezproblémové vytvoření spustitelných verzí programu.

**readme.txt**

Instrukce pro instalaci a spuštění aplikace, včetně požadavků pro jeho provoz.

Navíc CD/DVD obsahuje:

**install/**

Instalátory aplikací, knihoven a jiných souborů nutných pro možnost kompilace aplikace, které nejsou standardní součástí operačního systému.

U veškerých odjinud převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovoluují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro materiály, u kterých toto není splněno, je uveden jejich zdroj (webová adresa) v textu dokumentace práce nebo v souboru `readme.txt`.