

# **Aplikace lokace budov UTB ve Zlíně pro OS Android**

Jan Hruban

---

Bakalářská práce  
2014



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

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

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan HRUBAN**  
Osobní číslo: **A10122**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Informační a řídicí technologie**  
Forma studia: **prezenční**

Téma práce: **Aplikace lokace budov UTB ve Zlíně pro OS Android**

Zásady pro vypracování:

1. Vypracujte kompletní přehled všech budov a umístění Univerzity Tomáše Bati ve Zlíně spolu s detašovanými pracovišti.
2. Seznamte se s vývojovými prostředky pro vývoj aplikací OS Android.
3. Vytvořte aplikaci OS Android s přehledem těchto adres.
4. Vytvořte navigační modul k daným adresám (komunikace přístroje s GPS modulem a mapami).
5. Vytvořte uživatelskou dokumentaci.
6. Vytvořenou aplikaci otestujte. Vytvořte záznam výhod a nevýhod Vašeho řešení.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **MURPHY, Mark L. Android 2: Průvodce programováním mobilních aplikací. Vyd. 1. Brno: Computer Press, 2011, 375 s. ISBN 978-80-251-3194-7.**
2. **Android Developers. Android Developers [online]. 2009 [cit. 2014-01-26]. Dostupné z:<http://developer.android.com/>**
3. **ALLEN, Grant. Android 4: průvodce programováním mobilních aplikací. 1. vyd. Překlad Jakub Mužík. Brno: Computer Press, 2013, 656 s. ISBN 978-80-251-3782-6.**
4. **VÁVRŮ, Jiří a Miroslav UJBÁNYAI. Programujeme pro Android. 2., rozš. vyd. Praha: Grada, 2013, 250 s. Průvodce (Grada). ISBN 978-80-247-4863-4.**
5. **HEROUT, Pavel. Java a XML. 1. vyd. České Budějovice: Kopp, 2007, 313 s. ISBN 978-80-7232-307-4.**

Vedoucí bakalářské práce:

**Ing. Dalibor Slovák**

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

**28. února 2014**

Termín odevzdání bakalářské práce:

**13. června 2014**

Ve Zlíně dne 28. února 2014



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



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

## **Prohlašuji, že**

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

## **Prohlašuji,**

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

Ve Zlíně

.....  
podpis diplomanta

## **ABSTRAKT**

Tato bakalářská práce se zabývá vývojem aplikace pro operační systém Android. V první části práce je popsán samotný operační systém, jeho historie a základní architektura. Ve druhé části se práce věnuje aplikacím, jejich struktuře a vývojovým prostředkům. Třetí část práce je věnována návrhu konkrétní aplikace, obsahuje shrnutí základních požadavků na aplikaci, grafický návrh a návrh architektury aplikace. Čtvrtá část práce se již zabývá implementací, testováním a také tvorbou uživatelské dokumentace.

Klíčová slova: Android, tablety, chytré telefony, programování, lokace budov, mapy

## **ABSTRACT**

This bachelor thesis deals with application development for operating system Android. In the first part of this thesis is described the operating system itself, its history and basics of its architecture. The second part analysis applications their structure and development. Third part describes design of specific application, which include basic requirements, graphic design and architecture of application. Last part of this thesis deals with implementation, testing and distribution of application itself and also creating of user guide.

Keywords: Android, Tablets, Smartphones, Programming, Buildings Location, Maps

Na tomto místě bych rád poděkoval vedoucímu mé bakalářské práce panu Ing. Daliboru Slovákovi za poskytování cenných rad a také své rodině za podporu během studia.

# OBSAH

<b>ÚVOD .....</b>	<b>9</b>
<b>I TEORETICKÁ ČÁST .....</b>	<b>10</b>
<b>1 ANDROID .....</b>	<b>11</b>
1.1 HISTORIE ANDROIDU .....	11
1.1.1 Android 1.5 .....	12
1.1.2 Android 1.6 .....	12
1.1.3 Android 2.0 .....	12
1.1.4 Android 2.2 .....	13
1.1.5 Android 2.3 .....	13
1.1.6 Android 3.0/3.1/3.2.....	13
1.1.7 Android 4.0 .....	14
1.1.8 Android 4.1/4.2/4.3.....	14
1.1.9 Android 4.4 .....	14
1.2 ROZTRÍŠTĚNOST ANDROIDU.....	14
1.3 ARCHITEKTURA ANDROIDU.....	16
1.3.1 Linux Kernel .....	16
1.3.2 Libraries .....	17
1.3.3 Android runtime .....	17
1.3.4 Application Framework .....	18
1.3.5 Applications .....	18
1.4 DISTRIBUČNÍ SÍŤ GOOGLE PLAY.....	18
<b>2 ANDROID APLIKACE.....</b>	<b>20</b>
2.1 VÝVOJOVÉ PROSTŘEDKY .....	20
2.1.1 Java Development Kit.....	21
2.1.2 Android Software Development Kit.....	21
2.1.3 Eclipse.....	22
2.1.4 Android development tool .....	22
2.2 ADRESÁŘOVÁ STRUKTURA .....	22
2.2.1 Popis adresářů v novém Android projektu.....	23
2.3 KOMPONENTY APLIKACE.....	23
2.3.1 Activity .....	24
2.3.2 Service .....	25
2.3.3 Content providers .....	27
2.3.4 Broadcast receivers .....	27
2.3.5 Intenty .....	28
2.3.5.1 Explicitní intenty .....	28
2.3.5.2 Implicitní intenty .....	28
2.3.6 Fragmenty .....	29
2.4 PROSTŘEDKY APLIKACE.....	31
2.4.1 Přehled jednotlivých prostředků.....	32
2.5 WIDGETY.....	32
2.5.1 TextView.....	32
2.5.2 Button.....	32
2.5.3 ImageView, ImageButton .....	33

2.5.4	ListView.....	33
2.5.5	Tabs.....	33
2.6	SOUBOR ANDROIDMANIFEST.XML.....	34
<b>II</b>	<b>PRAKTICKÁ ČÁST .....</b>	<b>35</b>
<b>3</b>	<b>PŘÍPRAVA PODKLADŮ A NÁVRH APLIKACE.....</b>	<b>36</b>
3.1	NÁVRH APLIKACI .....	36
3.1.1	Shrnutí základních požadavků .....	36
3.1.2	Struktura aplikace .....	37
3.1.3	Grafický návrh.....	37
3.2	PŘÍPRAVA PODKLADŮ.....	38
<b>4</b>	<b>VÝVOJ APLIKACE .....</b>	<b>39</b>
4.1	POUŽITÉ VÝVOJOVÉ PROSTŘEDKY .....	39
4.2	ZALOŽENÍ PROJEKTU .....	39
4.3	TVORBA ZÁKLADNÍHO VZHLEDU APLIKACE.....	40
4.3.1	Support Library .....	40
4.3.2	Změny v hlavní aktivitě a definice stylu.....	42
4.4	POPIS HLAVNÍCH KOMPONENT.....	44
4.4.1	MainActivity .....	44
4.4.2	AddressActivity.....	45
4.4.3	BuildingsListFragment .....	45
4.4.4	AddressFragment.....	45
4.4.5	MapFragment .....	45
4.4.6	BuildingsListAdapter.....	45
4.5	ANDROIDMANIFEST.XML .....	45
4.6	TESTOVÁNÍ APLIKACE .....	46
4.6.1	Výsledky testů .....	46
4.7	UŽIVATELSKÁ DOKUMENTACE.....	47
	<b>ZÁVĚR .....</b>	<b>51</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>52</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>54</b>
	<b>SEZNAM TABULEK .....</b>	<b>56</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>57</b>

## ÚVOD

Chytrá mobilní zařízení, ať už se jedná o chytré telefony, tablety nebo dokonce chytré hodinky, zažívají během posledních pár let obrovský boom. Je to způsobeno především pokrokem v oblasti miniaturizace hardwaru, který tato zařízení pohání. Není tomu tak dávno, co byl čtyř jádrový procesor výsadou těch nejdražších počítačových sestav. Dnes je čtyř jádrový procesor standardem většiny chytrých mobilních telefonů. Tato chytrá zařízení tak v sobě kombinují funkce osobních počítačů, například přístup k internetu, poslech hudby, přehrávání videí, psaní textu atd., s výhodami mobilních telefonů, jako je telefonování, využívání mobilního internetu nebo navigace přes GPS. To vše zabaleno do minimálních rozměrů, umožňujících přenášet tato zařízení bez problémů z místa na místo a mít všechny tyto funkce kdykoliv po ruce. K nevýhodám těchto zařízení patří bezesporu výdrž současných baterií. Velké dotykové displeje jsou velmi náročné na energii a průměrná výdrž, při průměrném používání chytrého telefonu, se v dnešní době pohybuje v řádu desítek hodin.

Aby bylo možné tato chytrá zařízení používat, musí být poháněny operačním systémem. K neznámějším operačním systémům na poli chytrých zařízení jsou Android od společnosti Google, iOS od Applu, Windows od Microsoftu a BlackBerry, přičemž prvně dva jmenované ovládají drtivou většinu dnešního trhu s chytrými telefony a tablety. Operační systém Android v dnešní době ovládá téměř 80% trhu s chytrými telefony. Jeho oblíbenost spočívá především v tom, že je na rozdíl od ostatních OS založen na Open Source platformě, navíc je Android od počátku navržen tak, aby ho bylo možné provozovat na různých konfiguracích hardwaru. Android tak dnes využívá mnoho předních výrobců chytrých zařízení jako je Samsung, Sony, LG, HTC a mnoho dalších.

Cílem této bakalářské práce bude v první části popsat právě OS Android a vývoj aplikací pro tuto platformu a ve druhé ukázat vývoj konkrétní aplikace od návrhu až po samotnou implementaci.

## **I. TEORETICKÁ ČÁST**

## 1 ANDROID

Operační systém Android pohání miliony mobilních zařízení ve více než 190 zemích světa. Jedná se momentálně o nejrozšířenější platformu pro mobilní zařízení a každým dnem přibývá více než milion dalších uživatelů. [1]

Android vyvíjí společnost Google a je založený na open source platformě tzn. s otevřeným zdrojovým kódem. Otevřenost zdrojového kódu zde představuje nejen technickou dostupnost, ale i licenci, která umožňuje uživateli, při dodržení jistých podmínek, zdrojový kód využívat nebo i upravovat. [2, s. 15]

Android je založen na Linuxovém jádře 2.6. To se stará o zabezpečení systému, správu paměti a procesů, přístup k síti a ovladače senzorů a komponent zařízení. Android je primárně vyvíjen pro mobilní zařízení, jako jsou tablety a chytré telefony a jeho základní filosofií, je umožnit vývojářům vytvářet aplikace plně využívající vlastností těchto zařízení, jako je například telefonování, posílání SMS zpráv, fotografování, GPS, přístup k mobilnímu internetu a mnoho dalších. [2, s. 15]

### 1.1 Historie androidu

Společnost Android Inc. Byla založena v Kalifornii v říjnu 2003. V srpnu roku 2005 firmu Android Inc. odkoupil Google Inc. a udělal z ní svoji dceřinou společnost. V listopadu roku 2007 bylo vytvořeno uskupení Open Handset Alliance, které zahrnovalo společnosti zabývající se výrobou mobilních zařízení nebo jejich součástí a společnostmi zabývajícími se vývojem mobilních aplikací jako např. Google, HTC, Samsung, Intel, LG, Motorola a dalších, s cílem vytvořit otevřený standard pro mobilní zařízení. Následovalo vydání první Android SDK pro vývojáře pod licencí open source. [2, s. 16] [3]

V říjnu roku 2008 přichází na trh první telefon s OS Android - HTC Dream a spolu s ním bylo uvolněno i SDK 1.0. Od uvolnění první verze bylo vydáno několik aktualizací, které opravují chyby a přidávají nové funkce. Jednotlivé verze jsou pojmenovány po cukrářských dezertech (od verze 1.5) a drží se abecedního pořadí. Zatím poslední verze 4.4 (k březnu 2014) nese název KitKat a stala se tak první verzí Androidu nesoucí název konkrétního výrobku. [2, s. 16]

### 1.1.1 Android 1.5

Verze 1.5, nesoucí název Cupcake, byla uvolněna v dubnu 2009 a přinesla sebou mnoho nových funkcí jak pro uživatele, tak pro vývojáře.

Následuje seznam některých nových funkcí:

- Zpřesnění klíčových prvků uživatelského rozhraní
- Animace při přechodu mezi obrazovkami
- Rotace obrazovky, při rotaci zařízení
- Rychlejší naběh fotoaparátu a pořizování fotografií
- Rychlejší komunikace s GPS
- Domovská obrazovka a widgety
- Softwarová klávesnice a mnoho dalších...

### 1.1.2 Android 1.6

Verze 1.6 nese název Donut a byla uvolněna v září 2009.

Nové funkce:

- Vylepšený android market
- Box pro rychlé vyhledávání
- Vylepšené prostředí pro fotoaparát a galerii
- Zrychlení fotoaparátu
- Indikátor použití baterie a další...

### 1.1.3 Android 2.0

Verze 2.0 s názvem Eclair byla vydána v říjnu 2009 a přinesla sebou například:

- Vylepšená správa kontaktů a účtů
- Nové prostředí webového prohlížeče s podporou HTML5
- Vylepšení softwarové klávesnice
- Podpora pro více rozlišení a velikostí displeje
- Digitální zoom pro fotoaparát
- Podpora pro Bluetooth 2.1 a mnoho dalších

#### 1.1.4 Android 2.2

Verze 2.2 s názvem Froyo byla vydána v květnu 2010 a přinesla sebou opět několik nových funkcí:

- Vylepšené zabezpečení android zařízení
- Nové funkce fotoaparátu
- Možnost vytvořit ze zařízení WiFi hotspot
- Možnosti změny jazykového nastavení klávesnice
- Zlepšení výkonu

#### 1.1.5 Android 2.3

Verze 2.3 s názvem Gingerbread byla vydána v prosinci roku 2010

- Podpora protokolu SIP pro internetovou telefonii
- Nové Google Maps s 3D přístupem
- Upravená softwarová klávesnice
- Vylepšení funkce kopírovat/vložit

#### 1.1.6 Android 3.0/3.1/3.2

Verze 3.0 nesoucí název Honeycomb, byla představena v únoru 2011 a je určena výhradně pro tablety a přináší několik nových funkcí týkající se především uživatelského prostředí, které bylo nutno přepracovat pro potřeby větších displejů. Později následovaly verze 3.1 a 3.2 přinášející opravy chyb a několik menších vylepšení.

Některé nové funkce:

- Optimalizace pro velké obrazovky
- Nový design
- Přidány fragmenty
- Vylepšení výběru textu a funkcí kopírovat a vložit
- Vylepšení prohlížeče, kamery, galerie, emailu a kontaktů
- Přístup ke Google Books

### 1.1.7 Android 4.0

Verze 4.0 nese název Ice Cream Sandwich a byla poprvé představena v říjnu 2011. Tato verze měla přinést novou designovou filosofii a klíčové novinky z tabletového Honeycomb i na chytré telefony. Mimo to přišla i s vlastními novinkami, jako například:

- Odemykání telefonu obličejem
- Vylepšená práce s kontakty a sociálními sítěmi
- Vylepšení fotoaparátu
- Android Beam pro výměnu informací mezi dvěma zařízeními s NFC čipem
- Přepřelávaný launcher a další [2, s. 16-19]

### 1.1.8 Android 4.1/4.2/4.3

Android 4.1 (později 4.2 a 4.3) s názvem Jelly Bean byl poprvé představen v červenci 2012 a mezi novinky patří například:

- Projekt Butter pro zrychlení uživatelského prostředí
- Vylepšená notifikační lišta
- Více uživatelských účtů
- Možnost změny velikosti widgetů
- Vylepšená aplikace fotoaparátu a další... [4]

### 1.1.9 Android 4.4

Prozatím poslední verze (k březnu 2014) nese název KitKat a byla poprvé představena v září 2014. Mezi novinky patří například:

- Optimalizace OS pro zařízení s nižší kapacitou paměti RAM
- Podpora pro krokové senzory
- Průhledná notifikační lišta a lišta pro softwarová tlačítka
- Odstraněna podpora pro Flash
- Aplikace pro SMS zprávy nahrazena aplikací Hangouts [5]

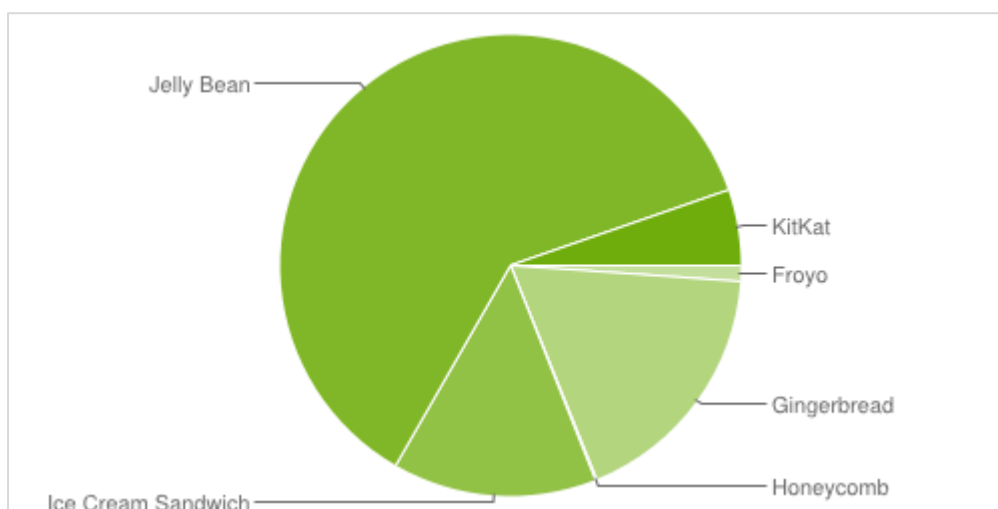
## 1.2 Roztříštěnost androidu

Velké rozdíly mezi konfiguracemi přístrojů, používajících OS Android, je největší slabinou tohoto systému. Především starší zařízení se slabší konfigurací již nepostačují pro novější verze Androidu a z toho plyne velká roztržitost tohoto systému. Tato roztržitost se

projevila především při příchodu Androidu 3.0 resp. 4.0, které přinesly mnoho radikálních změn. Mnoho starších zařízení tak dnes stále běží na verzi 2.3 Gingerbread a mnoho vývojářů Android aplikací tak musí dělat mnoho kompromisů (toto se však zlepšuje s příchodem Support library), nebo nevyvíjet aplikace kompatibilní se staršími zařízeními a zaměřit se pouze na vyšší verze Androidu.

Tab. 1. Přehled používaných verzí na Android zařízeních [6]

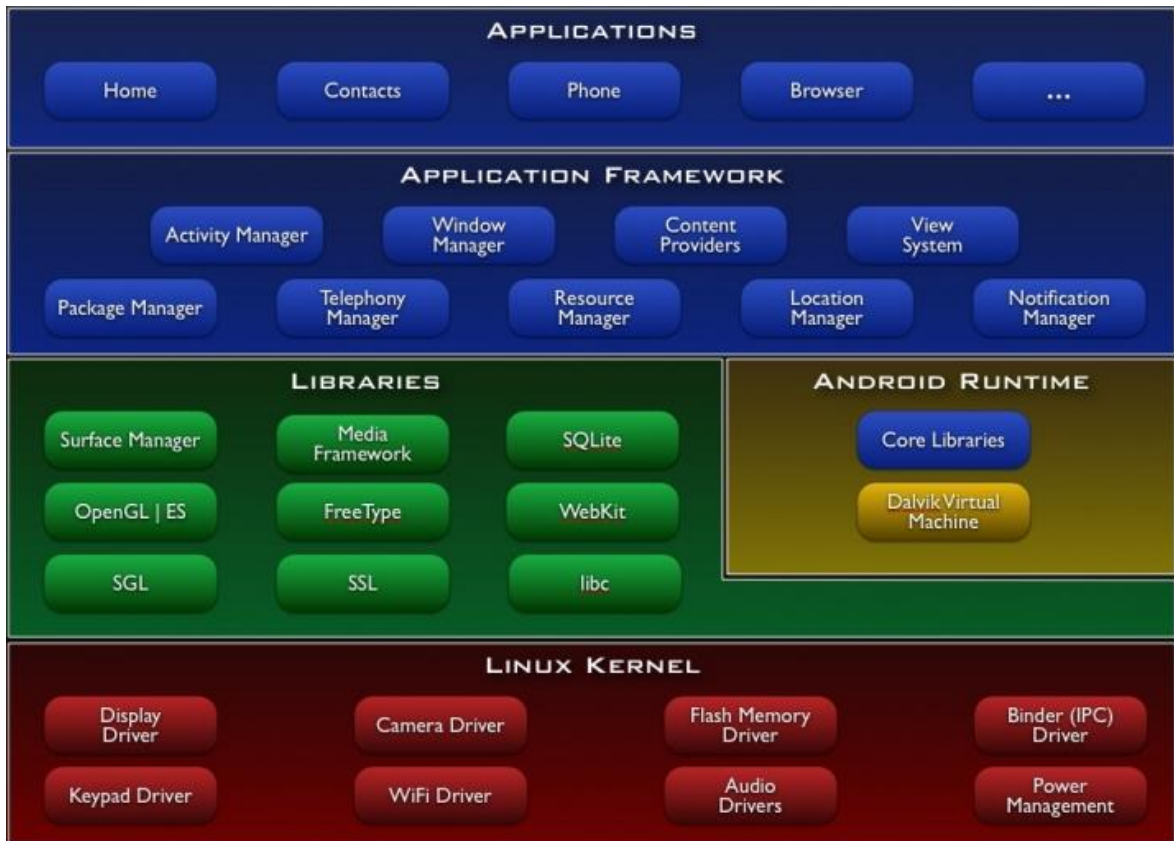
Verze	Název	Počet zařízení [%]
2.2	Froyo	1,1
2.3.3 - 2.3.7	Gingerbread	17,8
3.2	Honeycomb	0,1
4.0.3 – 4.0.4	Ice Cream Sandwich	14,3
4.1.x	Jelly Bean	34,4
4.2.x		18,1
4.3.x		8,9
4.4	KitKat	5,3



Obr. 1. Graf používaných verzí API na Android zařízeních (k 1. dubnu 2014) [6]

### 1.3 Architektura androidu

Architektura OS Android se skládá z pěti základních vrstev. Každá vrstva slouží k různým účelům a vystupují víceméně samostatně. V praxi však může docházet ke spolupráci jednotlivých částí vrstev a ty tak nejsou mezi sebou striktně odděleny. [2, s. 19]



Obr. 2. Architektura OS Android [7]

#### 1.3.1 Linux Kernel

Základní vrstvou je Linuxové jádro. Systém Android je postaven na Linuxovém jádru 2.6 s některými architektonickými změnami provedené firmou Google. Linuxové jádro se stará především o interakci s hardwarem zařízení a obsahuje všechny základní ovladače hardwaru. Pro příklad: většina zařízení obsahuje Bluetooth hardware a Linuxové jádro musí zprostředkovat komunikaci mezi tímto hardwarem a jeho ovladačem. Linuxové jádro taktéž implementuje abstrakci mezi hardwarem a dalšími softwarovými vrstvami. Android také používá Linux pro všechny procesorové funkcionality, jako je správa paměti, správa procesů, síťování, bezpečnostní nastavení atd. [7]

### 1.3.2 Libraries

Další vrstvou jsou nativní Android knihovny. Tato vrstva umožňuje zařízení manipulovat s různými typy dat. Tyto knihovny jsou psané v jazyce C nebo C++ a jsou specifické pro jednotlivé části hardwaru. Některé z důležitých knihoven: [7]

- **android.util** – hlavní balíček knihoven obsahuje nízko-úrovňové třídy pro specializované kontejnery, řetězcové formátovače, nebo nástroje pro parsování XML.
- **android.os** – knihovny poskytující přístup k základním službám operačního systému, jako je interprocesní komunikace, předávání zpráv, funkce hodin, atd.
- **android.graphics** – knihovny podporující vykreslování grafických prvků na obrazovku.
- **android.text** – knihovny obsahující nástroje pro zpracování a analýzu řetězců a jejich zobrazení.
- **android.content** – knihovny obsahující nízko-úrovňové třídy potřebné pro práci s kurzory v rámci databáze.
- **com.google.android.maps** – balíček knihoven, umožňující vývojáři přístup a ovládání aplikace Google Maps.
- **Android.app** – knihovny poskytující přístup k aplikačnímu modelu a obsahující API funkce aktivit a služeb, které tvoří základ pro všechny Android aplikace.
- **SQLite** – knihovna obsahující databázový model používaný v Androidu pro ukládání dat.
- **WebKit** – knihovny obsahující funkce pro práci s webovým obsahem.
- **OpenGL** – knihovna sloužící k podpoře 3D grafiky založené na OpenGL ES. [2, s. 20] [7]

### 1.3.3 Android runtime

Vrstva Android runtime se skládá z **Dalvik Virtual Machine (DVM)** a **Core Java Libraries**. DVM je typ Java Virtual Machine (JVM) – jedná se o Java virtuální stroj, používaný na Android zařízeních pro spouštění aplikací a je optimalizován pro nízký výpočetní výkon a nízké paměťové nároky. Na rozdíl od JVM nespouští soubory s příponou .class, ale místo toho spouští soubory s příponou .dex. Dex soubory jsou vytvářeny ze souborů .class při kompilaci a poskytují vyšší efektivitu v prostředí s nižším výpočetním výkonem. DVM umožňuje současné, vícenásobné vytváření instancí

virtuálního stroje, které poskytují zabezpečení, izolaci, správu paměti a podporu vláken. [7]

**Core Java Libraries** jsou knihovny, které poskytují většinu z funkcionalit Java SE (Standard Edition), jejímž základem jsou JVM, API knihovny základních funkcí a API knihovny pro vytváření desktopových aplikací (AWT, Swing), s tím rozdílem, že AWT a Swing byly u OS Android nahrazeny knihovnami pro tvorbu uživatelského rozhraní a přibyly také knihovny Apache pro práci se sítí. [2, s. 21] [7]

### 1.3.4 Application Framework

Tato vrstva umožňuje vývojářům přístup k různým službám, které poté mohou využívat ve svých aplikacích a ty jim poté umožní např. používat hardware zařízení, nastavovat alarmy, spouštět jiné aplikace atd. Některé důležité služby: [2, s. 21]

- **View System** – umožňuje používat pro tvorbu GUI prvky, jako textová pole, tlačítka, seznamy, přepínače, zaškrtačací políčka posuvníky atd.
- **Notification Manager** – umožňuje zobrazení notifikační lišty s vlastním upozorněním všem aplikacím.
- **Content Providers** – umožňuje přistoupit a pracovat s obsahem jiných aplikací, jako například kontakty, kalendáře atd.
- **Activity Manager** – ovládá životní cyklus aplikací, jejich start, průběh a ukončení.
- **Resource Manager** – poskytuje přístup ke zdrojům aplikace, jako jsou řetězce, grafika, barvy atd. [2, s. 21]

### 1.3.5 Applications

Poslední, nejvyšší vrstva představuje samotné aplikace, které jsou používány uživateli. Tyto aplikace je možné instalovat prostřednictvím online katalogu Google Play (oficiální). Nebo jsou již na zařízení předinstalované, jako například SMS klient, webový prohlížeč nebo správce kontaktů. [2, s. 21]

## 1.4 Distribuční síť Google Play

Google Play je online distribuční služba, která vznikla v roce 2012 a nahradila tehdejší Android Market. Google Play slouží primárně k distribuci aplikací pro OS Android a tato část se nazývá Google Play Store. Mimo to Google Play poskytuje další tři služby a sice: Google Play Music pro distribuci hudby, Google Play Books pro distribuci knih

v elektronické podobě a také Google Play Movies & TV pro distribuci filmů, která však není v České Republice zatím podporována.

## 2 ANDROID APLIKACE

Aplikace pro OS Android se programují v jazyce Java a fungují na principu nejnižších privilegií. To znamená, že standardně má aplikace přístup pouze k těm částem systému, které potřebuje pro svoji práci. To vytváří relativně bezpečné prostředí, ve kterém aplikace nemá přístup k částem systému, ke kterým jí nebyla udělena oprávnění. [8]

- OS Android je více-uživatelský Linuxový systém, ve kterém každá aplikace představuje jiného uživatele.
- Standardně přiděluje systém každé aplikaci unikátní uživatelské ID (ID je používáno pouze systémem a aplikace jej nezná). Systém nastavuje oprávnění pro všechny soubory v aplikaci tak, že pouze aplikace s odpovídajícím uživatelským ID má k těmto souborům přístup.
- Každý proces má svůj vlastní virtuální stroj – virtual machine (VM), tudíž aplikační kód běží izolovaně od ostatních aplikací.
- Standardně běží každá Android aplikace ve vlastním Linuxovém procesu. Android nastartuje proces, jestliže některá z komponent aplikace potřebuje být vykonána a ukončuje jej, pokud byl proces úspěšně proveden, nebo pokud je potřeba uvolnit paměť pro ostatní aplikace. [8]

Nicméně, existují způsoby, jak může více aplikací sdílet stejná data, nebo jak povolit aplikaci přístup k systémovým službám. Aplikace může například požadovat přístup ke kontaktům, SMS zprávám, kameře, bluetooth, a dalším. Tyto požadavky však musí být odsouhlaseny uživatelem před instalací aplikace. [8]

### 2.1 Vývojové prostředky

Vývoj aplikací pro Android, stejně jako vývoj většiny aplikací pro mobilní zařízení, probíhá v tzv. Host-Target vývojovém prostředí. To znamená, že zařízení, na kterém je aplikace vyvíjena, je odlišné od toho, pro které je aplikace určena. V případě android aplikací probíhá vývoj převážně na osobním počítači, který obsahuje vývojové prostředí a samotná aplikace je používána na mobilním zařízení. Za použití emulátoru je možné simulovat prostředí OS Android i na počítači, a spouštět zde Android aplikace. Toto se však používá výhradně k testovacím účelům. [2, s. 25]

Pro vývoj Android aplikací jsou potřebné tyto nástroje:

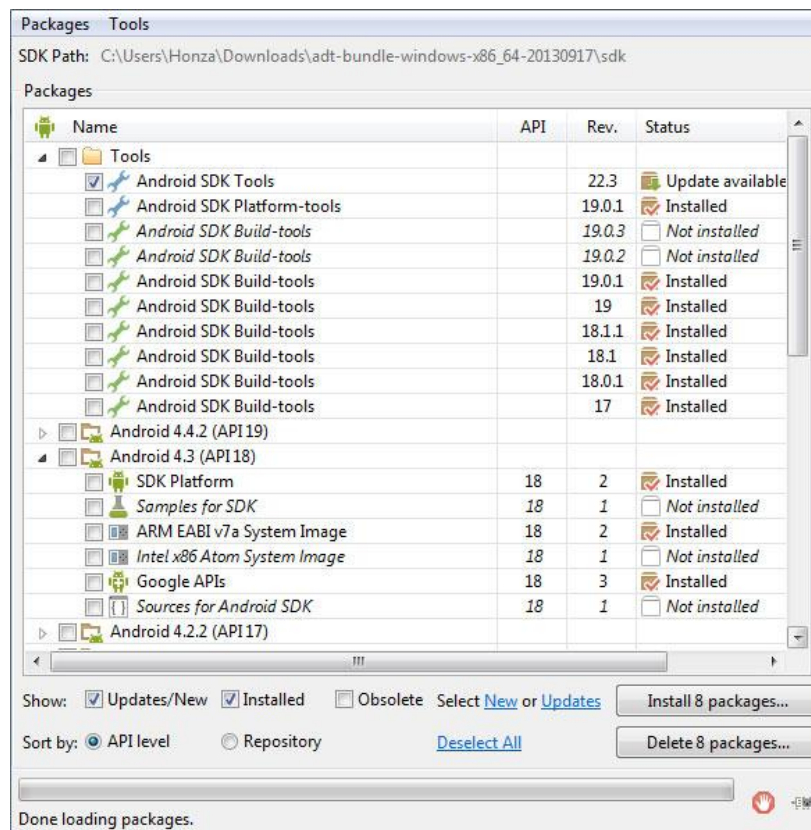
- Java Development Kit (JDK)
- Android Software Development Kit (SDK)
- Vývojové prostředí Eclipse
- Android Development Tool (ADT)

### 2.1.1 Java Development Kit

JDK je soubor základních nástrojů a knihoven, které slouží k vývoji aplikací pro platformu Java. Základní součástí JDK je Java Runtime Environment (JRE), jež slouží ke spuštění aplikací, obsahuje překladač, debugger a další. [2, s. 25]

### 2.1.2 Android Software Development Kit

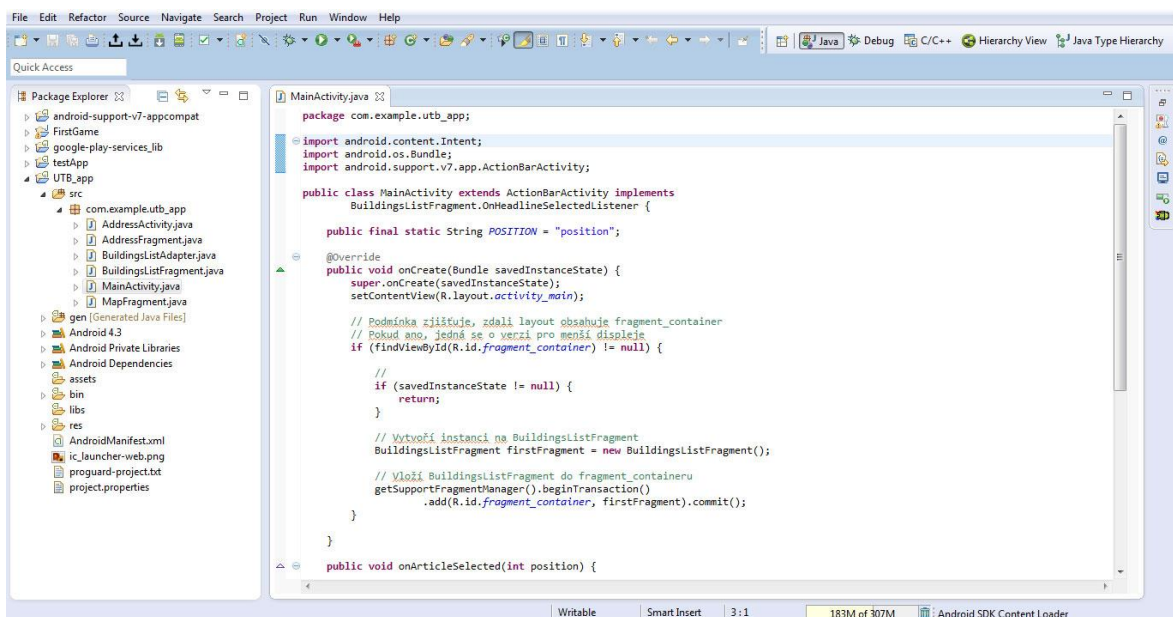
Android SDK poskytuje API knihovny a vývojové nástroje, nezbytné pro vývoj, testování a ladění aplikací pro Android. Obsahuje také emulátor, který umožní vytvořit virtuální mobilní zařízení s OS Android na počítači. Další důležitou součástí je **SDK Manager**, který umožňuje stahování aktuálních API. [2, s. 26-30]



Obr. 3. SDK Manager.

### 2.1.3 Eclipse

Eclipse je vývojové prostředí (IDE) určené primárně k programování v jazyce Java. Výhodou Eclipse oproti jiným prostředím je především snadná rozšiřitelnost o podporu jiných programovacích jazyků nebo o různé nástroje pro tvorbu GUI. Eclipse je také oficiálně podporovaným prostředím pro vývoj Android aplikací. K vývoji aplikací pro Android lze však použít jakékoliv jiné prostředí umožňující programovat v Javě, jako například NetBeans, Oracle JDeveloper, BlueJ a jiné, nicméně použití Eclipse, představuje nejjednodušší a nejpohodlnější cestu pro vývoj Android aplikací. [2, s. 31-32]



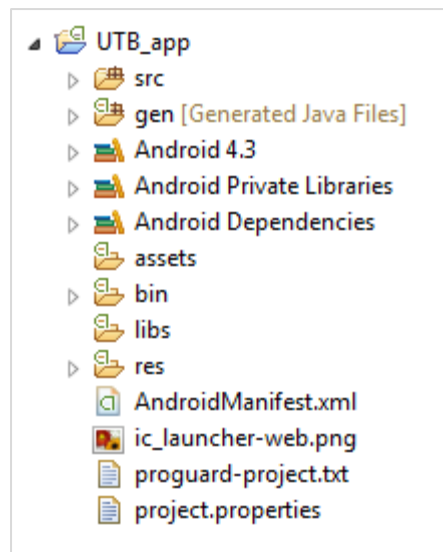
Obr. 4. Vývojové prostředí Eclipse

### 2.1.4 Android development tool

Aby bylo možné pohodlně vyvíjet Android aplikace v prostředí Eclipse, je nutné propojit jej s Android SDK. Toto propojení je reprezentováno pluginem ADT, jež rozšiřuje Eclipse o sadu nástrojů, které usnadňují vývoj Android aplikací. [2, s. 34]

## 2.2 Adresářová struktura

Po založení nového projektu v prostředí Eclipse se v kořenovém adresáři automaticky vytvoří několik dalších podadresářů. Tyto podadresáře slouží k uchování zdrojových kódů aplikace, knihoven potřebných pro správný chod aplikace a také k ukládání tzv. resources (prostředků). Po založení projektu se také automaticky vygeneruje několik souborů. Jedná se především o soubory *AndroidManifest.xml* a *project.properties*.



Obr. 5. Adresářová struktura android projektu

### 2.2.1 Popis adresářů v novém Android projektu

- **src** – tento adresář obsahuje zdrojové kódy aplikace v jazyce Java.
- **gen** – jedná se o adresář, ve kterém se generují zdrojové soubory jazyka Java, vygenerované pomocí nástrojů OS Android, sloužících pro překlad jazyka.
- **Android 4.3** – tento adresář obsahuje knihovny, které jsou potřeba pro správnou funkčnost projektu. Jméno složky vychází z názvu a verze OS, na kterém jsme se rozhodli vyvíjet.
- **assets** – adresář slouží k uchování externích souborů, jež jsou potřebné v aplikaci, jako například obrázky, hudba, video. Od prostředků v adresáři *res* se liší tím, že jsou uloženy v tzv. surovém formátu a přístup k nim je možný pouze programově.
- **Bin** – tento adresář uchovává aplikaci po jejím zkompileování.
- **Res** – adresář sloužící k uchování externích prostředků, jež jsou v aplikaci, např. GUI (Grafické uživatelské prostředí), textové řetězce, obrázky atd. [2, s. 56]

### 2.3 Komponenty aplikace

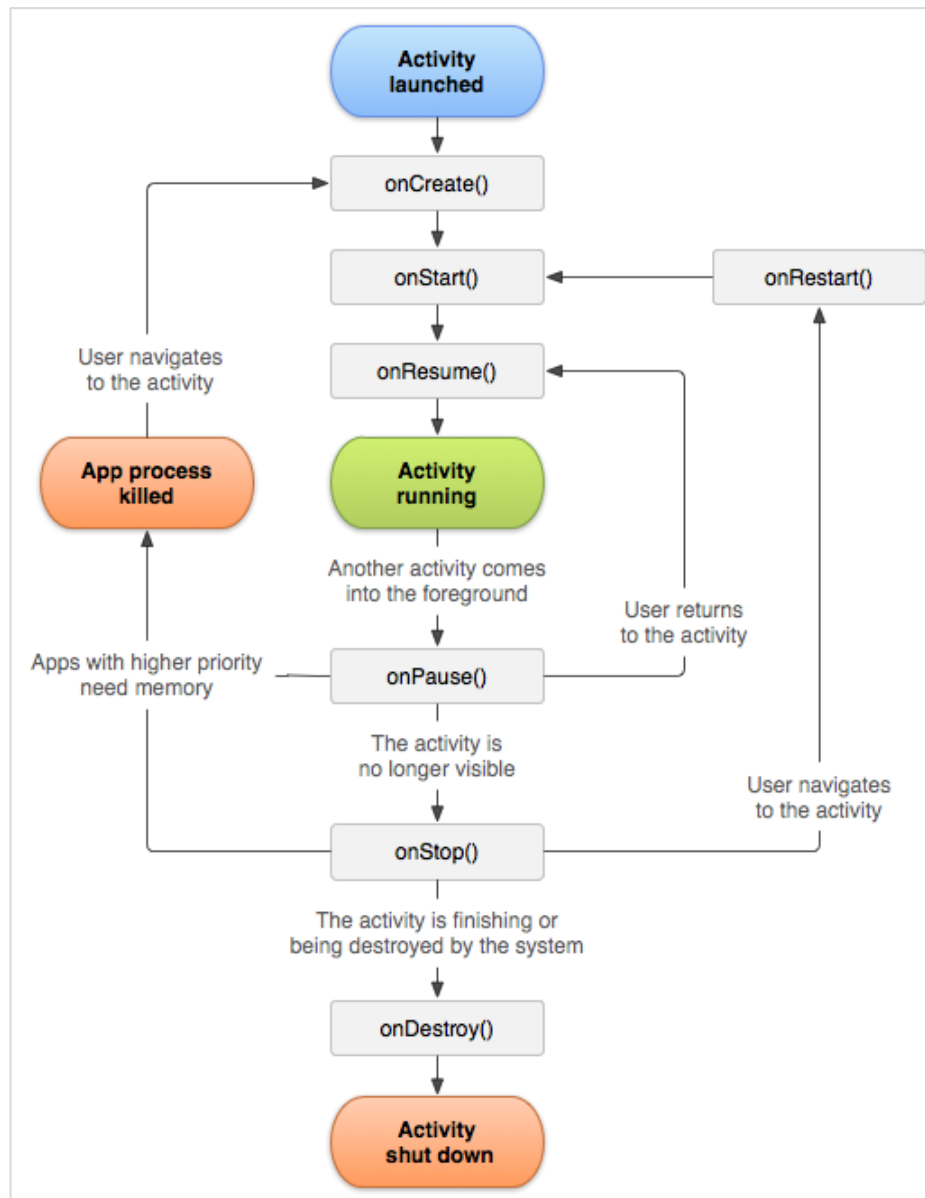
Základní stavební kameny android aplikací se nazývají komponenty aplikace. Patří mezi ně tyto čtyři typy: Activity, Service, Broadcast Reciver a Content Provider. Každý z těchto typů slouží k jinému účelu a má svůj vlastní životní cyklus.

### 2.3.1 Activity

Activity je komponenta, která umožňuje uživateli interakci s aplikací prostřednictvím uživatelského rozhraní. Aplikace se většinou skládá z několika Activity, které jsou vzájemně propojené. Jedna Activity může spustit jinou. Předchozí Activity je zastavena a nově spuštěná Activity je umístěna na zásobník „back stack“ (BS). Tento zásobník funguje na principu LIFO (poslední dovnitř, první ven) a slouží k návratu na předchozí Activity. Jeli například stisknuto zpětné tlačítko na android zařízení, současná Activity je vyjmuta ze zásobníku (a ukončena) a pokračuje se předchozí Activity. [9]

Každá Activity má svůj vlastní životní cyklus. Dojde-li ke změně stavu Activity, např. je-li vytvořena, pozastavena, zastavena, ukončena atd., dojde k vyvolání příslušné callback metody. Tyto metody umožňují naprogramovat adekvátní chování Activity při změně jejich stavu. [9]

- **onCreate** – tato metoda se volá při vytvoření Activity.
- **onStart** – tato metoda se volá bezprostředně po metodě onCreate a také pokud dojde k zastavení aktivity metodou onStop a následném znovu spuštění Activity.
- **onResume** – volá se tehdy, dostane-li se pozastavená Activity z viditelného pozadí, zpět do popředí.
- **onPause** – tato metoda je vyvolána tehdy, když se do popředí dostane jiná Activity, ale původní aktivita je stále, alespoň částečně, viditelná. Dostane-li se původní Activity opět do popředí, je volána metoda onResume.
- **onStop** – tato metoda se volá při zastavení Activity. K tomu dochází tehdy, jeli aktivita zcela neviditelná na zařízení, například přepnutí do jiné aplikace. Taková Activity stále zabírá místo v paměti.
- **onDestroy** – tato metoda se volá při ukončení Activity, buď uživatelem, nebo systémem kvůli nedostatku paměti. [9]



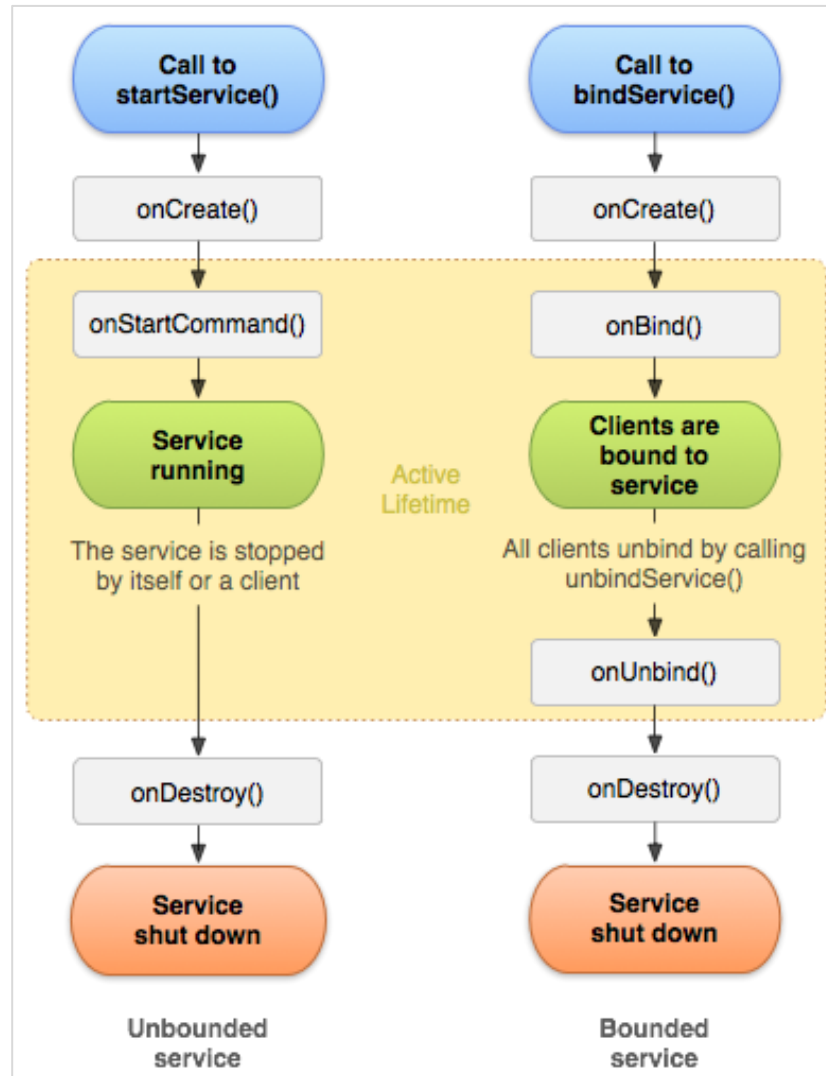
Obr. 6. Životní cyklus Activity [9]

### 2.3.2 Service

Service neboli služba je komponenta aplikace, která provádí déle běžící operace na pozadí a neposkytuje žádné uživatelské rozhraní. Službu je možné spustit pomocí jiné komponenty. Je-li jednou spuštěna, pokračuje v práci, i když uživatel právě používá jinou aplikaci. Služby mohou například provádět síťové transakce, přehrávat hudbu, provádět vstupně/výstupní operace se soubory nebo komunikovat s Content providery a to vše na pozadí. [10]

Služby se vyskytují ve dvou formách:

1. Služba je **started** jestliže je spuštěna jinou komponentou voláním metody *startService()*. Jakmile je služba spuštěna, běží již nezávisle na této komponentě. Může běžet, i když byla komponenta již ukončena. Tyto služby se používají především k vykonání jednorázových operací a zpravidla nic nevracejí. Mohou se používat např. pro stahování souborů přes síť. Tyto služby se většinou zastavují samy, jakmile dokončí svou práci. [10]
2. **Bound** jsou takové služby, které jsou k nějaké komponentě navázané metodou *bindService()*. Tyto služby nabízí rozhraní klient-server, které umožňuje interakci ostatních komponent se službou, posílat jim požadavky a přijímat výsledky. Dojde-li k ukončení komponenty, na kterou je služba navázána, dojde i k ukončení této služby. [10]



Obr. 7. Rozdíl mezi životními cykly started a bound služby a jejich vztah k životnímu cyklu Activity [10]

### 2.3.3 Content providers

Content providery spravují sdílené části dat aplikace. Data aplikace je možno uchovávat v souborovém systému, v SQLite databázi, na webu atd. Ostatní aplikace mohou skrz Content provider k těmto datům přistupovat nebo je dokonce modifikovat (pokud to Content provider povolí). Systém Android například poskytuje Content provider, který spravuje kontakty uživatele. [8]

### 2.3.4 Broadcast receivers

Broadcast receiver je komponenta, která zodpovídá za rozesílání systémových zpráv, tzv. broadcastů. Například broadcast oznamující, že byla vypnuta obrazovka, že je nízká úroveň nabití akumulátoru nebo že byla pořízena fotografie. Aplikace samotné také mohou

iniciovat broadcast – např. oznámit ostatním aplikacím, že byla stažena nějaká data a ty jsou nyní k dispozici. Přestože Broadcast recivery nedisponují uživatelským rozhraním, mohou vytvářet notifikace, které se zobrazí v notifikační liště. Každý broadcast je doručován jako Intent objekt. [8]

### 2.3.5 Intenty

Intent je „zpravující“ objekt, který se používá, jestliže požadujeme provést nějakou akci z jiné aplikační komponenty. Rozlišujeme Intenty **implicitní** a Intenty **explicitní**. Intenty lze používat k více účelům, nicméně hlavní jsou tyto tři:

1. Pro **nastartování aktivity**. Aktivita představuje jednu obrazovku na android zařízení. Novou aktivitu je možné nastartovat voláním metody `startActivity()` a přidáním Intentu jako jejího parametru. Intent popisuje aktivitu, která se má spustit a může nést i data potřebná při startu nové aktivity.
2. Pro **nastartování služby**. Služby je možné spustit pomocí metody `startService()` pro jednorázové operace běžící na pozadí s Intentem jako parametrem. Tento Intent charakterizuje službu, která se má spustit a nese veškerá nezbytná data.
3. Pro **doručení Broadcastu**. Broadcast je zpráva, kterou může obdržet kterákoliv aplikace. Systém doručuje různé broadcasty pro různé systémové události jako např. při bootování systému nebo při připojení nabíječky. Broadcasty je možné doručovat různým aplikacím za použití Intentu jako parametru metody `sendBroadcast()` eventuálně `sendOrderBroadcast()` nebo `sendStickyBroadcast()`. [11]

#### 2.3.5.1 Explicitní intenty

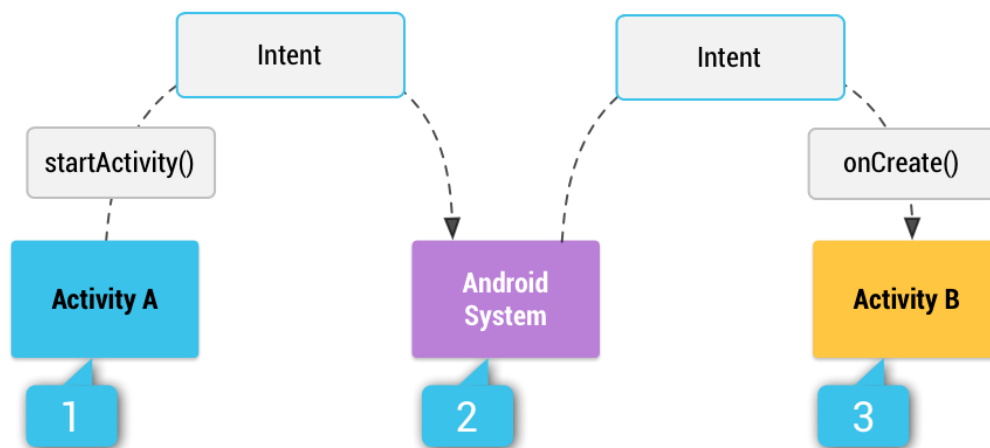
Určují komponentu, která se má spustit, podle jména (úplného jména třídy). Explicitní intenty se využívají především v rámci jedné aplikace, protože jsou jména jednotlivých komponent známá. Příkladem pro použití explicitního intentu je například spuštění služby nebo aktivity. [11]

#### 2.3.5.2 Implicitní intenty

Implicitní intenty nespecifikují komponenty, ale místo toho deklarují obecné činnosti, které se mají vykonat. To umožňuje komponentám z jiných aplikací nějakým způsobem na ně reagovat. Chceme-li, například ukázat uživateli nějakou polohu na mapě a naše aplikace

toto neumožňuje, použijeme implicitní Intent s požadavkem aby nějaká jiná kompetentní aplikace toto provedla.

Při vytváření implicitního Intentu, systém Android hledá odpovídající komponentu tak, že porovnává obsah Intentu s Intent filtry deklarovanými v souboru Manifest ostatních aplikací v zařízení. Je-li nalezena shoda, systém nashoduje příslušnou komponentu a poskytne Intent jako objekt. [11]

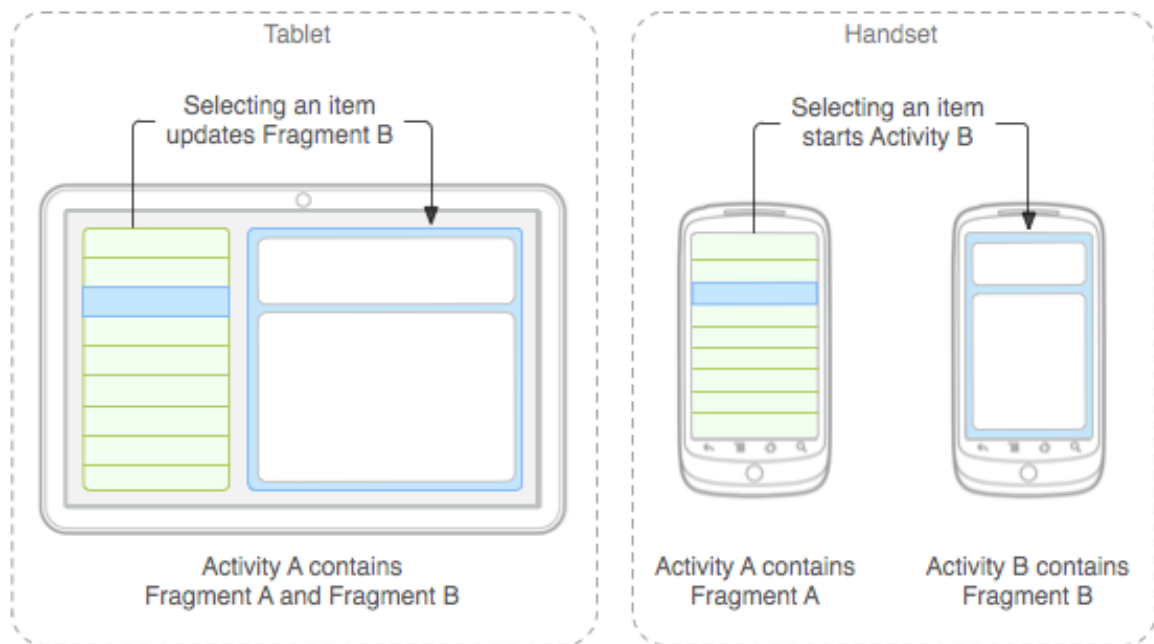


Obr. 8. Intenty [10]

### 2.3.6 Fragmenty

Fragmenty se normálně mezi základní aplikační komponenty neřadí, jelikož jsou výhradně součástí aktivit. Nicméně, s rozmachem android zařízení s většími displeji, především tabletů, nabývá tato komponenta na důležitosti. Jelikož budou důležité i z hlediska naší aplikace, tak si je zde dovolím uvést. Fragmenty se poprvé objevili s Androidem 3.0 ale při použití Support Library se dají fragmenty používat zpětně až po verzi 1.6.

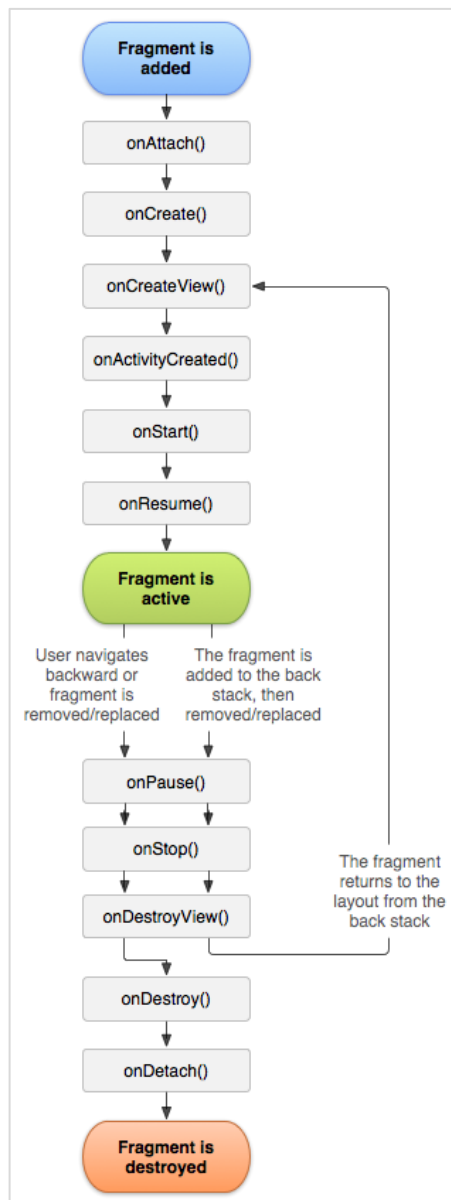
Fragmenty umožňují rozdělení uživatelského rozhraní v rámci jedné aktivity. Za použití fragmentů lze například vytvořit více panelové rozhraní, kde jeden fragment představuje navigační část, a další fragmenty mohou představovat část obsahovou. Toto se hodí především pro zařízení s většími displeji, jako jsou tablety, kde by rozdělení navigační a obsahové části na dvě obrazovky, znamenalo nevyužití velkého prostoru. Další výhodou fragmentů je, že se dají znovupoužít v kterémkoliv místě aplikace. [12]



Obr. 9. Příklad použití fragmentů na různých typech zařízení [12]

Fragmenty mají vlastní životní cyklus, podobný jako aktivity. Oproti aktivitám ale obsahuje některé další metody.

- **onAttach** – volá v momentě, kdy je Fragment poprvé připojen k Activity, poté se volá metoda **onCreate**
- **onCreate** – tato metoda se volá při vytvoření Fragmentu.
- **onCreateView** – volá se při vytváření uživatelského rozhraní. Tato metoda vrací objekt *View* popřípadě *null* u Fragmentů bez uživatelského rozhraní.
- **onActivityCreated** – volá se tehdy, když Activity, ke které je Fragment připojen dokončí svou vlastní metodu **onCreate**
- **onStart** – po zavolání této metody se Fragment stává viditelným pro uživatele
- **onResume** – volá se tehdy, dostane-li se pozastavená Activity, ke které je Fragment připojen, z viditelného pozadí, zpět do popředí.
- **onPause**, **onStop** – tyto metody jsou volány tehdy, jestliže se zavolají stejnojmenné metody v Activity, ke které je fragment připojen.
- **onDestroyView** – umožňuje Fragmentu vyčistit zdroje spojené s jeho *view*.
- **onDestroy** – tato metoda se volá při ukončení Fragmentu.
- **onDetach** – volá se tehdy, když Fragment již není součástí Activity [13]



Obr. 10. Životní cyklus Fragmentu. [12]

## 2.4 Prostředky aplikace

Android aplikace se mimo zdrojového kódu skládá z mnoha dalších souborů, jako například obrázků, audio souborů a především XML souborů, které definují layouty komponent s GUI, barvy, textové řetězce, styly a mnoho dalších. Jak už bylo řečeno výše, tyto soubory se ukládají do adresáře `res` v kořenovém adresáři.

Velká výhoda těchto prostředků, oddělených od zdrojového kódu, je schopnost poskytnout alternativní prostředky pro zařízení různých konfigurací. Pro příklad uveďme soubor `strings.xml`, který uchovává textové řetězce. Tento soubor se standardně nachází v adresáři

*res/values*. Vytvoříme-li nový adresář přidáním kvalifikátoru *-cs* vznikne adresář *res/values-cs*. Přidáme-li do něj soubor *strings.xml* s českými textovými řetězci, veškerá zařízení s českým jazykovým nastavením budou čerpat textové řetězce ze souboru *res/values-cs/strings.xml* namísto *res/values/strings.xml*. Mimo jazykových mutací se kvalifikátory využívají především pro adresář *layouts*, kde jsou uloženy šablony GUI, a pro adresář *drawable*, kde se díky kvalifikátorům mohou ukládat obrázky s různým rozlišením a zařízení s malým rozlišením obrazovky tak nemusí načítat obrázky s velkým rozlišením. [8]

#### 2.4.1 Přehled jednotlivých prostředků

- **drawable** – do tohoto adresáře se ukládají obrázky nebo XML soubory.
- **layout** – v tomto adresáři jsou umístěny XML soubory, které definují grafické rozložení jednotlivých komponent aplikace nebo widgetů.
- **colors** – XML soubor, ve kterém jsou definovány jednotlivé barvy, které jsou v aplikaci využívány.
- **strings** – XML soubor s textovými řetězci.
- **styles** – XML soubor s definicemi jednotlivých stylů, například výchozí barva pozadí celé aplikace, nebo pozadí navigační lišty.

## 2.5 Widgety

OS Android obsahuje sadu nástrojů pro tvorbu GUI, která obsahuje různé vstupně/výstupní prvky. Tyto prvky se nazývají widgety, a patří mezi ně například textová pole, popisky, tlačítka, seznamy atd. Tyto widgety se definují buď v XML šablonách v adresáři *res/layout* nebo je možné je zapsat přímo ve zdrojovém kódu. Dovolím si zde uvést několik základních widgetů, které se často využívají v OS Android. [14, s. 93]

### 2.5.1 TextView

Tento jednoduchý widget představuje textové pole a má za úkol zobrazovat textové řetězce. V OS Android je reprezentován třídou `TextView`.

### 2.5.2 Button

Widget `Button` se používá k definování jednoduchých tlačítek. Na tyto tlačítka lze aplikovat tzv. naslouchač, pomocí něhož lze reagovat na stisknutí tlačítka uživatelem a

provést nějakou akci. Tento nasloucháč je realizován ve zdrojovém kódu metodou *OnClickListener()*.

### 2.5.3 ImageView, ImageButton

Tyto widgety slouží pro zobrazování obrázků. ImageButton navíc umožňuje přiřadit nasloucháč, podobně jako u widgetu Button a reagovat na stisknutí obrázku uživatelem.



*Obr. 11. Ukázka Widgetů Button a ImageButton*

[15]

### 2.5.4 ListView

Jeden z nejdůležitějších widgetů systému Android a velice často používaný. Slouží k vytváření seznamu. Seznam může být jednoduchý výpis textových řetězců, ale také lze na tento seznam navázat nasloucháč a může sloužit jako navigační menu. Navíc lze takovému seznamu přiřadit adaptér, který se stará o vykreslování vzhledu jednotlivých řádků. Řádek tak může obsahovat obrázky, několik řádků textu s různými styly atd.

### 2.5.5 Tabs

Tabs jsou přepínače, které bývají většinou umístěny v horní části obrazovky. Slouží k přepínání mezi několika obrazovkami. Tyto přepínače mohou být fixní nebo rolovací.



*Obr. 12. Ukázka widgetu Tabs [16]*

## 2.6 Soubor AndroidManifest.xml

Soubor Android Manifest je nutnou součástí každé Android aplikace. Tento soubor je psán v jazyce XML a je umístěn v kořenovém adresáři aplikace. Používá se především pro deklaraci komponent aplikace, deklaraci minimální, doporučené a základní API, požadavky na oprávnění k přístupu k internetu, GPS, a další.

Primárním úkolem Android Manifestu je informovat systém o komponentách aplikace. Před tím než systém aplikaci spustí, nahlédne do souboru AndroidManifest.xml, jestli je tato komponenta deklarována. Pokud nejsou deklarovány Aktivity, Služby nebo Content providery, nejsou pro systém viditelné a pokus o jejich spuštění pravděpodobně skončí pádem aplikace. Výjimkou jsou Broadcast receivers, které lze vytvářet dynamicky v kódu (voláním metody *registerReceiver()*).

Další důležitou úlohou Android Manifestu, je deklarace požadovaných verzí API a požadavků na hardwarové vybavení zařízení. Většina z těchto deklarací jsou informativní a systém je přehlíží. Důležité jsou však pro distribuční síť aplikací Google Play, která podle těchto deklarací filtruje aplikace pro uživatele, kteří chtějí tyto aplikace stahovat. Disponuje-li například nějaký uživatel API verzí 2.1 a aplikace má v souboru Android Manifest deklarovanu minimální verzi API 2.2, tato aplikace se uživateli na Google Play nezobrazí ke stažení. [8]

## **II. PRAKTICKÁ ČÁST**

### 3 PŘÍPRAVA PODKLADŮ A NÁVRH APLIKACE

Předtím, než se začnu věnovat samotnému vývoji aplikace, je nutné si ujasnit, pro jaká zařízení budu aplikaci vyvíjet, pro koho bude aplikace určena a zvolit vhodnou cestu samotného vývoje. Také si v této části předpřipravím podklady pro aplikaci, jako jsou adresy, kontakty a také zeměpisné souřadnice pro navigační modul.

#### 3.1 Návrh aplikaci

Při návrhu aplikace jsem se zaměřil na dvě základní otázky. První otázkou bylo: **Pro koho bude aplikace určena?** Využití aplikace vidím především u studentů, nastupujících do prvních ročníků, nepocházejících ze Zlína. Aplikace by tedy měla být jednoduchá a přehledná. Aplikace by měla podle zadání obsahovat adresy budov a navigační modul, tedy mapu s vyznačenou polohou těchto budov. Z vlastní zkušenosti však vím, že i to někdy nestačí, a proto jsem se rozhodl v aplikaci použít i obrázky těchto budov a kontakty, které mají na tyto budovy nějakou vazbu.

Druhou, pro mě důležitou otázkou bylo: **Pro jaká zařízení bude aplikace určena?** OS Android v dnešní době využívá mnoho typů zařízení a mnoho jich pravděpodobně přibude v budoucnu. Z logiky samotného zaměření aplikace však vyplývá, že aplikace bude používána především na mobilních zařízeních. Proto jsem při vývoji aplikace zaměřil na chytré telefony a tablety.

I když tabletů s OS Android rok od roku přibývá a v dnešní době se drží celosvětově na prvním místě v počtu aktivních zařízení, stále je podle mého názoru málo aplikací, které jsou optimalizované pro tablety a jiná zařízení s větším displejem. Proto jsem se rozhodl zaměřit na tuto problematiku a při vývoji aplikace jsem vycházel z konceptu dvou různých grafických rozložení za použití fragmentů: pro chytré telefony a pro tablety.

##### 3.1.1 Shrnutí základních požadavků

- Jednoduché a přívětivé grafické prostředí
- Rozdělení grafického prostředí pro menší a větší obrazovky za použití fragmentů
- Přehledný seznam jednotlivých budov
- Zobrazení adres
- Zobrazení kontaktů a nabídka jejich využití pomocí vhodné aplikace
- Zobrazení obrázků budov

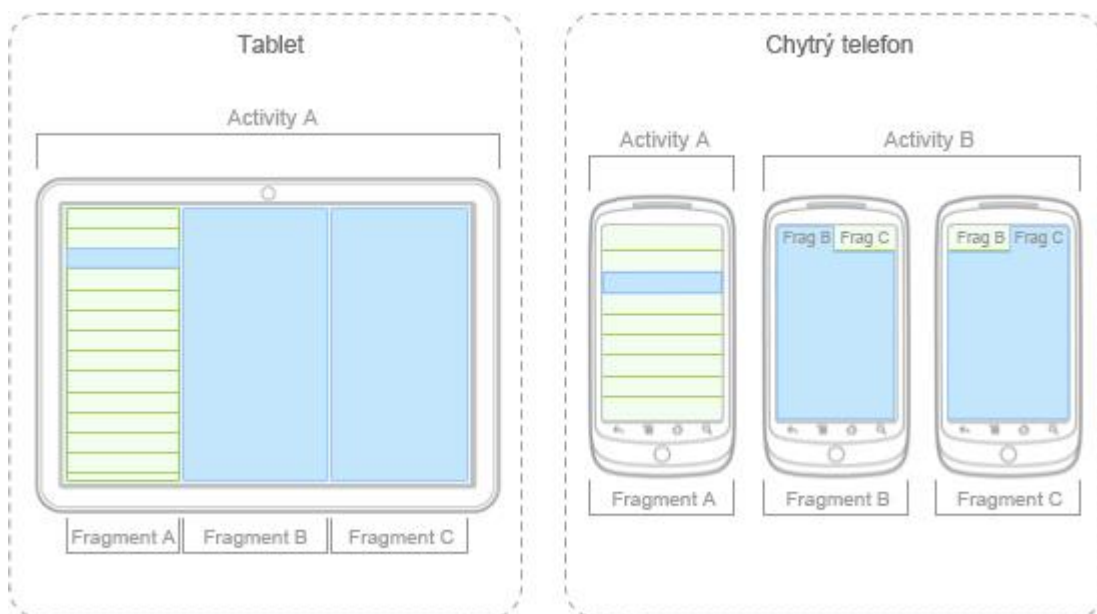
- Použití map s vyznačenou polohou budov a aktuální polohou uživatele, pokud to zařízení umožní

### 3.1.2 Struktura aplikace

Při promyšlení počtu komponent a jejich provázanosti jsem vycházel z návrhu na *Obr. 9. Příklad použití fragmentů na různých typech zařízení*. Na tomto obrázku jsou znázorněny dvě Activities a dva Fragments. V našem případě půjde o tři Fragments, protože Fragment, zobrazující obsahovou část, rozdělím na Fragment obsahující obrázek budovy, adresu a kontakty a na Fragment s navigačním modulem.

V případě mobilní verze bude první Activity „obalovat“ navigační Fragment a při výběru nějaké položky spustí druhou Activity, která bude obalovat Fragment s informacemi a Fragment s navigačním modulem a obsluhovat přepínání mezi nimi.

V případě tabletové verze bude první Activity obalovat všechny tři fragmenty a zároveň je aktualizovat.

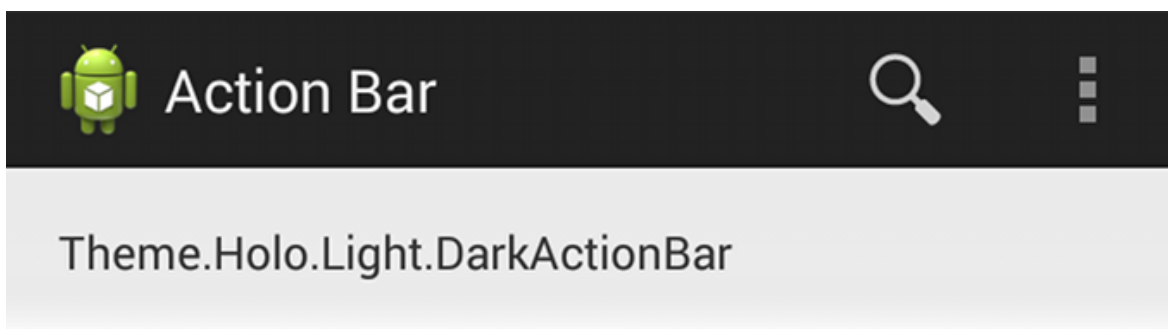


*Obr. 13. Návrh struktury aplikace*

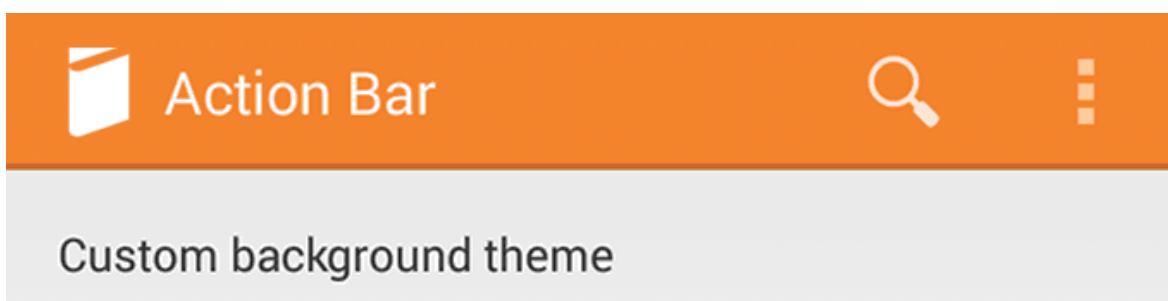
### 3.1.3 Grafický návrh

Při návrhu vzhledu aplikace jsem vycházel z požadavku na jednoduchost a aktuálních designových trendů Android aplikací. Jako základ jsem zvolil oficiální téma s názvem Holo Light s tmavým Action Barem (lišta umístěná v horní části obrazovky), dále jen AB.

Pozadí AB jsem změnil na oranžové, reprezentující oficiální barvu UTB ve Zlíně a také jsem použil logo univerzity v bílé barvě.



*Obr. 14. Základní grafické téma Holo Light DarkActionBar*



*Obr. 15. Návrh základního vzhledu aplikace*

### 3.2 Příprava podkladů

Určil jsem čtyři důležité informace (+ obrázek) pro každou budovu a sice: Název budovy, adresa budovy, kontakty spojené s budovou a souřadnice budovy (hodnoty zeměpisné šířky a délky). Tabulka vypracovaných informací o budovách je umístěna v příloze PI.

## 4 VÝVOJ APLIKACE

### 4.1 Použité vývojové prostředky

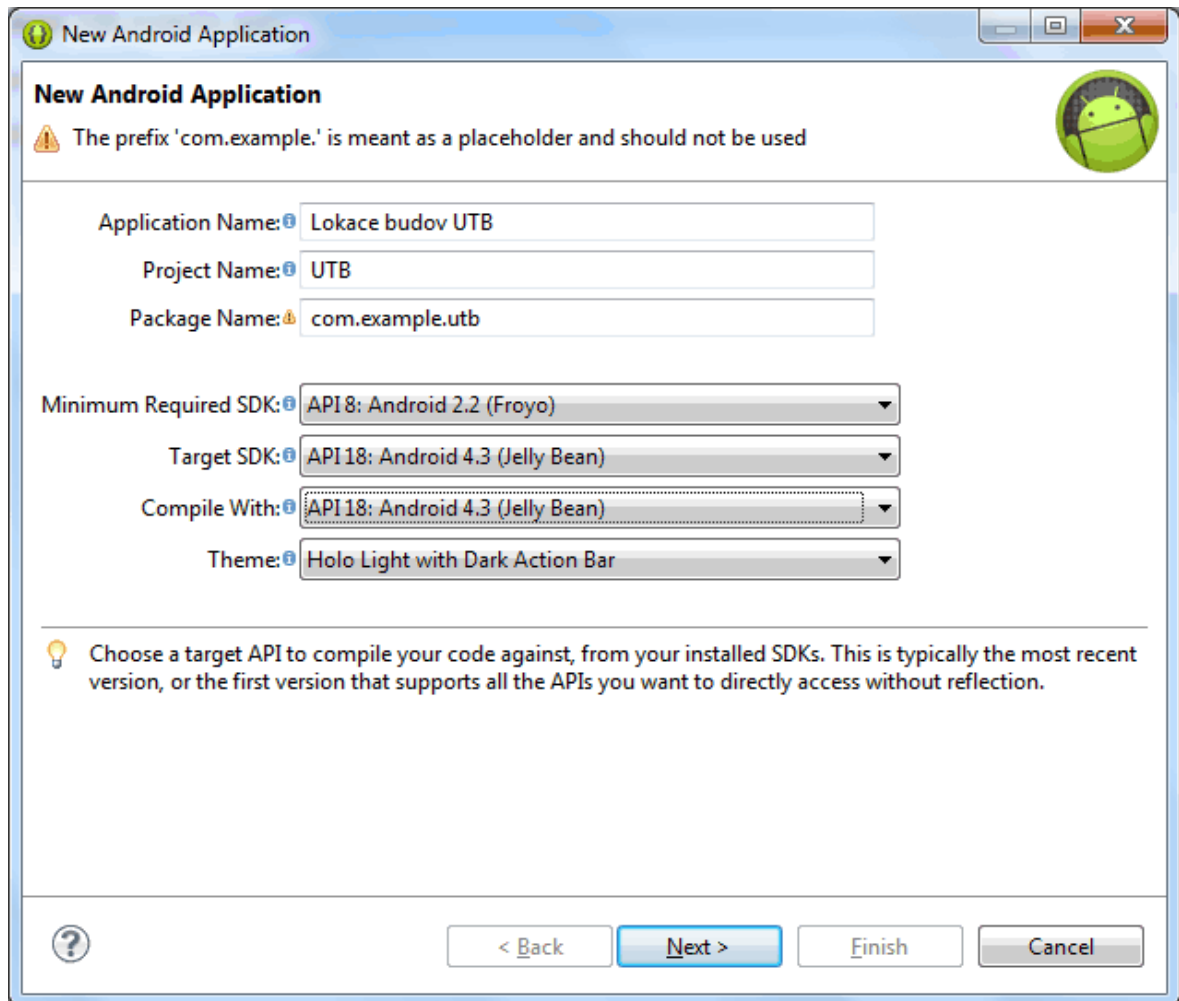
Pro vývoj aplikace jsem se rozhodl použít oficiálně podporované vývojové prostředí Eclipse s nadstavbou ADT. Pro jednoduchost jsem zvolil verzi Eclipse s již zabudovaným ADT, kterou je možné stáhnout z oficiálních stránek Android Developers. Po spuštění Eclipse ještě bylo nutné stáhnout knihovny k aktuálnímu API a některé další nástroje pro vývoj.

### 4.2 Založení projektu

V okamžiku, kdy jsem měl k dispozici vše potřebné pro vývoj aplikace, přišlo na řadu založení nového Android projektu. To jsem provedl v prostředí Eclipse, pomocí *File*→*New*→*Android Application Project*. Otevřelo se mi dialogové okno, kde jsem vyplnil *jméno projektu*, *jméno aplikace* a *jméno balíčku (package name)*, což je obdoba jmenných prostorů. Poté bylo nutné vybrat *minimální SDK*, pod kterým bude možné aplikaci spustit, *cílové* a *doporučené SDK* a také základní grafické schéma. Všechny tyto atributy jsem zvolil následovně:

- **Jméno aplikace:** Lokace budov UTB
- **Jméno projektu:** UTB\_app
- **Jméno balíčku:** com.example.utb\_app
- **Minimální požadované SDK:** API 8: Android 2.2 (Froyo)
- **Cílové SDK:** API 18: Android 4.3 (Jelly Bean)
- **Doporučené SDK:** API 18: Android 4.3 (Jelly Bean)
- **Schéma:** Holo Light with Dark Action Bar

V dalším dialogovém okně jsem zvolil možnost *Create Activity*, která nám po dokončení zakládání projektu vytvoří aktivitu, jenž bude sloužit, jako aktivita hlavní (spouštěcí) a poté také zvolil adresář workspace. V dalším dialogovém okně jsem si mohl vybrat rozložení základní aktivity. Zvolil jsem *blank activity* (prázdnou aktivitu). V posledním dialogovém okně jsem byl vyzván k zadání názvu aktivity a názvu příslušné XML šablony. Nechal jsem předvyplněné názvy, tedy *MainActivity* pro název aktivity a *activity\_main* pro jméno XML šablony.



Obr. 16. Založení nového Android projektu v prostředí Eclipse

### 4.3 Tvorba základního vzhledu aplikace

Jako první jsem se rozhodl, vytvořit základní grafický kabát, který jsem si navrhl. Abych toho docílil, bylo nutné učinit tři kroky a sice:

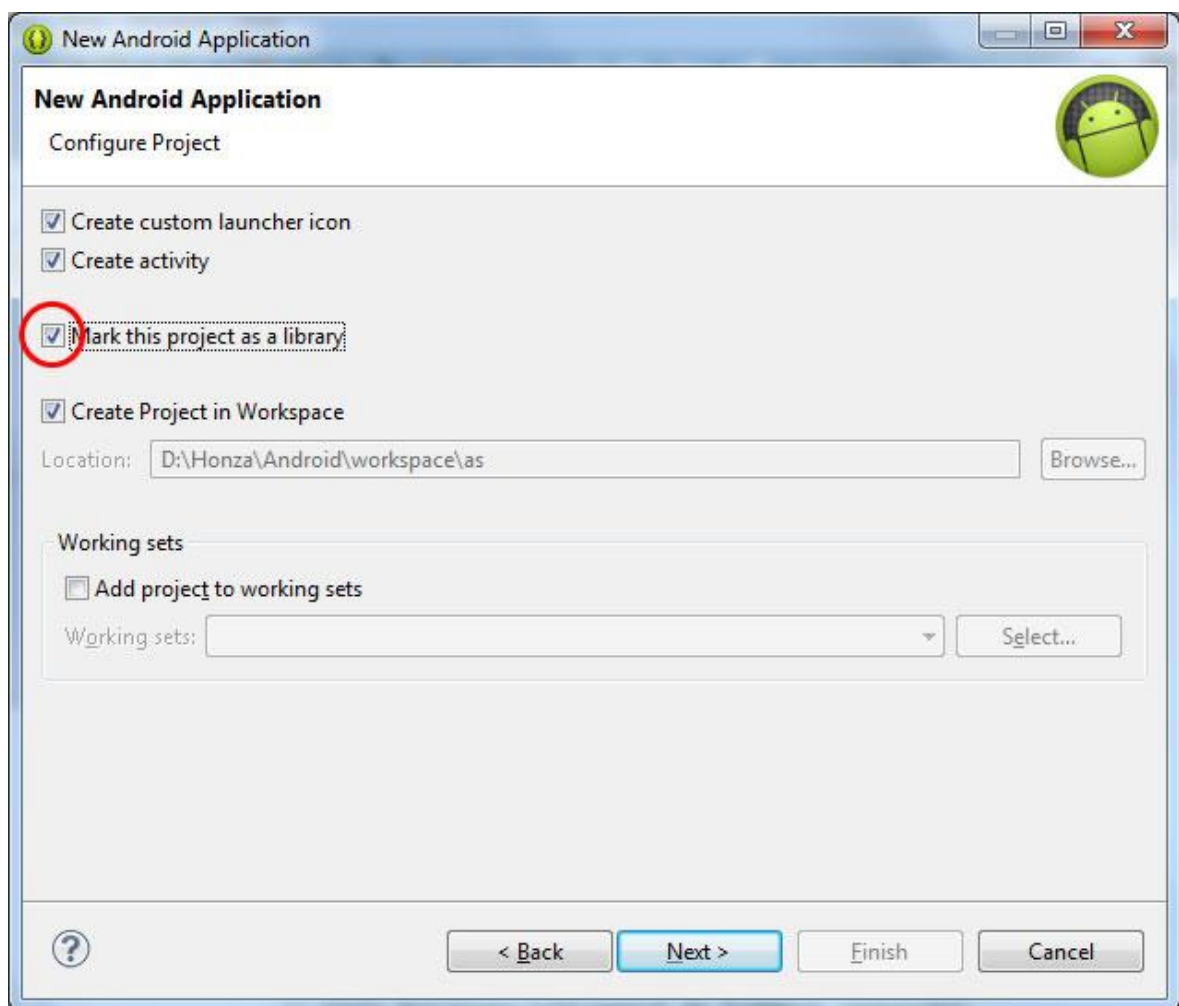
1. Připojit k projektu Support Library
2. Změnit rodičovskou třídu *MainActivity*
3. Definovat nový styl v prostředcích

#### 4.3.1 Support Library

Připojení Support Library k projektu je důležitá hned ze dvou důvodů: jelikož jsem jako minimální podporované API určil verzi 8 (tedy Android 2.2), které v základu neumožňuje použití AB, který jsem navrhl v základním grafickém vzhledu aplikace, a také proto, že aplikace bude využívat fragmentů, které jsou bez Support Library podporované až od Androidu 3.0. V projektu tedy využijeme Support Library verze 7, která umožňuje použití

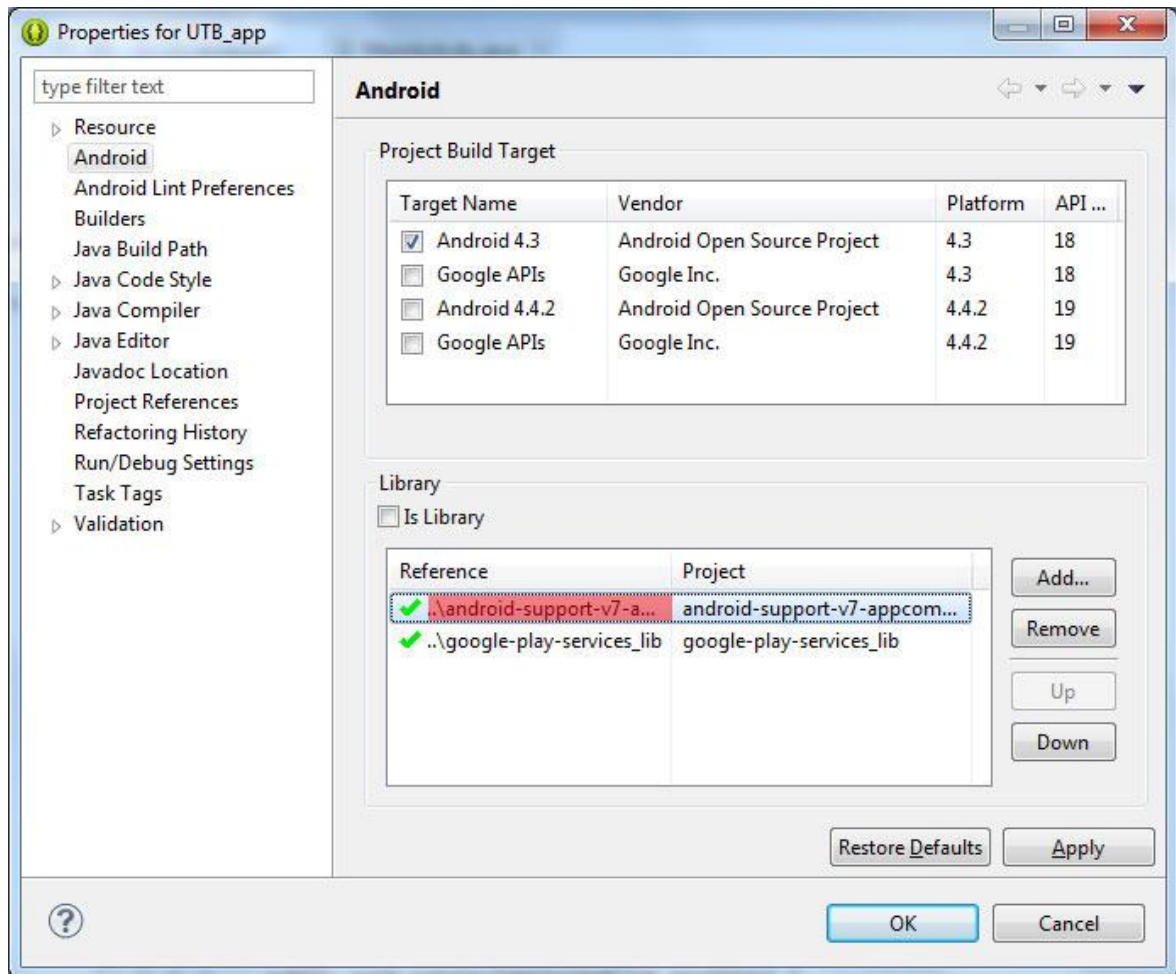
AB až do verze 2.1 a která navíc obsahuje knihovny ze Support Library verze 4, jenž umožňují použití fragmentů až po verzi Androidu 1.6.

V první řadě bylo nutné stáhnout všechny Support Libraries pomocí *SDK Manageru*. Poté jsem založil nový Android Projekt a označil jej jako knihovnu. Do projektu jsem importoval stažené knihovny a dokončil vytváření projektu. Nakonec jsem soubory *android-support-v4.jar* a *android-support-v7-appcompat.jar* v adresáři *libs/* přidal do Build Path (viz Obr. 17).



Obr. 17. Projekt bylo nutné označit jako knihovnu

Nyní již stačilo pouze přidat tyto knihovny do našeho projektu. To jsem provedl tak, že jsem otevřel *vlastnosti* projektu a v záložce *Android* oddíl *Library*, jsem tyto dvě knihovny přidal.



Obr. 18. Přidání Support library do projektu

### 4.3.2 Změny v hlavní aktivitě a definice stylu

Aby se po spuštění aplikace zobrazila aktivita s AB, musel jsem změnit třídu, ze které dědí vygenerovaná třída *MainActivity*. Původní třídu *Activity* jsem tedy nahradil třídou *ActionBarActivity*, která je právě součástí Support Library v7 a která nám umožní používat aktivitu s AB i na starších Androidech.

Ukázka třídy *MainActivity.java*:

```

1 package com.example.utb_app;
2
3 import android.support.v7.app.ActionBarActivity;
4
5 public class MainActivity extends ActionBarActivity {
6     @Override
7     public void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);

```

```
9         setContentView(R.layout.activity_main);
10     }
11 }
```

Následuje definice vlastního stylu. To se provádí v souboru *styles.xml* v adresáři *res/values/*. Nejprve jsem vytvořil styl celé aplikace, který jsem nazval *AppTheme* a do něj definoval styl AB s názvem *ActionBarTheme*. Poté bylo nutné tento styl vytvořit a uvnitř tohoto stylu jsem definoval cestu k XML souboru, který bude reprezentovat pozadí AB a také cestu k ikoně UTB, která bude v levé části AB tak, jak bylo znázorněno v grafickém návrhu aplikace.

Definice stylů v souboru *res/values/styles.xml*:

```
1 <!-- Téma aplikace -->
2 <style name="AppTheme"
   parent="@style/Theme.AppCompat.Light.DarkActionBar">
3     <item name="android:actionBarStyle">@style/ActionBarTheme</item>
4     <item name="actionBarStyle">@style/ActionBarTheme</item>
5 </style>
6
7 <!-- Téma ActionBaru -->
8 <style name="ActionBarTheme"
   parent="@style/Widget.AppCompat.Light.ActionBar.Solid.Inverse">
9     <item name="android:background">@drawable/bg_actionbar</item>
10    <item name="background">@drawable/bg_actionbar</item>
11
12    <item name="android:icon">@drawable/ic_menu</item>
13    <item name="icon">@drawable/ic_menu</item>
14 </style>
```

Kvůli zpětné kompatibilitě bylo každý prvek nutné definovat dvakrát, jak s prefixem *android:*, tak bez prefixu. Dále následuje ukázka souboru *res/drawable/bg\_actionbar*, ve které je definováno pozadí AB:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <layer-list
   xmlns:android="http://schemas.android.com/apk/res/android">
3
4     <item>
5         <shape android:shape="rectangle">
6             <solid android:color="@color/main_orange_dark" />
7         </shape>
8     </item>
```

```
9
10 <item android:bottom="2dp">
11     <shape android:shape="rectangle">
12         <solid android:color="@color/main_orange" />
13     </shape>
14 </item>
15
16 </layer-list>
```

Díky tomuto zápisu bude mít AB čtvercový tvar pozadí a bude oranžové barvy s tmavě oranžovou čarou na spodní hraně. Jednotlivé barvy jsou definovány pomocí hexadecimálního zápisu v souboru *res/values/colors.xml*.

## 4.4 Popis hlavních komponent

Základní vzhled aplikace je hotov a nyní se již budu věnovat třídám psaných v jazyce Java, které představují komponenty aplikace. Programování jednotlivých tříd jsem prováděl samozřejmě průběžně a některé úpravy jsem realizoval zpětně, ale pro přehlednost nebudu uvádět chronologický postup, ale pokusím se popsat jednotlivé třídy zvlášť. Veškeré zdrojové kódy jsou dostupné v CD příloze této práce.

### 4.4.1 MainActivity

Tato třída bude představovat hlavní Activity, která bude načtena při spuštění aplikace. Třidu jsem nemusel vytvářet, protože byla automaticky vygenerována při zakládání projektu. Hlavním úkolem této Activity bude rozhodnout, zdali zobrazit verzi pro mobilní telefony nebo verzi pro tablety. Toho jsem dosáhl pomocí dvou layoutů s názvem *activity\_main.xml*, jeden umístěný v adresáři *res/layout/* a *res/layout-large*. Oba tyto layouts obsahují jiné grafické prvky a právě testování na přítomnost těchto prvků rozhodne o tom, jakým způsobem bude aplikace zobrazovat data. O tom, ze kterého adresáře bude layout přidělen k Activity, rozhoduje Android podle velikosti obrazovky daného zařízení.

Dále bude tato Activity obsluhovat seznam budov a reagovat na výběr konkrétní budovy uživatelem. Výběr adekvátního rozložení obrazovky bude probíhat v metodě *onCreate()*.

Třída navíc implementuje obsluhu výběrového seznamu z komponenty *BuildingsListFragment*. V případě mobilní verze předá identifikační číslo řádku, který byl ze seznamu vybrán, Intentu, a pomocí tohoto Intentu spustí *AdressActivity*. V případě tabletové verze, pouze aktualizuje fragmenty *AddressFragment* a *MapFragment*.

#### 4.4.2 AddressActivity

Tato Activity se používá pouze u mobilní verze a slouží k přepínání mezi Fragmenty *AddressFragment* a *MapFragment* pomocí widgetu *TabHost*.

#### 4.4.3 BuildingsListFragment

Tato třída bude představovat výběrový seznam budov. Na každém řádku seznamu bude název budovy a krátký popis budovy, především tedy název fakulty, ke které budova přísluší. Seznam bude rozdělen na několik částí: jednak pro budovy ve Zlíně a jednak pro budovy detašovaných pracovišť. Proto bylo nutné vytvořit vlastní komponentu, která bude rozhodovat o vzhledu jednotlivých řádků, tato komponenta se jmenuje adapter, v našem případě bude tento adapter představovat třída *BuildingsListAdapter*.

#### 4.4.4 AddressFragment

Tento Fragment se stará o zobrazení obrázku budovy, adresy a kontaktů. Také se stará o obsluhu těchto kontaktů. Tato obsluha je realizována pomocí explicitních Intentů, což znamená, že při výběru nějakého typu kontaktu, nabídne OS Android výběr kompetentních aplikací, které umožní s tímto kontaktem dále pracovat.

#### 4.4.5 MapFragment

*MapFragment* implementuje rozhraní *GoogleMaps* a vykresluje pozici budovy do mapy. Mimo pozici budovy zobrazuje také polohu uživatele (pokud ji zná) a tuto polohu se také snaží udržovat aktuální pomocí GPS nebo Internetu. Do mapy jsou také přidána tlačítka, která umožní vycentrovat pozici budovy nebo uživatele, a také tlačítko pro vyvolání nabídky jiné navigační aplikace, které předá souřadnice budovy.

#### 4.4.6 BuildingsListAdapter

Pomocí této třídy se přiřazuje layout jednotlivým řádkům seznamu. Pokud bychom měli jednoduchý seznam, nebyla by tato třída nutná, ale v našem seznamu se nachází dva typy řádků: nadpis oddílu a řádek s názvem budovy. Řádek budovy navíc obsahuje i krátký popis. Proto je nutné rozlišit každý řádek a přidělit mu příslušný layout.

### 4.5 AndroidManifest.xml

Mimo již vygenerovaných SDK požadavků a zaregistrované *MainActivity*, bylo nutné pro správný chod aplikace dále zaregistrovat komponentu *AdressActivity* a také zapsat

požadavky na udělení oprávnění k některým systémovým funkcím, jako přístup k internetu, nebo k GPS. Dále zde bylo nutné uvést vygenerovaný klíč, který umožňuje používání Google Map.

## 4.6 Testování aplikace

Testování aplikace jsem prováděl na pěti různých zařízeních, z toho byly tři mobilní telefony a dva tablety. Test se sestával ze spuštění aplikace, výběru budovy, otestování funkčnosti kontaktů a také test map a aktualizací pozice uživatele.

*Tab. 2. Přehled testovacích zařízení*

Číslo	Model	Verze Androidu	Velikost Obrazovky
1	Samsung Galaxy SIII	Android 4.3	4,8''
2	LG G Pad V500	Android 4.4.2	8,3''
3	Samsung Galaxy S Duos	Android 4.0.1	4''
4	Lenovo Yoga Tablet	Android 4.2.2	10''
5	Samsung Galaxy mini	Android 2.3.3	3,14''

### 4.6.1 Výsledky testů

Testem prošla všechna zařízení bez problému s výjimkou zařízení č. 5. Na tomto zařízení nebyly nainstalované Služby Google, což při pokusu o načtení mapy, způsobilo pád aplikace, proto bylo nutné toto dodatečně ošetřit. Po nainstalování Služeb Google, již aplikace fungovala správně i na tomto zařízení.

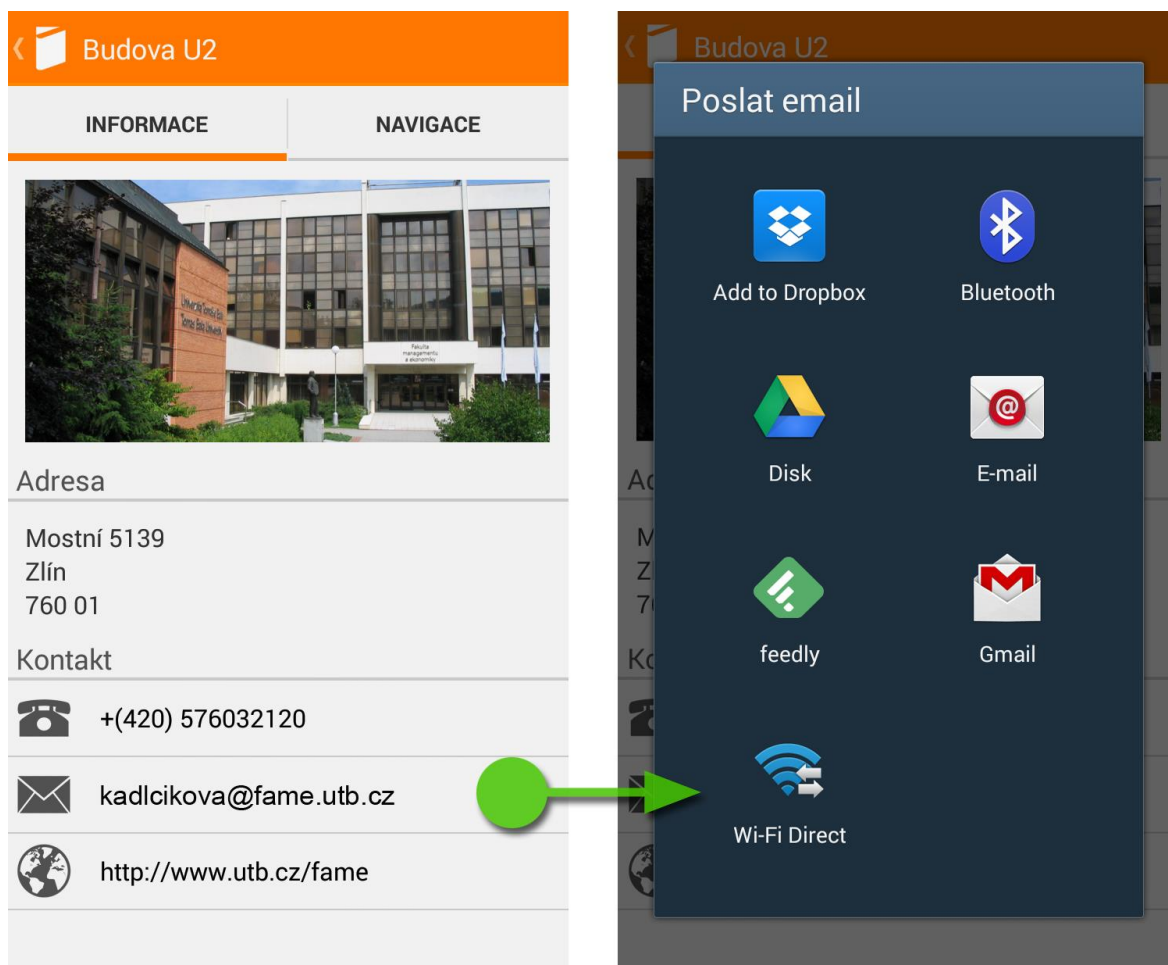
## 4.7 Uživatelská dokumentace

Po spuštění aplikace se na první obrazovce zobrazí výběrový seznam s budovami UTB ve Zlíně, rozřazené do několika kategorií podle měst. Pokud nevidíte budovu, kterou si přejete zobrazit, zkuste kliknout na název města, ve kterém se budova nachází, seznam totiž může být sbalený. Pokud již vidíte příslušnou budovu, klikněte na její název.



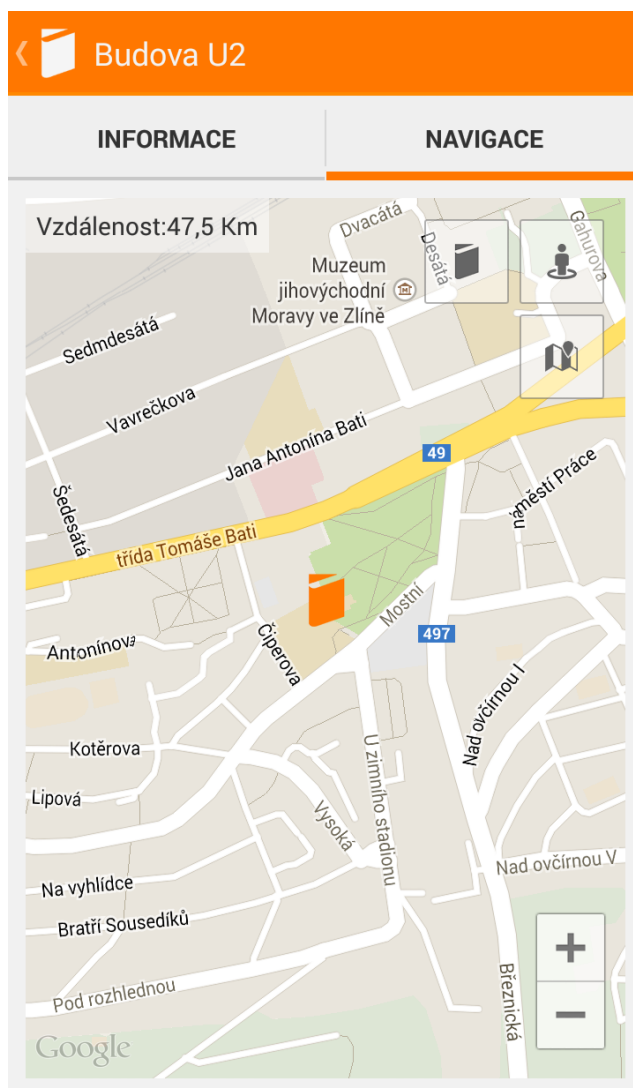
Obr. 19. Obrazovka s výběrovým seznamem

Nyní by se Vám měla zobrazit obrazovka s obrázkem budovy, adresou a kontakty v podobě telefonního čísla, emailové adresy a webové stránky. Pokud na nějaký z těchto kontaktů kliknete, vyvoláte nabídku aplikací, které jsou schopny s tímto kontaktem dále pracovat. Kliknete-li například na e-mailovou adresu, zobrazí se Vám nabídka aplikací, prostřednictvím kterých je možné posílat e-mailové zprávy. Všimněte si také dvojice tlačítek v horní části obrazovky nazvaných *Informace* a *Navigace*. V sekci *Informace* se právě nacházíte. Kliknete-li na tlačítko *Navigace*, přesunete se do sekce s mapou.




Obr. 20. Obrazovka s adresou a kontakty, ukázka vyvolání nabídky aplikací

Nyní se nacházíte na obrazovce s mapou, ve které by měla být vyznačena poloha budovy. Pokud se Vám mapa nezobrazuje, bude nutné se připojit k internetu, aby aplikace stáhla příslušné mapy v okolí budovy. Na mapě se nachází také několik tlačítek. V dolním pravém rohu se nacházejí tlačítka pro přiblížení a oddálení mapy, toho lze dosáhnout také přibližovacím a oddalovacím gestem. V pravém horním rohu se nachází další trojice tlačítek. Tlačítko s logem univerzity slouží k vycentrování mapy k pozici budovy, tlačítko s panáčkem slouží k vycentrování polohy uživatele a tlačítko s mapou vás přesune do aplikace Mapy, které zároveň předá polohu budovy. Pro co nejpřesnější určování Vaší polohy, doporučuji mít zapnuté GPS.




Obr. 21. Obrazovka s mapou a vyznačenou polohou budovy

V případě verze pro tablety, má aplikace naprosto stejné funkce, jen s tím rozdílem, že všechny tři části, tedy seznam s budovami, adresu s kontakty a mapu máte k dispozici na jedné obrazovce.


Budova U2

^ Zlín


- Budova U1**  
Fakulta technologická
- Budova U2**  
Fakulta managementu a ekonomiky
- Budova U3**  
Universitní institut
- Budova U4**  
Fakulta multimediálních komunikací
- Budova U5**  
Fakulta aplikované infromatiky
- Budova U5/56**  
Vědeckotechnický park
- Budova U6**  
Kolej Antonínova
- Budova U7**  
Kolej Štefánikova
- Budova U10**  
Celoškolská pracoviště
- Budova U11**  
Technologický park
- v Uherské hradiště
- v Kroměříž





**Adresa**

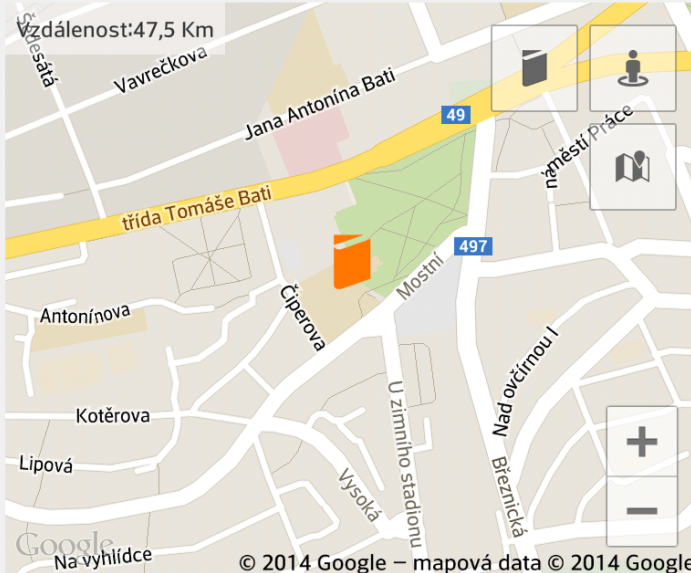
Mostní 5139  
 Zlín  
 760 01

**Kontakt**

 **+(420) 576032120**

 **kadlcikova@fame.utb.cz**

 **<http://www.utb.cz/fame>**



S vzdáleností: 47,5 Km

© 2014 Google – mapová data © 2014 Google

Obr. 22. Ukázka verze pro tablety

## ZÁVĚR

Cílem této bakalářské práce bylo vytvořit aplikaci pro OS Android. Úkolem této aplikace je lokace budov náležících k Univerzitě Tomáše Bati ve Zlíně. V teoretické části této práce jsem se pokusil popsat OS Android, jeho historii a architekturu a také nastínit vývoj aplikací pro tuto platformu. V praktické části jsem se již věnoval samotnému vývoji aplikace. S procesem vývoje aplikace jsem z dřívější doby neměl žádné zkušenosti, a proto jsem se pokusil aplikovat vlastní proces vývoje. Tento proces jsem začal návrhem vzhledu aplikace a ujasněním si základních funkcí. Poté jsem vypracoval obsahovou část aplikace formou tabulky, což byl také jeden z bodů zadání této bakalářské práce. Následoval už samotný vývoj aplikace, který jsem se pokusil i nastínit v textové části, nicméně jsem zjistil, jak je obtížné popisovat zdrojový kód aplikace a proto jsem se v této části omezil pouze na popis jednotlivých tříd. V závěru praktické části jsem vytvořil uživatelskou dokumentaci a popsal proces distribuce aplikace.

Pokud bych měl nějak popsat výhody mého řešení, myslím si, že výsledná aplikace je přehledná a má jednoduchý příjemný vzhled, umožňuje okamžité využití kontaktů spojených s budovou a mezi výhody bych zařadil také optimalizaci pro zařízení s větším displejem.

Vzhledem k existenci velkého množství Android zařízení s různou konfigurací jsem neměl možnost testovat aplikaci na větším počtu těchto zařízení, a tudíž postrádám zpětnou vazbu o její bezchybné funkčnosti. Avšak na mně dostupných testovaných zařízeních aplikace fungovala bez problémů.

Co se týče přínosu, jsem přesvědčen, že by aplikace našla využití u studentů nastupujících do prvních ročníků Univerzity Tomáše Bati ve Zlíně. Aplikace by jim mohla pomoci se zorientovat, pomocí mapy, umístěné v aplikaci se dopravit k příslušné budově a pomocí obrázku budovu identifikovat.

**SEZNAM POUŽITÉ LITERATURY**

- [1] Android, the world's most popular mobile platform. *Android Developers* [online]. 2011 [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/about/index.html>
- [2] VÁVRŮ, J. a M. UJBÁNYAI. *Programujeme pro Android*. 2. rozš. vyd. Praha: Grada, 2013. ISBN 978-80-247-4863-4.
- [3] Industry Leaders Announce Open Platform for Mobile Devices. *Open Handset Alliance* [online]. 2007 [cit. 2014-05-03]. Dostupné z: [http://www.openhandsetalliance.com/press\\_110507.html](http://www.openhandsetalliance.com/press_110507.html)
- [4] Jelly Bean. *Android Developers* [online]. 2011 [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/about/versions/jelly-bean.html>
- [5] KitKat. *Android Developers* [online]. 2011 [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/about/versions/kitkat.html>
- [6] Dashboards. *Android Developers* [online]. 2011 [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/about/dashboards/index.html>
- [7] Android Architecture. *Android App Market* [online]. 2012 [cit. 2014-04-04]. Dostupné z: <http://www.android-app-market.com/android-architecture.html>
- [8] Application Fundamentals. *Android Developers* [online]. 2011 [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/guide/components/fundamentals.html>
- [9] Activities. *Android Developers* [online]. 2011 [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/guide/components/activities.html>
- [10] Services. *Android Developers* [online]. 2011 [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/guide/components/services.html>
- [11] Intents and Intent Filers. *Android Developers* [online]. 2011 [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/guide/components/intents-filters.html>
- [12] Fragments. *Android Developers* [online]. 2011 [cit. 2014-04-04]. Dostupné z:

<http://developer.android.com/guide/components/fragments.html>

- [13] Fragment Reference. *Android Developers* [online]. 2011 [cit. 2014-04-04]. Dostupné z: <http://developer.android.com/reference/android/app/Fragment.html>
- [14] ALLEN, G. *Android 4: průvodce programováním mobilních aplikací..* Překlad Jakub MUŽÍK. Brno: Computer Press, 2013. ISBN 978-80-251-3782-6.
- [15] Buttons. *Android Developers* [online]. 2011 [cit. 2014-05-02]. Dostupné z: <http://developer.android.com/guide/topics/ui/controls/button.html>
- [16] Tabs. *Android Developers* [online]. 2011 [cit. 2014-05-02]. Dostupné z: <http://developer.android.com/design/building-blocks/tabs.html>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

ADT	Android Development Tool
API	Application Programming Interface
DVM	Dalvik Virtual Machine
GPS	Global Positioning System
GUI	Grafické uživatelské rozhraní
IDE	Integrated Development Enviroment
JDK	Java Development Kit
Layout	Grafické rozložení
NFC	Near Field Communication
Open Source	Software s otevřeným zdrojovým kódem
OS	Operační systém
SDK	Software Development Kit
SMS	Short Message Service
VM	Virtual Machine
Widget	Prvek grafického uživatelského rozhraní

**SEZNAM OBRÁZKŮ**

Obr. 1. Graf používaných verzí API na Android zařízeních (k 1. dubnu 2014) [6] .....	15
Obr. 2. Architektura OS Android [7] .....	16
Obr. 3. SDK Manager.....	21
Obr. 4. Vývojové prostředí Eclipse .....	22
Obr. 5. Adresářová struktura android projektu .....	23
Obr. 6. Životní cyklus Activity [9].....	25
Obr. 7. Rozdíl mezi životními cykly started a bound služby a jejich vztah k životnímu cyklu Activity [10].....	27
Obr. 8. Intenty [10].....	29
Obr. 9. Příklad použití fragmentů na různých typech zařízení [12] .....	30
Obr. 10. Životní cyklus Fragmentu. [12].....	31
Obr. 11. Ukázka Widgetů Button a ImageButton [15].....	33
Obr. 12. Ukázka widgetu Tabs [16] .....	33
Obr. 13. Návrh struktury aplikace.....	37
Obr. 14. Základní grafické téma Holo Light DarkActionBar .....	38
Obr. 15. Návrh základního vzhledu aplikace.....	38
Obr. 16. Založení nového Android projektu v prostředí Eclipse .....	40
Obr. 17. Projekt bylo nutné označit jako knihovnu .....	41
Obr. 18. Přidání Support library do projektu .....	42
Obr. 19. Obrazovka s výběrovým seznamem .....	47
Obr. 20. Obrazovka s adresou a kontakty, ukázka vyvolání nabídky aplikací .....	48
Obr. 21. Obrazovka s mapou a vyznačenou polohou budovy .....	49
Obr. 22. Ukázka verze pro tablety .....	50

**SEZNAM TABULEK**

Tab. 1. Přehled používaných verzí na Android zařízeních [6] .....	15
Tab. 2. Přehled testovacích zařízení.....	46

## SEZNAM PŘÍLOH

- P I PŘEHLED BUDOV UTB VE ZLÍNĚ
- P II SEZNAM PŘÍLOH NA CD-ROM

# PŘÍLOHA P I: PŘEHLED BUDOV UTB VE ZLÍNĚ

## Budovy ve Zlíně

Název	Podnázev	Adresa	Kontakt	Souřadnice
U1	Fakulta technologická	nám. T.G. Masaryka 275 Zlín 762 72	tel.: 576 031 111 e-mail: dekanat@ft.utb.cz www: http://www.utb.cz/ft/	49.224438 17.660528
U2	Fakulta managementu a ekonomiky	Mostní 5139 Zlín 760 01	tel.: 576 032 120 e-mail: kadlcikova@fame.utb.cz www: http://www.utb.cz/fame	49.221007 17.657628
U3	Univerzitní institut	Růmy 4046 Zlín 760 01	tel.: - e-mail: - www: http://www.utb.cz/uni	49.219833 17.668228
U4	Fakulta multimediálních komunikací	Štefánikova 2431 Zlín 760 01	tel.: 576 034 205 email: plamborova@fmk.utb.cz www: http://www.utb.cz/fmk	49.222629 17.666185
U5	Fakulta aplikované informatiky	Nad stráněmi 4511 Zlín 760 05	tel.: 576 035 221 e-mail: info@fai.utb.cz http://www.utb.cz/fai	49.230620 17.657510
U5/56	Vědeckotechnický park	Nad stráněmi 5656 Zlín 760 05	tel.: - e-mail: - www: http://www.utb.cz/fai	49.230162 17.656545
U6	Kolej Antonínova	Antonínova 4379 Zlín 760 01	tel.: 576 036 205 email: skubalkova@kmz.utb.cz www: http://www.utb.cz/kmz	49.220576 17.653062
U7	Kolej Štefánikova	Štefánikova 150 Zlín 760 01	tel.: 576 036 118 e-mail: maliskova@kmz.utb.cz www: http://www.utb.cz/kmz	49.224111 17.669399
U10	Celoškolská pracoviště	nám. T.G. Masaryka 1279 Zlín 760 01	tel.: - e-mail: - www: http://www.utb.cz/fhs	49.220203 17.663687
U11	Vědeckotechnický park	Nad Ovčímou 3685 Zlín 76001	tel.: - e-mail: - www: http://www.utb.cz/uni	49.219470 17.662352
U12	Kolej nám. T.G. Masaryka	nám. T.G. Masaryka 3050 Zlín 760 01	tel.: 576 038 002 e-mail: kratinova@kmz.utb.cz www: http://www.utb.cz/kmz	49.220009 17.666486
U13	Univerzitní centrum, knihovna	nám. T.G. Masaryka 5555 Zlín 760 01	tel.: 576 038 130 e-mail: knihovna@utb.cz www: http://www.utb.cz/knihovna	49.222595 17.664896
U14	FHS - Ústav zdravotnických věd	Havlíčkovo nábř. 600 Zlín 762 75	tel.: - e-mail: - www: http://www.utb.cz/fhs	49.230081 17.694004
U16	Fakulta multimediálních komunikací	tř. T. Bati 4342 Zlín 760 01	tel.: 576 034 205 e-mail: plamborova@fmk.utb.cz www: http://www.utb.cz/fmk	49.225511 17.679965
9G	Budova MSI	nám. T.G. Masaryka 1281 Zlín 760 01		49.220503 17.666188
9I	Obchodní akademie T. Bati	nám. T.G. Masaryka 3669 Zlín 760 01		49.222043 17.665070
S1	32. budova areálu Svit	ul. J.A. Bati Zlín 760 01		49.223218 17.656722
ZK	Dům kultury	Cahurova 5265 Zlín 760 01		49.224408 17.660608

## Budovy v Uherském Hradišti

Název	Podnázev	Adresa	Telefon	Souřadnice
UH1	Fakulta logistiky a krizového řízení	Studentské nám. 1532 Uherské hradiště 686 01	tel.: 576 032 080 e-mail: dekanat@flkr.utb.cz www: <a href="http://www.utb.cz/flkr">http://www.utb.cz/flkr</a>	49.070324 17.472873
UH2	Fakulta logistiky a krizového řízení	Studentské nám. 1532 Uherské hradiště 686 01	tel.: 576 032 080 e-mail: dekanat@flkr.utb.cz www: <a href="http://www.utb.cz/flkr">http://www.utb.cz/flkr</a>	49.070324 17.472873
UH3	Centrum celoživotního vzdělávání	Studentské nám. 1531 Uherské hradiště 686 01		49.070866 17.471927
UH4	Studentská kolej	Studentské nám. 1533 Uherské hradiště 686 01	tel.: 572 503 738 e-mail: <a href="mailto:info@eduha.cz">info@eduha.cz</a> www: <a href="http://www.eduha.cz">http://www.eduha.cz</a>	49.069579 17.472794
UH5	Studentská kolej, menza	Studentské nám. 1535 Uherské hradiště 686 01	tel.: 571 891 286 e-mail: <a href="mailto:info@eduha.cz">info@eduha.cz</a> www: <a href="http://www.eduha.cz">http://www.eduha.cz</a>	49.069676 17.471157

## Budovy v Kroměříži

Název	Podnázev	Adresa	Telefon	Souřadnice
K1	Fakulta technologická	Štěchovice 1358 Kroměříž 767 54	tel.: 573 334 937 e-mail: <a href="mailto:vospaspsm@vospaspsm.cz">vospaspsm@vospaspsm.cz</a> www: <a href="http://vospaspsm.cz">http://vospaspsm.cz</a>	49.299960 17.383810

## **PŘÍLOHA P II: SEZNAM PŘÍLOH NA CD-ROM**

- **aplikace** – adresář obsahující projektovou složku aplikace
- **obrazky** – adresář obsahující originály obrázků budov, použitých v aplikaci
- **prehled.xlsx** – soubor aplikace Microsoft Excel, který obsahuje vypracovaný přehled budov UTB ve Zlíně