

# Mobilní aplikace pro klienty obchodních center

Bc. Radovan Holík

---

Diplomová práce  
2014



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

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

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Radovan Holík**  
Osobní číslo: **A12461**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Forma studia: **kombinovaná**

Téma práce: **Mobilní aplikace pro klienty obchodních center**

Zásady pro vypracování:

1. Prostudujte a popište možnosti a prostředky pro vývoj mobilních aplikací.
2. Popište způsoby optimalizace aplikace a návrh uživatelského rozhraní pro platformu Android.
3. Prostudujte a popište základy bezpečnosti a zranitelnosti mobilní platformy Android.
4. Navrhněte aplikaci pro obchodní centrum včetně rezervace vstupenek do kina.
5. Podle návrhu vytvořte prototyp aplikace.
6. Navrhněte a implementujte programové vybavení potřebné pro funkčnost aplikace.
7. Demonstrujte výsledky a proveďte vyhodnocení.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. MISRA, Anmol a Abhishek DUBEY. *Android Security: Attacks and Defenses*. Boca Raton, Fla: CRC Press, 2013, 255 pages. ISBN 14-398-9646-1.
2. SZYDLOWSKÁ, Markéta. *Bezpečnost OS Android*. Brno, 2012. Dostupné z: [http://is.muni.cz/th/325506/fi\\_b/?lang=en](http://is.muni.cz/th/325506/fi_b/?lang=en). Bakalářská práce. Masarykova Univerzita. Vedoucí práce Ing. Mgr. Zdeněk Říha, Ph.D.
3. ALLEN, Grant. *Android 4: průvodce programováním mobilních aplikací*. 1. vyd. Překlad Jakub Mužík. Brno: Computer Press, 2013. ISBN 978-80-251-3782-6.
4. MEDNIEKS, Zigurd, G. Blake MEIKE, Laird DORNIN a Zane PAN. *Enterprise Android: Programming Android Database Applications for the Enterprise*. 1st ed. US: Wrox, 2013, 408 s. ISBN 11-181-8349-5.
5. Google Inc., *Android Developers [online]*. 2014. Dostupné z: <http://developer.android.com/>

Vedoucí diplomové práce:

**Ing. Radek Šilhavý, Ph.D.**

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

**21. února 2014**

Termín odevzdání diplomové práce:

**20. května 2014**

Ve Zlíně dne 21. února 2014

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



doc. Mgr. Roman Jašek, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Tato práce se zabývá analýzou a zhodnocením možností pro vývoj mobilní aplikace určené především klientům obchodních center. Teoretická část se věnuje požadavkům a charakteristikám mobilních aplikací, analýze dostupných vývojových technik a především platformě Android. Část práce je věnována také všeobecné bezpečnosti a bezpečnostním mechanismům této platformy. Jsou popsány základní principy online rezervace a plateb pomocí mobilní aplikace. V praktické části je provedena analýza a hodnocení již existujících řešení spolu s vlastním návrhem prototypu aplikace. Hlavním cílem práce je navržení programového vybavení a jeho následná implementace do aplikace obchodního centra včetně rezervace vstupenek do kina.

Klíčová slova:

Android, mobilní platformy, bezpečnost platformy Android, notifikace, vývoj aplikací, mobilní platby

## **ABSTRACT**

The thesis is focused on analysis and evaluation of possible ways of development of mobile application for shopping centres. In the theory are described requirements and characteristics of mobile applications. There are also described available development techniques and Android platform, native security and another security mechanisms of Android. Next part of this work deals with possible ways of online reservation and payments by mobile application. In the practical part is done analysis of existing solutions and also a prototype of my own solution. The main goal of the thesis is to design software components and their next implementation into the mobile application of a shopping centre with reservation system of a cinema.

Keywords:

Android, mobile platforms, Android security, notifications, application development, mobile payments

Tímto bych rád poděkoval panu Ing. Radkovi Šilhavému, Ph.D. za cenné rady, připomínky a veškerý čas, který mi věnoval při odborném vedení této práce.

**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 práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně dne 19. 5. 2014

.....  
podpis diplomanta

**OBSAH**

|  |           |
|--|-----------|
| <b>ÚVOD</b> .....  | <b>10</b> |
| <b>I TEORETICKÁ ČÁST</b> .....   | <b>12</b> |
| <b>1 MOBILNÍ APLIKACE A JEJICH VÝVOJ</b> .....                                     | <b>13</b> |
| 1.1 WEBOVÉ APLIKACE .....  | 14        |
| 1.2 NATIVNÍ APLIKACE.....  | 15        |
| 1.3 HYBRIDNÍ APLIKACE .....  | 17        |
| 1.4 ZÁVĚR.....   | 18        |
| <b>2 OPERAČNÍ SYSTÉM ANDROID</b> .....   | <b>20</b> |
| 2.1 HISTORIE OS ANDROID .....  | 20        |
| 2.2 ARCHITEKTURA OS ANDROID.....   | 22        |
| 2.2.1 Základní členění OS Android.....   | 23        |
| 2.2.2 Struktura Android aplikace .....   | 23        |
| 2.3 ROZŠÍŘENÍ ZÁKLADU OS ANDROID.....  | 25        |
| 2.3.1 Předinstalované aplikace .....   | 25        |
| 2.3.2 Aplikace instalované uživatelem.....   | 25        |
| 2.3.3 Google Play .....  | 26        |
| 2.3.4 Aktualizace.....   | 26        |
| <b>3 BEZPEČNOST OPERAČNÍHO SYSTÉMU ANDROID</b> .....                               | <b>28</b> |
| 3.1 MÝTY O BEZPEČNOSTI ANDROIDU.....   | 28        |
| 3.2 KAŽDÁ APLIKACE VE SVÉM VLASTNÍM ADRESNÍM PROSTORU .....                        | 29        |
| 3.2.1 Povolení přístupu k systémovým zdrojům .....                                 | 29        |
| 3.3 APLIKACE A VIRY .....  | 31        |
| 3.4 SPOUŠTĚNÍ ŠKODLIVÉHO KÓDU .....  | 31        |
| 3.5 POUŽITÍ ANTIVIRU .....   | 31        |
| 3.6 COVER TECHNIQUE - NEJNOVĚJŠÍ PRAKTIKA ŠKODLIVÉHO SOFTWARE .....                | 32        |
| 3.7 MÁ VÝZNAM TZV. „ROOT“ ZAŘÍZENÍ? .....  | 33        |
| <b>4 ZPŮSOBY OPTIMALIZACE PRO RŮZNÁ ZAŘÍZENÍ A TYPY NA PLATFORMĚ ANDROID</b> ..... | <b>34</b> |
| 4.1 ANDROID DESIGN GUIDELINES .....  | 34        |
| 4.2 OPTIMALIZACE RŮZNÝCH VELIKOSTÍ A HUSTOTY OBRAZOVEK.....                        | 38        |
| 4.2.1 Prvky ovlivňující optimalizaci .....   | 39        |
| 4.3 TABLETY A VĚTŠÍ ZAŘÍZENÍ.....  | 41        |
| 4.3.1 Fragmenty .....  | 43        |
| <b>5 ZPŮSOBY ONLINE REZERVACE A MOŽNOSTI PLATEB PŘES MOBILNÍ APLIKACI</b> .....    | <b>46</b> |
| 5.1 ONLINE REZERVACE .....   | 46        |
| 5.2 MOBILNÍ PLATBY .....   | 46        |
| 5.2.1 Premium SMS .....  | 47        |
| 5.2.2 Přímé zpoplatnění u operátora .....  | 48        |
| 5.2.3 Mobilní webové platby .....  | 49        |
| 5.2.3.1 Platební karty .....   | 49        |

|           |   |           |
|-----------|---|-----------|
| 5.2.3.2   | Elektronické peněženky .....                  | 50        |
| <b>II</b> | <b>PRAKTICKÁ ČÁST .....</b>                   | <b>52</b> |
| <b>6</b>  | <b>ANALÝZA .....</b>                          | <b>53</b> |
| 6.1       | ANALÝZA JIŽ EXISTUJÍCÍCH ŘEŠENÍ .....         | 53        |
| 6.2       | HODNOCENÍ .....                               | 58        |
| <b>7</b>  | <b>POŽADAVKY APLIKACE .....</b>               | <b>60</b> |
| 7.1       | SPECIFIKA POŽADAVKŮ .....                     | 60        |
| 7.1.1     | Jazyk aplikace .....                          | 61        |
| 7.1.2     | Potenciální uživatelé .....                   | 61        |
| 7.1.3     | Softwarové požadavky .....                    | 61        |
| 7.1.4     | Hardwarové požadavky .....                    | 62        |
| 7.1.5     | Zdroj dat .....                               | 62        |
| 7.2       | POŽADOVANÁ FUNKCIONALITA .....                | 62        |
| 7.2.1     | Základní popis .....                          | 62        |
| 7.2.2     | Účel .....                                    | 63        |
| 7.2.3     | Rozsah .....                                  | 63        |
| 7.2.4     | Uživatelé .....                               | 63        |
| 7.2.5     | Omezující podmínky .....                      | 64        |
| <b>8</b>  | <b>NÁVRH PROTOTYPU APLIKACE .....</b>         | <b>65</b> |
| 8.1       | USE-CASE DIAGRAM .....                        | 65        |
| 8.2       | UŽIVATELSKÉ ROZHRANÍ .....                    | 67        |
| 8.2.1     | Hlavní aktivita, menu a fragment ‚News‘ ..... | 67        |
| 8.2.2     | Fragment ‚Shops‘ .....                        | 68        |
| 8.2.3     | Fragment ‚Services‘ .....                     | 69        |
| 8.2.4     | Fragment ‚Map‘ a ‚About‘ .....                | 70        |
| 8.2.5     | Aktivita ‚Cinema‘ .....                       | 71        |
| 8.2.5.1   | Fragment ‚Week Schedule‘ .....                | 71        |
| 8.2.5.2   | Aktivita ‚Film Details‘ .....                 | 72        |
| 8.2.5.3   | Aktivita ‚Reservation‘ .....                  | 73        |
| 8.2.5.4   | Fragment ‚My Reservations‘ .....              | 74        |
| 8.2.6     | Aktivita ‚Settings‘ .....                     | 75        |
| 8.2.7     | Notifikace .....                              | 76        |
| <b>9</b>  | <b>NÁVRH ARCHITEKTURY .....</b>               | <b>77</b> |
| 9.1       | CLASS DIAGRAMY .....                          | 77        |
| <b>10</b> | <b>IMPLEMENTACE .....</b>                     | <b>81</b> |
| 10.1      | ZÍSKÁVÁNÍ DAT .....                           | 81        |
| 10.2      | UKLÁDÁNÍ DAT .....                            | 83        |
| 10.2.1    | Databáze v Androidu .....                     | 83        |
| 10.2.2    | Shared Preferences .....                      | 85        |
| 10.3      | AUTOMATICKÉ STAHOVÁNÍ DAT .....               | 86        |
| 10.3.1    | Alarm manager a pending intent .....          | 86        |
| <b>11</b> | <b>VÝSLEDNÁ ANDROID APLIKACE .....</b>        | <b>88</b> |
| 11.1      | UKÁZKA APLIKACE .....                         | 88        |
| 11.2      | MOŽNOSTI ROZŠÍŘENÍ .....                      | 95        |
|           | <b>ZÁVĚR .....</b>                            | <b>96</b> |

---

|  |            |
|--|------------|
| <b>SEZNAM POUŽITÉ LITERATURY.....</b>          | <b>97</b>  |
| <b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b> | <b>101</b> |
| <b>SEZNAM OBRÁZKŮ .....</b>                    | <b>102</b> |
| <b>SEZNAM TABULEK.....</b>                     | <b>105</b> |
| <b>SEZNAM KÓDŮ .....</b>                       | <b>106</b> |
| <b>SEZNAM PŘÍLOH.....</b>                      | <b>107</b> |

## ÚVOD

Smartphony neboli chytré telefony jsou dnes již běžnou záležitostí. Zcela nahrazují starší typy mobilních telefonů a k tomu přinášejí spoustu dalších funkcí, jako je bezdrátové připojení, GPS, gyroskop a další. Hlavně díky internetovému připojení se enormně rozšířilo pole působnosti právě chytrých telefonů, které dnes už nejsou pouze nástrojem pro uskutečnění telefonních hovorů a krátkých SMS zpráv. Také nesmíme zapomínat, že trh si začínají podmaňovat i tablety, které se těší stále větší oblibě mezi běžnými uživateli a svým výkonem a mobilitou dokážou mnohdy předčít leckterý notebook. Tato zařízení dosáhla obrovského technologického pokroku a v dnešní době i téměř úroveň běžných počítačů. Můžeme instalovat aplikace, dohledávat informace na internetu, přistupovat k sociálním sítím, hrát hry, sledovat filmy, pořizovat snímky a videa, přistupovat do internetového bankovníctví a spoustu dalšího. To vše pomocí jediného zařízení.

Mobilní telefon je nejintimnější médium, kterým se dá na člověka působit, protože mobilní telefon máme většinou všichni stále u sebe na rozdíl od počítače. Proto při plánování marketingových strategií mají mobilní zařízení stále vyšší prioritu a s tím souvisí i následná optimalizace dat, které chceme uživateli poskytnout. Můžeme využít responzivního webu anebo mobilní aplikace, která je schopna se lépe přizpůsobit dané platformě a využít všechny funkce zařízení. Problematika rozličnosti vývoje mobilních aplikací je popsána v první kapitole této diplomové práce.

V teoretické části práce je představen operační systém Android, kde je i stručně popsána jeho bezpečnost a rizika plynoucí z jeho užívání. V této části jsou rozebrány také způsoby optimalizace tohoto operačního systému pro různé velikosti obrazovek a různé rozlišení displejů. V poslední kapitole teoretické části jsou popsány možnosti rezervace a mobilních plateb pomocí mobilního zařízení.

V praktické části práce bylo nejprve třeba zanalyzovat již několik existujících řešení. Bylo vytvořeno odpovídající srovnání a hodnocení, které pomohlo spolu se stanovenými požadavky vytvořit návrh vylepšeného řešení a prototypu mobilní aplikace obchodního centra včetně implementace rezervačního systému vstupenek do kina. Návrh systému zahrnuje i vlastní nápady a poznatky. Další kapitola v této části je věnována implementaci a představení jednotlivých funkcí a části kódu, které jsou v aplikaci využívány.

Výsledná aplikace je zaměřena zejména na chytré telefony, ale správně funguje i na tabletech a v režimu „na šířku“. Pro svůj plnohodnotný provoz potřebuje alespoň občasné

připojení k internetu, pomocí kterého si stahuje nová data ze serveru obchodního centra. Uživatel má tak dostupné pohromadě všechny informace z obchodního centra včetně těch aktuálních. Aplikace slouží i k zobrazení aktuálního programu kina s možností rezervace vstupenek.

## **I. TEORETICKÁ ČÁST**

## 1 MOBILNÍ APLIKACE A JEJICH VÝVOJ

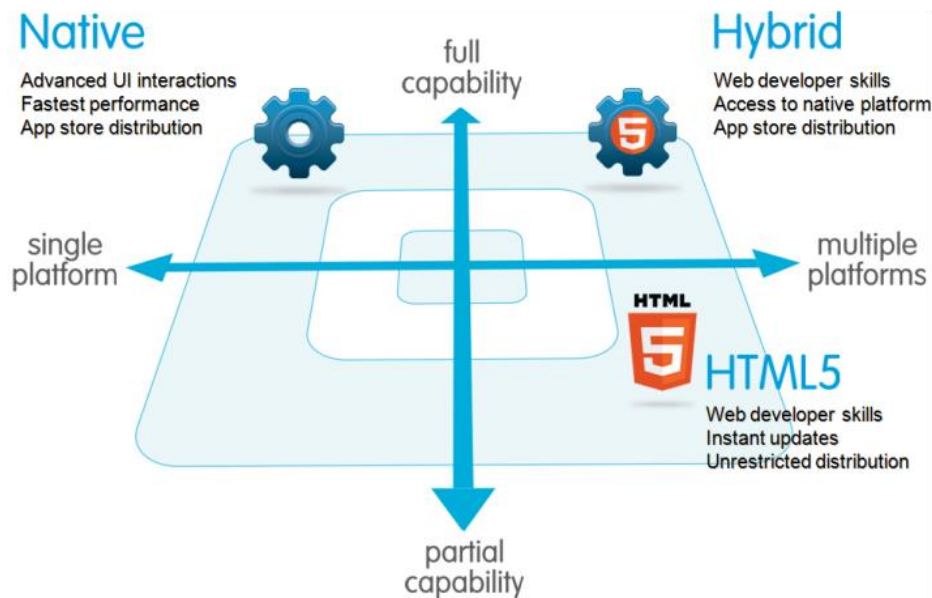
Mobilní aplikace, často označována též jako „*mobile app*“, je typem aplikačního softwaru, který je navržen k běhu na mobilních zařízeních (tj. chytrý telefon, tablet, televize atd.). Mobilní aplikace často slouží k poskytování stejných služeb, ke kterým mohou uživatelé přistoupit z klasického počítače (např. e-mail klient, kalendář atd.). Aplikace jsou obvykle jednoduché a plní funkci softwaru s omezenými funkcemi. Momentálně se čím dál víc můžeme setkat s názorem, že mobilní aplikace nejsou pouze určitým marketingovým nástrojem, ale že jsou pro lidi opravdu užitečné a žádané.

Je zde mnoho faktorů, které ovlivňují mobilní strategii při vývoji aplikace, jako jsou zkušenosti vývojového týmu, nezbytné funkce daného zařízení, důležitost zabezpečení, schopnost fungovat bez nutnosti připojení k Internetu, schopnost systému vzájemně si poskytovat služby a efektivně spolupracovat a mnoho dalších, které je třeba brát v úvahu. Z hlediska vývoje není nejdůležitější vědět, co přesně aplikace bude dělat, ale jakým způsobem jsme ji schopni naprogramovat, aby se chovala, jak očekáváme nebo potřebujeme. Mobilní aplikace se dají vyvíjet několika způsoby a každý z těchto způsobů má jisté výhody i nevýhody oproti ostatním. [15][17][18]

Mobilní aplikace mohou být:

- **webové** – využívají standardní webové technologie – typicky HTML5, Javascript a CSS. Vyznačují se přístupem „*napiš jednou a spust' všude*“, kdy je vytvořena cross-platformová mobilní aplikace, která je schopna běžet na různých zařízeních s odlišným operačním systémem. Vývojáři jsou sice schopni vytvořit sofistikovanou aplikaci za použití HTML5 a Javascriptu, ale stále přetrvávají některá omezení, která zabraňují využití plného potenciálu chytrého zařízení (tj. zabezpečení offline paměti a přístup k nativním funkcionalitám) [20]
- **nativní** – jsou specifické pro každou mobilní platformu (iOS, Android, WP...) a pro jejich vývoj se používá vývojových nástrojů a programovacího jazyka, které příslušná platforma podporuje (tj. Xcode a Objective-C pro iOS, Java a Eclipse nebo Android Studio pro Android, C# a VisualStudio pro WP). Nativní aplikace běží nejrychleji a také vypadají nejlépe (myšleno tak, že komponenty jsou navrženy přímo pro danou platformu i verzi a tím zapadají do celkového konceptu, jak to vývojový tým operačního systému zamýšlel).

- **hybridní** – jak už název naznačuje, jedná se o kombinaci předchozích dvou způsobů. Tedy, jakési zabudované HTML5 aplikace, která je obalena nativním kontejnerem (prohlížečem). Kombinuje nejlepší (a také nejhorší) prvky nativních a webových aplikací.



Obr. 1. Přehled možných mobilních aplikací a jejich hlavních kladů [20]

## 1.1 Webové aplikace

Webové aplikace mají jednu obrovskou výhodu, a sice tu, že vývoj v HTML5 je mnohem jednodušší než v objektově orientovaných programovacích jazycích zejména proto, že má jednodušší syntaxi. HTML5 aplikace je prakticky jedna nebo více webových stránek, které jsou navrženy k práci na úzkých obrazovkách. Tyto aplikace jsou uloženy na serveru a může k nim být přistoupeno pomocí moderních prohlížečů. Protože obsah je na webu, může poskytnout větší komfort např. při vyhledávání v e-shopu.

Nespornou výhodou je, že pro webové vývojáře, kteří mají zkušenosti s většími webovými aplikacemi, je vývoj webových mobilních aplikací velice snadnou záležitostí, ale i pro nové začínající vývojáře je daleko jednodušší vyvíjet mobilní webové aplikace, než nativní nebo hybridní aplikace. Dá se říct, že každé mobilní zařízení má vlastní představu o tom, co představuje velikost využitelné obrazovky a rozlišení a s tím související testování na různých zařízeních a emulátorech. S tím samozřejmě může souviset i možná nekompatibilita funkcí na daných prohlížečích (zejména na Androidu).

Důležitou součástí metodiky HTML5 „*napiš jednou, spust' všude*“ je, že distribuce a podpora je mnohem jednodušší, než je tomu u nativních aplikací. Pouze se nahraje nová verze na server a ta je ihned dostupná všem uživatelům. V předchozích dvou letech se HTML5 ukázal jako velmi populární způsob pro vývoj mobilních aplikací a také vývoj komponent šel rychle dopředu. Nyní existují UI frameworky, které poskytují různé komponenty pro vývoj webových aplikací, např. JQuery Mobile a Sencha Touch poskytují elegantní mobilní komponenty se stovkami různých pluginů, které nabízejí prakticky vše, co může programátor potřebovat (tj. od základních až po nejpropracovanější ovládací prvky, které svým designem napodobují nativní komponenty). [18]

Přestože je HTML5 aplikace jednodušší vyvíjet, jsou lépe udržovatelné a nezávislé na operačním systému, mají i své nevýhody. Ve většině případů není možné přistupovat k nativním komponentám a funkcím na zařízení. S tím souvisí, že uživatelé nemají z designu a funkčnosti tak přirozený pocit, jako je tomu u aplikací nativních a ani nemůžou používat různá gesta, na která mohou být zvyklí. Ovšem prohlížeče jsou postupně vylepšovány a je k dispozici stále více a více funkcí, která mohou výše uvedené údaje brzo zvrátit. Poslední verze prohlížečů podporují hardwarově akcelerované vlastnosti CSS3 a animace, které poskytují plynulý pohyb pro posuvné panely i přechody mezi obrazovkami, ale ani tak se nemůžou rovnat ve výkonu a flexibilitě s aplikacemi nativními. Prozatím není možné např. multitouch, neboli schopnost zaznamenávat doteky několika prsty na obrazovce současně. Nejvýznamnějším omezením pro HTML5 webové aplikace je nemožnost nasazení do podnikové sféry, protože nedokážou zajistit bezpečnost při ukládání do interního úložiště. Data v interním úložišti sice mohou být šifrována, ale zdaleka neposkytují stejnou bezpečnost jako asymetrická šifra, které chrání jednotlivé nativní aplikace s certifikátem vývojáře. Další nevýhodou je, že uživatel by se k firemnímu serveru vždy musel prokázat platným uživatelským jménem a heslem při každém přihlášení a to je pro něj obvykle nepříjemný zážitek. Obecně platí, že provádění i triviálních bezpečnostních opatření na nativní platformě může být velice složité až neuskutečnitelné pro webové mobilní vývojáře. [17]

## 1.2 Nativní aplikace

Stručně řečeno, nativní aplikace poskytují nejlepší použitelnost, ty nejlepší vlastnosti a celkově nejlepší mobilní zážitky, kterých lze s mobilní aplikací dosáhnout. Je zde pár věcí, kterých lze dosáhnout pouze pomocí nativní aplikace:

- **multitouch** – dvojité tahy a jiná složená UI gesta.
- **rychlá grafika** – nativní platforma poskytuje nejrychlejší grafiku, kterou lze docílit na mobilních zařízeních.
- **schopnost „tekutých“ animací** – tyto animace se vztahují k předchozímu bodu, kdy využívají nejrychlejší grafické API se schopností „tekutých“ animací. Tato vlastnost je obzvláště důležitá ve hrách, v interaktivních oznámeních nebo intenzivních výpočetních algoritmech pro transformaci fotografií a zvuků.
- **vestavěné komponenty** – fotoaparát, adresář, geolokace a další funkce, které mohou být integrovány do nativní mobilní aplikace. Další výhodou je šifrování datového úložiště.
- **snadné použití** – nativní platformy, to je to, na co jsou lidé zvyklí. V případě integrace nativních komponent do mobilní aplikace poskytují to, co uživatelé očekávají.
- **dokumentace** – k dispozici je nespočet knih věnující se vývoji nativních aplikací, mnoho článků, blogů a podobných technických vláken na místech, jako je StackOverflow<sup>1</sup>.

Nativní aplikace jsou obvykle vyvíjeny pomocí integrovaného vývojového prostředí<sup>2</sup> (dále jen IDE). IDE poskytují nástroje pro vytváření, ladění, projekt management, verzovací systém a další nástroje pro profesionální vývoj. Úroveň zkušeností vývojáře je požadována vyšší, než je tomu u webového vývoje. Ve většině případů neplatí, že lze snadno zkopírovat např. Java kód nalezený někde na Internetu, vložit do vývojového prostředí a očekávat, že bude fungovat. Je skutečnosti samotné „*know how*“ vývojáře, či vývojového týmu je důležitým faktorem při vývoji nativních aplikací.[17]

Nativní aplikace jsou nejpřívětivější pro koncového uživatele a navíc uživatel ve většině případů hledá mobilní aplikaci právě v obchodě (Appstore, Google Play, Windows Store), kterou si poté nainstaluje a může ji rovnou spustit jedním klikem. Jak už bylo napsáno výše, nativní aplikace poskytuje vysoký výkon, příjemný pocit ze vzhledu celistvé konzistence platformy. V případě, že aplikace potřebuje aktualizaci, sama upozorní

---

<sup>1</sup> Více na: <http://stackoverflow.com/>

<sup>2</sup> IDE – Integrated Development Environment

uživatele, aby aktualizoval. Jednoduše, mobilní aplikace může poskytnout uživateli vše, čeho lze na dané platformě dosáhnout. Bezespору největší a asi i jedinou nevýhodou je časová a tím pádem i finanční náročnost při vývoji nativních aplikací. Chceme-li vyvinout aplikaci nativně na všechny platformy, musejí se na každou platformu vyvíjet zvlášť.

### 1.3 Hybridní aplikace

Hybridní vývoj kombinuje nejlepší (nebo i nejhorší) z obou výše uvedených způsobů vývoje mobilních aplikací. Hybridní aplikaci můžeme definovat jako webovou aplikaci, primárně postavenou na HTML5 a Javascriptu, která je pak obalena nativním kontejnerem (obvykle prohlížečem), který poskytuje přístup k nativním vlastnostem platformy. Apache Cordova<sup>3</sup> (dříve PhoneGap) je nejznámějším typem frameworku, který se používá k vývoji mobilních hybridních aplikací.

Z celkového pohledu hybridní aplikace poskytují to nejlepší z obou světů. Existující weboví vývojáři, kteří mají zkušenosti s Javascriptem a CSS mohou vytvořit dobře vypadající layout a s přidáním HTML5 kódu, který funguje na každé platformě, vyvinout sofistikovanou mobilní aplikaci, která dokáže pracovat s nativními komponentami. V předchozích podkapitolách bylo řečeno, že nativní aplikace jsou nainstalovány v přístroji, zatímco HTML5 webové aplikace jsou umístěny na webovém serveru. Jak je to tedy s hybridními aplikacemi? Ve skutečnosti existují dva způsoby, jak implementovat hybridní aplikace:

- **lokálně** – HTML a Javascript kód lze zabalit do binárního kódu podobným způsobem, jako na struktury nativní aplikace. V tomto způsobu může být použito REST API k přesunu dat tam a zpět mezi zařízením a cloudovým úložištěm.
- **serverově** – může být implementována plně webová aplikace ze serveru (s volitelnou cache pro lepší výkon), pouze s použitím kontejneru, který obalí UIWebView (nativní kit prohlížeče). V tomto případě uživatel může kliknout na ikonku a spustí se mu aplikace stejným způsobem jako nativní, ale chová se jako webová.

---

<sup>3</sup> Více na: <http://cordova.apache.org/>

## 1.4 Závěr

Nativní a hybridní aplikace jsou uloženy v app storech (el. obchodech), zatímco webové aplikace jsou mobilně-optimalizované webové stránky vypadající jako aplikace. Hybridní aplikace i webové aplikace využívají HTML5, ale hybridní používá prohlížeče z nativní platformy. Mobilní vývoj je neustále v pohybu a prakticky každých 6 měsíců je tu nový operační systém s unikátními vlastnostmi, které jsou přístupné pouze s nativními API. Frameworky, které dokážou vytvořit hybridní aplikaci schopnou používat nativní komponenty, dostávají tyto novinky později, protože je třeba nejdříve tyto novinky implementovat do vlastních funkcí. Přehled jednotlivých způsobů vývoje mobilních aplikací a jejich funkcí můžeme vidět v Tab. 1, 2, 3. Jako nejlepší volba se jeví aplikace hybridní a nativní. Pro hybridní je největší výhoda v tom, že poskytuje tzv. Cross-platformní vývoj a v některých případech i mnohem nižší náklady na tvorbu mobilních aplikací, které jsou určeny pro více platforem.

Tab. 1. Vlastnosti jednotlivých mobilních aplikací

| Vlastnosti aplikací | Nativní     | Webové            | Hybridní          |
|---------------------|-------------|-------------------|-------------------|
| Grafika             | Nativní API | HTML, Canvas, SVG | HTML, Canvas, SVG |
| Výkon               | Rychlé      | Pomalé            | Pomalé            |
| Vzhled              | Nativní     | Emulovaný         | Emulovaný         |
| Distribuce          | Obchody     | Web               | Obchody           |

Tab. 2. Přístup k nativním prvkům jednotlivých mobilních aplikací

| Přístup k nativním prvkům | Nativní           | Webové      | Hybridní                          |
|---------------------------|-------------------|-------------|-----------------------------------|
| Kamera                    | Ano               | Ne          | Ano                               |
| Notifikace                | Ano               | Ne          | Ano                               |
| Kontakty, kalendář        | Ano               | Ne          | Ano                               |
| Offline úložiště          | Bezpečné úložiště | Sdílená SQL | Bezpečné úložiště,<br>sdílená SQL |
| Geolokace                 | Ano               | Ano         | Ano                               |

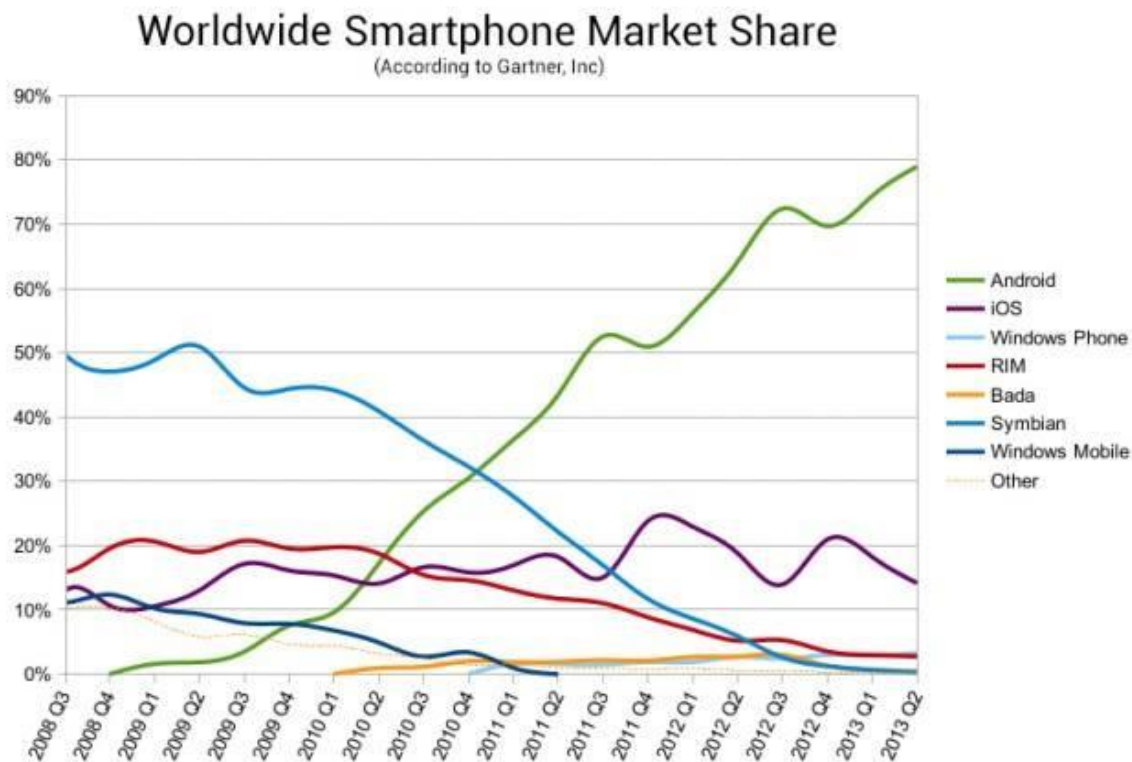
Tab. 3. Interaktivní vlastnosti jednotlivých mobilních aplikací

| <b>Gesta</b>               | <b>Nativní</b>  | <b>Webové</b>             | <b>Hybridní</b>           |
|----------------------------|-----------------|---------------------------|---------------------------|
| Swipování                  | Ano             | Ano                       | Ano                       |
| Složité animace            | Ano             | Ne                        | Ano                       |
| <b>Konektivita</b>         | Online, offline | Online                    | Online, offline           |
| <b>Nezbytné zkušenosti</b> | Obj C, Java, C# | HTML5, CSS,<br>Javascript | HTML5, CSS,<br>Javascript |

V nynější době se mobilní aplikace vyvíjejí převážně dvěma způsoby, a to právě hybridním a nativním způsobem. Výhodou nativních je perfektní uživatelské zkušenost a přívětivost, nevýhodou je vývoj mobilní aplikace pro každou platformu zvlášť. Pro hybridní naopak snížení nákladů při vývoji, tedy vyvinout stejné jádro pro všechny platformy a na konci jej upravit pro každou platformu zvlášť, což také není úplně jednoduché a občas se stane, že délka vývoje trvá stejně dlouho, ne-li déle, než při vývoji nativních aplikací pro každou platformu jednotlivě. Největší nevýhodou hybridní aplikace je její rychlost, která nikdy nedosáhne rychlosti nativní. Např. při rotaci na low-endových telefonech může způsobovat řadu problémů, které se odhalí až v průběhu vývoje nebo až uživatelé začnou hlásit problémy na zařízeních, které vývojáři neměli možnost pořádně otestovat. Také vzhled hybridní ani webové aplikace se nebude rovnat nativnímu vzhledu, přestože existují frameworky, které pracují s „napodobeninami“ nativních komponent. Momentálně můžeme říci, že vývoj hybridních aplikací je na vzestupu a že se dostal do bodu, kdy je použitelný i pro komerční sféru, ale stále nedokáže využít plný potenciál mobilního zařízení. Zda někdy dožene nativní aplikace, zjistíme až v budoucnu, protože prozatím jim nejsou schopny plně konkurovat z hlediska uživatelské přívětivosti. [15][16][18]

## 2 OPERAČNÍ SYSTÉM ANDROID

Operační systém Android je nejpoužívanějším a nejprodávanějším mobilním operačním systémem na světě. Ve druhém období loňského roku 2013 dosáhla téměř 80% na celosvětovém trhu se smartphony (podle průzkumu agentury Gartner, Inc., viz. Obr. 2). [1]



Obr. 2. Celosvětový podíl operačních systémů na smartphonech za 2013 Q2 [1]

Uživatelé chytrých zařízení mají OS Android ve velké oblibě nejen kvůli propojení s Google aplikacemi, ale také díky sociálním sítím jako je Facebook nebo Twitter. Uživatel si může vybírat nyní již z milionů aplikací na Google Play, které jsou ve většině případů zdarma.

### 2.1 Historie OS Android

Firma Google Inc. je známa především díky internetovému vyhledávači Google, nicméně vznik základů operačního systému Android sahá až do roku 2003, kdy tehdy společnost Android Inc. byla založena a vytvořila základy OS Android tak, jak ho dnes známe. V roce 2005 byla odkoupena právě firmou Google Inc., který z ní udělal dceřinou společnost.

V roce 2007 vznikla Open Handset Alliance<sup>4</sup> (dále jen OHA), jejímž cílem bylo vyvinout otevřený standard pro mobilní zařízení. OHA se skládá z pěti velkých skupin:

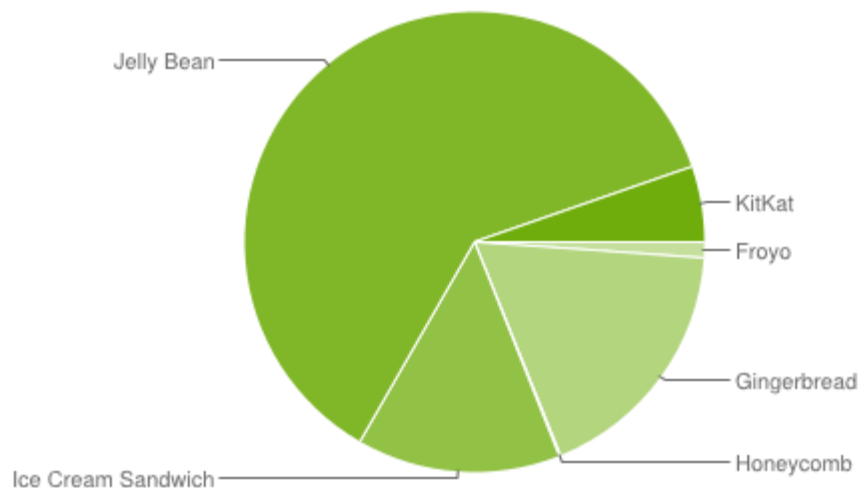
- Mobilní operátoři (např. T-Mobile, Telefónica, Vodafone, AT&T...)
- Výrobci mobilních telefonů (např. Acer Inc., HTC, Huawei, Motorola, Samsung...)
- Výrobci polovodičových součástek (Intel Corporation, MediaTek, NVIDIA, Qualcomm...)
- Softwarové společnosti (eBay Inc., Google Inc., ...)
- Obchodní společnosti

Ten samý den OHA ohlásil svůj první produkt, Android. Jak bylo napsáno výše, první verze OS Android byla vydána v USA na podzim roku 2008 a zároveň s tím bylo uvolněno SDK 1.0.

Od roku 2008 již vyšlo několik verzí OS Android a můžeme říci, že právě kvůli tomu je platforma značně fragmentována a vinou toho velice často kritizována. Každá nová verze tohoto OS má za úkol odstranit bezpečnostní chyby, vždy přináší nové funkcionality a snaží se být více uživatelsky přívětivá. Aktuální verze je 4.4.2 KitKat, kterou disponuje momentálně 5,3% vlastníků mobilního zařízení s Androidem (viz. Obr. 3). Google uveřejňuje tyto statistiky (viz. Tab. 4) vždy po třech měsících. Od verze 4.0 můžeme říci, že je tato platforma určena jak pro mobily, tak i pro zařízení s větší velikostí obrazovky (tablety, chytré televize). [10][12]

---

<sup>4</sup> Open Handset Alliance je konsorcium, které zahrnuje společnosti např. Google, HTC, Intel, LG, Motorola, NVIDIA, Qualcomm, Samsung, Texas Instruments a dalších 25 společností.



Obr. 3. Podíl Android verzí na trhu [12]

Tab. 4. Přehled Android verzí na trhu podle procent [12]

| Verze            | Kódové jméno       | API | Rozšířenost |
|------------------|--------------------|-----|-------------|
| 2.2              | Froyo              | 8   | 1.0%        |
| 2.3.3 – 2.3.7    | Gingerbread        | 10  | 16.2%       |
| 3.2              | Honeycomb          | 13  | 0.1%        |
| 4.0.3 -<br>4.0.4 | Ice Cream Sandwich | 15  | 13.4%       |
| 4.1.x            | Jelly Bean         | 16  | 33.5%       |
| 4.2.x            |                    | 17  | 18.8%       |
| 4.3              |                    | 18  | 8.5%        |
| 4.4              | Kitkat             | 19  | 8.5%        |

## 2.2 Architektura OS Android

OS Android je otevřená platforma využívající vespělý hardware a software, vlastní i sdílená data. Při vývoji OS Android byl kladen důraz na jednoduchost tvorby aplikací. Vývoj pro OS Android je pro vývojáře do značné míry usnadněn, co se týče bezpečnostních prvků. Celý operační systém byl navržen tak, aby i neznalost vývojáře v oblasti bezpečnosti nebyla velkou překážkou a vývojář tak mohl vyvinout aplikaci, která bude bezpečná. [2]

OS Android je proto rozčleněn do několika vrstev, což zajišťuje nejen pružnost platformy, ale i dostatečnou ochranu jejích uživatelů. [10]

### 2.2.1 Základní členění OS Android

- **Linux Kernel**

Je nejnižší vrstvou architektury. Jejím úkolem je správa systémových zdrojů. To je kromě telefonního modulu, GPS či fotoaparátu, například starost o volnou paměť RAM, vnitřní paměť a samozřejmě i kontrola aplikací. [3]

- **Knihovny**

Tato vrstva reprezentuje nativní Android knihovny, jako jsou OpenGL, SQLite, SSL atd. Většinou jsou napsány v C/C++.

- **Application Framework**

Tato vrstva je určena pro řízení hlavních bloků, ze kterých se skládá Android aplikace. Na této úrovni můžeme nalézt softwarové moduly jako *Activity Manager*, *Resource Manager* a *Content Provider*.

- **Aplikace**

Vrstva reprezentující uživatelské aplikace, jako prohlížeč nebo kontakty.

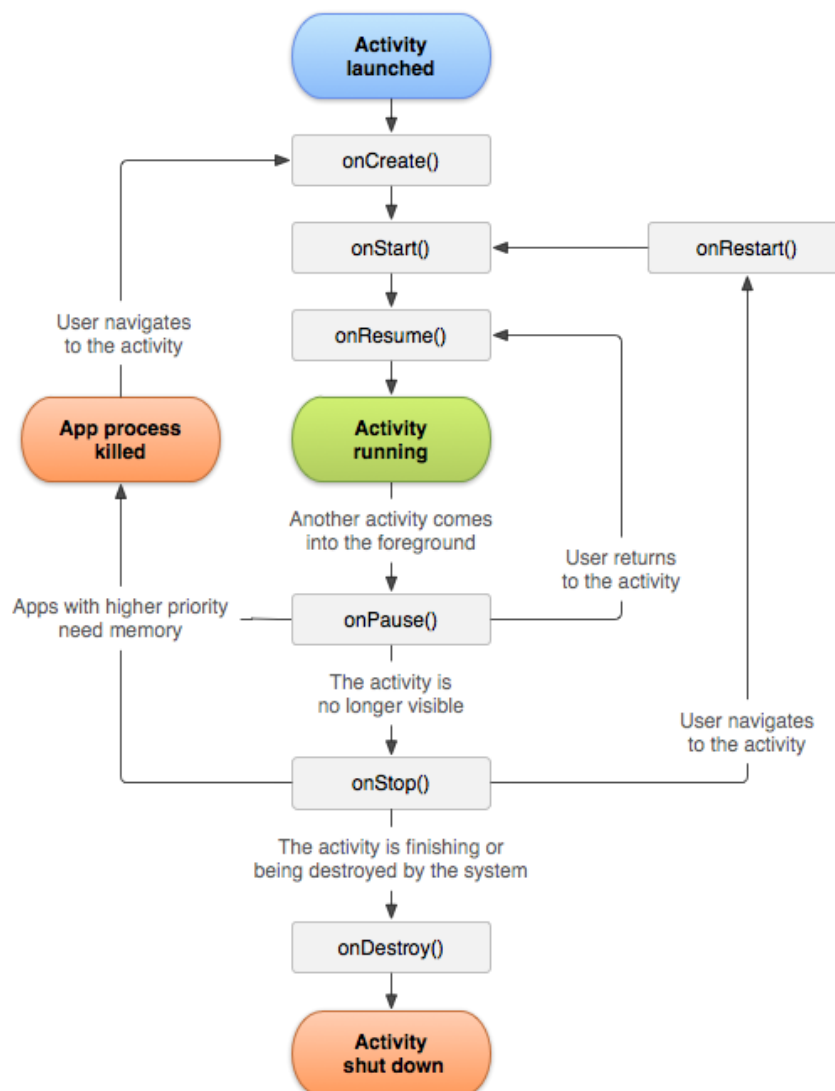
### 2.2.2 Struktura Android aplikace

Android aplikace jsou v podstatě zkomprimované soubory s příponou .apk. Tento soubor obsahuje [10]:

- **Zdroje** (suroviny) potřebné pro běh aplikace (tj. audio, rozložení jednotlivých komponent v .xml, obrázky, menu...)
- Soubor **classes.dex**, který je spouštěcí soubor tzv. Dalvik Virtual Machine, který dělá několik věcí:
  - Kompiluje všechny Java třídy, které jsou obsaženy v aplikaci pomocí Java Virtual Machine.
  - Převádí soubory z Java byte code z předchozího bodu do Dalvik byte code. Dalvik je podobný Java Virtual Machine (JVM), ale optimalizován pro mobilní zařízení, které může mít omezenou paměť nebo výkon.
- **Manifest.xml** definuje hlavní komponenty aplikace (často známé jako *entry-points*), povolení k užívání aplikace (tj., žádá o WiFi, posílání sms...) a také speciální akce používané k aktivování specifických komponent (intents).

Hlavní komponenty aplikace jsou definovány jako [4][22]:

- **Activity**, které mohou reprezentovat jednotlivé obrazovky v aplikaci. Princip je velice podobný webovým stránkám. Uživatel může přepínat mezi aktivitami uvnitř stejné aplikace anebo také mezi aktivitami jiných aplikací. Např. kontakty, obrazovka „Hello World“ nebo mp3 přehrávač, toto všechno jsou příklady aktivit.



Obr. 4. Životní cyklus aktivity [19]

- **Services** jsou částí aplikace, která běží na pozadí a uživatel o nich vůbec nemá tušení. Příklad *service* může být přehrávač se hudba, zatímco uživatel brouzdá na internetu pomocí prohlížeče.
- **Broadcast Receivers**, neboli můžeme říci i posluchače. Čekají na nějakou událost, po které dojde k aktivaci určitého kódu, např. když nám někdo volá,

vyskočí aktivita, umožňující vzít nebo zavěsit hovor. Dalším takovým *broadcast receiverem* je třeba kontrola nových e-mailů v případě zapnutí internetového připojení.

- **Content Providers**, velice často komunikují s databázemi a jsou nezbytné při poskytování dat dalším aplikacím. *Content providery* jsou tedy nějakí poskytovatelé obsahu, které dokážou předat data dalším aplikacím. Aplikace, které mají implementovaný *content provider* jsou např. nativní aplikace sloužící v telefonu k uchovávání tel. čísel a sms. Další aplikace se pomocí tzv. *Content resolveru* může připojit k aplikaci, která poskytuje *content provider* a získat tak např. zmiňovaná telefonní čísla a sms zprávy.

## 2.3 Rozšíření základu OS Android

Google poskytuje rozšíření základní verze OS dvěma způsoby – přeinstalované aplikace a aplikace instalované uživatelem.

### 2.3.1 Předinstalované aplikace

OS Android obsahuje řadu již přeinstalovaných aplikací, které jsou uživateli hojně využívány (email, kalendář, prohlížeč, kontakty atd.). Tyto programy jsou klíčové aplikace, protože mají přístup k hlavním funkcím telefonu. Předinstalované aplikace mohou být součástí přímo OS Android nebo jsou rozšířením některého z výrobců telefonů daného zařízení a často je nejde ani odinstalovat. [5]

### 2.3.2 Aplikace instalované uživatelem

OS Android podporuje vývoj aplikací třetích stran, které jsou s většinou pravděpodobností vždy umístěny na Google Play. K dnešnímu dnu lze v Google Play nalézt více než milion aplikací od různých firem a vývojářů.

Celý operační systém byl navržen jak s ohledem na vývojáře, tak především na uživatele, protože jsou to právě uživatelé, kteří mají přehled nad tím, jak aplikace fungují a které funkce využívají. Google při návrhu OS Android předpokládal, že útočník bude využívat běžné útoky (např. sociální inženýrství) k tomu, aby donutil uživatele nainstalovat škodlivou aplikaci do svého zařízení, a také že bude využívat aplikace třetích stran. Snahou Googlu je tedy minimalizovat šanci jakéhokoliv útoku,

popř. dopady, kdyby některý z útoků byl úspěšný. (více o této problematice v následující kapitole)

### 2.3.3 Google Play

Google Play (dříve Android market) je cloudové úložiště obsahující nepřeberně množství aplikací. Je to internetový obchod, pomocí něhož mohou uživatelé snadno vyhledat a stáhnout, popř. i koupit aplikaci a následně ji stáhnout do svého zařízení. Google play poskytuje také hodnocení jednotlivých aplikací, licenční ověření a bezpečnostní služby. Na Google Play může své aplikace nahrát každý, kdo si zaplatí (25 dolarů) a založí vývojářský účet. Aplikace je poté kontrolována speciálním programem (robotem), která kontroluje aplikaci na podezřelé či škodlivé chování. To je prováděno pomocí tzv. „robotů“, není tomu tedy stejně jako u Apple iOS, kdy aplikaci testuje fyzická osoba. To může být velká nevýhoda, neboť útočník může vytvořit nějakou aplikaci, která se uživateli jeví jako naprosto nevinná např. poznámkový blok a přitom žádá o speciální práva, která pro danou aplikaci ani nepotřebuje. Každá aplikace v Google Play má povolena určitá přístupová práva tzv. *permissions*, která definují, k jakým funkcím telefonu bude mít aplikace po nainstalování přístup. Toto ověřující procedura po vložení aplikace do Google Play vývojářem nemusí odhalit a po stažení této aplikace (např. tapety na plochu) může dojít ke škodlivému chování a zneužití dat (např. shromažďování tel. čísel a e-mailů a posílání těchto dat na jiný server).

### 2.3.4 Aktualizace

Jak už bylo napsáno výše, tak aktualizace představují z hlediska bezpečnosti mobilního zařízení důležitý faktor. Kromě úpravy UI a přidávání některých funkcí, dochází velice často k opravě nalezených bezpečnostních chyb a mezer v předchozích verzích a přispívají tak k celkovému lepšímu zabezpečení zařízení. Pro uživatele je tedy doporučeno aktualizovat, pokud je aktualizace k dispozici. V momentě, kdy Google vydá novou verzi systému Android, tak výrobci chytrých zařízení si ji upraví pro jednotlivé modely podle UI, které je pro danou značku typické a mnohdy i přidají některé softwarové vylepšení. Z tohoto důvodu taky aktualizace na zařízení, které není

z tzv. Google Edition, trvá mnohem delší dobu a navíc starší modely nemusejí být aktualizovány vůbec.

Existují dva způsoby aktualizace Android OS. Prvním z nich je tzv. Over The Air<sup>5</sup> (dále jen OTA), kdy se jedná o aktualizaci přímo z nabídky „Nastavení“ na daném zařízení. Aktualizace se stáhne a zařízení se samo aktualizuje na novější verzi. Druhou možností je vydání tzv. firmware od jednotlivých výrobců chytrých zařízení. Tento firmware je třeba stáhnout nejprve do PC, rozbalit a postupovat podle předem předepsaného postupu od výrobce, aby nedošlo ke ztrátě informací a aktualizace proběhla bez problémů. [6]

---

<sup>5</sup> <http://overtheair.org/blog/category/about/about-over-the-air/>

### 3 BEZPEČNOST OPERAČNÍHO SYSTÉMU ANDROID

Kapitola o bezpečnosti mobilní platformy Android z akademického pohledu by vydala na samostatnou knížku a ani potom bychom možná nebyli u konce. V následujících odstavcích proto bude uveden přehled bezpečnosti systému z pohledu uživatele, tím je myšleno jakým způsobem se může uživatel dostat do kontaktu se škodlivým softwarem, jak se proti němu bránit a také je popsán princip vybraného škodlivého software, se kterým se může uživatel setkat.

#### 3.1 Mýty o bezpečnosti Androidu

Kvůli otevřenosti a fragmentaci celého systému Android je většina lidí přesvědčená, že se do systému mohou bez uživatelského přičinění dostat viry, které napáchají nejrůznější škody. Pravděpodobně největší pověra je, že právě otevřenost celého systému stojí za většinou problémů škodlivých aplikací. Ovšem, jak už bylo popsáno v kapitole 2., vývojáři Androidu mysleli na jeho bezpečnost už při budování základů celého operačního systému. Mechanismus spouštění jednotlivých aplikací je totiž rozdílný oproti klasickému počítači a ze svého pohledu prakticky neprůstředný. Slabým článkem tohoto řetězce je ovšem ve většině případů sám uživatel, kdy jeho nezodpovědnost a občas i laxní přístup výrobců zařízení nebo operátorů může vést k problémům. Jsou známy kauzy typu *Carrier IQ* [7], které se přezdívá největší „špionážní“ skandál v historii. Bylo odhaleno, že desítky milionů (především Androidích) mobilů hlásí mobilním operátorům prakticky všechno, co s nimi uživatelé dělají (např. obsah SMS zpráv). Další kauzou může být bezpečnostní díra u telefonů firmy HTC [8], kdy výrobce implementoval aplikaci *htcloggers*, která sbírá nejrůznější citlivá data (např. pozice GPS, vytáčená čísla, údaje z uživatelských účtů, systémové výpisy ...) a všechny je umísťuje do jednoho souboru. Jiná aplikace, která má oprávnění přistupovat k internetu a čtení souborů v úložišti, pak může tento soubor číst a data odesílat směle do Internetu. Opět se ale jedná o chybu výrobce, nikoliv systému Android.

Problémy se tedy nejčastěji týkají aplikací, které do telefonu dodávala třetí strana. Pochybení tudíž nebývají způsobeny přímo v operačním systému. Samozřejmě, že i v OS Android se vyskytují chyby, které se snaží Google (či sdružení OHA) rychle řešit a eliminovat vydáváním potřebných bezpečnostních záplat, ale v očích veřejnosti je v tom mnohdy zcela nevinně.

## 3.2 Každá aplikace ve svém vlastním adresním prostoru

Jak bylo popsáno v kapitole 2., Android je založen na Linuxovém jádře a podobně jako u Windows je jeho hlavní podstatou správa systémových zdrojů. Každá aplikace je při svém běhu umístěna do tzv. Sandboxu, který přesně definuje každé aplikaci její hranice a zajišťuje bezpečnost mezi aplikacemi. Každá aplikace má k dispozici pouze určitou část paměti, kde si může zapisovat svoje potřebné údaje. Nikde jinde zapisovat nemůže, ani nikam jinam nemůže zasahovat nebo číst data ostatních aplikací. Ze svého adresního prostoru se tedy dostat nemohou. Bezpečnost aplikačního Sandboxu stejně jako ostatní bezpečnostní prvky OS Android není neprolomitelná. Aby ho však bylo možné obejít, bylo by potřeba prolomit bezpečnost Linuxového jádra. [22]

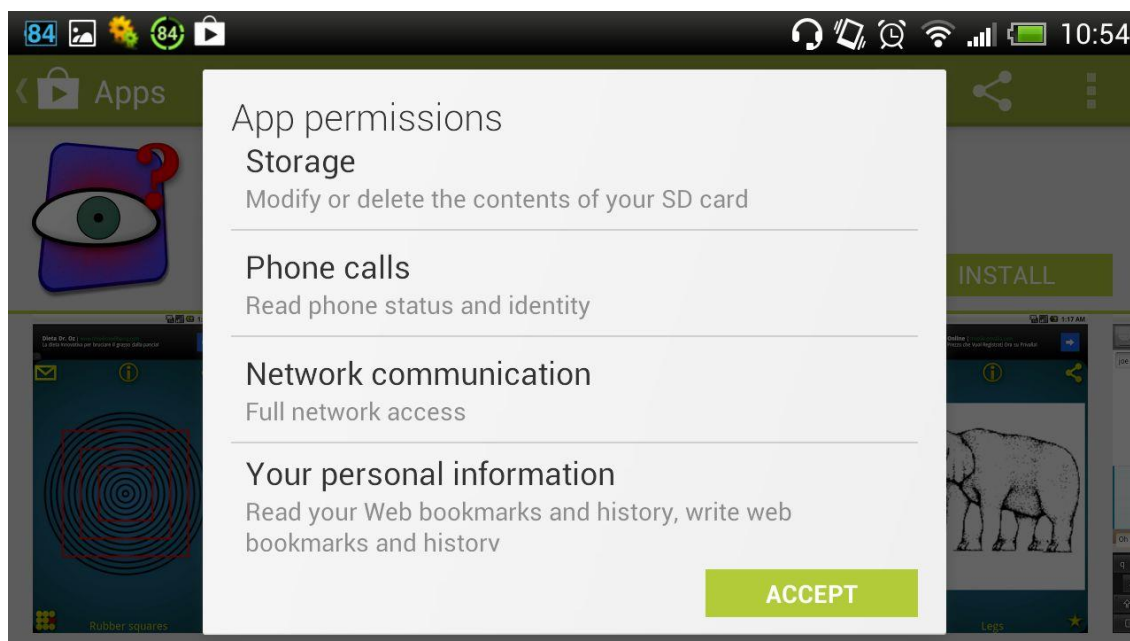
### 3.2.1 Povolení přístupu k systémovým zdrojům

Pokud některá aplikace chce získat přístup ke speciálním funkcím, tj. např. GPS poloze, čtení SMS, internetu, musí o to přímo požádat operační systém. Ten o tomto požadavku, zda umožní přístup k systémovým zdrojům, rozhodne na základě „permissions“, které jsou definovány vývojářem v souboru *Manifest.xml*, viz. Obr. 5.

```
<uses-sdk android:minSdkVersion="5" android:targetSdkVersion="5" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
```

Obr. 5. Povolení, pomocí kterých aplikace žádá k používání systémových zdrojů

Tyto stejné povolení musí každý uživatel schválit před instalací každé aplikace z Google Play. Na Obr. 6 můžeme vidět, že aplikace, které se tváří jako nevinná logická hra, vyžaduje přístup k externímu úložišti, dále k telefonním hovorům, síťové komunikaci, k osobním informacím. Leckterý uživatel si položí otázku: „Proč to ta aplikace vyžaduje?“ Uživatel by takovou aplikaci raději neměl instalovat, protože je jasné, že není něco v pořádku. Problém nastává, když si uživatel *permissions* ani nečte a rovnou instaluje aplikaci, protože si chce spustit tu logickou hru. [10]



Obr. 6. Příklad povolení vyžadující aplikace před instalací v Google Play [21]

Operační systém se tedy řídí rozhodnutím uživatele a vždy povolí, či nepovolí přístup k systémovým zdrojům aplikací podle přání uživatele. Zde se dostáváme ke slabému článku předchozího odstavce a tím je uživatel. Postup, kdy uživatel ani nečte seznam povolení, které vyžaduje aplikace, by se dal pochopit u známých aplikací a jejich vývojářů, ale např. u takto neznámé logické hry, jako je na Obr. 6. Je velice nebezpečné tuto aplikaci vůbec instalovat a doufat, že systémové zdroje vůbec nebude využívat. Takto totiž můžeme do svého zařízení nainstalovat hru, která nám na pozadí sbírá citlivá data a posílá na cizí server nebo v horším případě odesílá SMS na placená čísla a uživatel to vůbec nepozná.

Uživatel by tedy měl před každou instalací a aktualizací jednotlivé aplikace překontrolovat seznam oprávnění, zdali se na něm nenachází podezřelá položka. Její rozpoznání je velice snadné. Když aplikace určená k jednoduchému úkonu, např. již zmiňovaná logická hra, bude vyžadovat přístup k vytáčení čísel, čtení zpráv, tak není něco v pořádku. Naopak takový textový editor nebo kalendář určitě bude potřebovat přístup ke zprávám, kontaktům a podobným údajům. Pokud se uživateli tedy něco nezdá, může zkusit najít alternativu na Google Play, ve většině případů jich najde spoustu.

Jsou ale i aplikace, které vyžadují na první pohled podezřelá oprávnění, a ve své podstatě nadělají více užitku, než nepořádku. Jsou to aplikace většinou velice úspěšné a jsou vlastněny různými vládními agenturami nebo obrovskými korporacemi, které se snaží bojovat např. proti zneužívání dětí, dětské pornografii a kriminalitě obecně. Mezi

takové aplikace spadá např. Skype patřící Microsoftu, který je znám svým postojem k boji s dětskou pornografií. Další podobnou aplikací je velice rozšířená WhatsApp, která úzce spolupracuje s Národní bezpečnostní agenturou v USA (NSA), která filtruje jednotlivé zprávy a je schopna ihned zakročit v případě podezření. Tyto druhy aplikací tedy mají svůj důvod využívat i systémové zdroje, které nebyly v původních verzích potřeba, ale protože je používá několik desítek až stovek milionů lidí, mohou odhalit velké množství případů trestné činnosti a pomáhají ochránit i spoustu lidských životů. [11]

### 3.3 Aplikace a viry

Šance, že by uživatel narazil na škodlivou aplikaci, je relativně malá, nicméně se vždy vyplatí přečíst seznam povolení minimálně z toho důvodu, že na Google Play může aplikace nahrávat opravdu každý a schvalovací proces od Googlu to nemusí odhalit, přestože každá aplikace je před svým vývojem testována. Vývojář se špatnými úmysly může např. dát k aplikaci podobnou ikonu známého titulu či stejný název a s trochou štěstí a uživatelovy nepozornosti dosáhnout svého cíle. Škodlivé aplikace jsou mnohdy staženy z neoficiálních či nelegálních zdrojů, kdy k bezpečnému kódu je „přibalen“ tzv. škodlivý kód. To můžou být trojské koně na posílání SMS atd. Velice často to jsou pirátské verze placených aplikací, kdy uživatel musí před její instalací povolit v nastavení položku instalovat aplikace, které nepochází přímo z Google Play. Operační systém tedy nechává zodpovědnost plně na uživateli.

### 3.4 Spouštění škodlivého kódu

Žádná aplikace se nemůže nainstalovat do paměti telefonu sama od sebe např. přes internetový prohlížeč, či spustit nějaký škodlivý kód například z otevření přílohy e-mailu či souboru PDF. Vždy je potřeba schválení uživatele. Můžeme tedy říct, že uživatel by měl mít vždy přehled o tom, co má v telefonu nainstalováno.

### 3.5 Použití antiviru

Na základě informací z předchozích odstavců můžeme říci, že použití antivirových programů na mobilní platformě Android nemá velký význam. Alespoň antiviry, které neustále prohledávají úložiště zařízení a snaží se naleznout škodlivé aplikace. Antivir má smysl pouze v případě instalace škodlivé aplikace, kdy jsou schopné tyto závadné aplikace rozeznat. Podobně jako u PC k tomu mají k dispozici databázi určitých

charakteristických rysů jednotlivých rodin malware, na základě kterých se snaží určit škodlivý software.

Tyto antiviry mnohdy nabízejí i sekundární využití, kdy mohou pomoci s lokalizací ztraceného či odcizeného telefonu, filtrovat hovory či SMS, mazat paměť v případě krádeže a spoustu dalších užitečných funkcí. Nevýhodou je, že jejich používání spotřebovává o něco více energie.

### 3.6 Cover technique - nejnovější praktika škodlivého softwaru

Technika zvaná „*cover technique*“, neboli zakrývací technika, je rozdělena do dvou fází. Každou fází plní jednotlivá aplikace, to znamená, že ke správné funkčnosti této techniky je potřeba mít nainstalovány dvě aplikace, které svým způsobem komunikují a na základě snímání uživatelské činnosti získávají citlivá data. Např. jedna aplikace může snímat obsah tzv. „*private fields*“ (pomocí broadcast receiveru odposlouchávat např. pozici, kde bylo na display kliknuto a tím zjistit požadovanou číslici), neboli soukromá textová pole, kde se zadávají hesla, čísla kreditních karet, piny atd. Tato první aplikace je schopna na základě nějakého předem daného rozmezí měnit hlasitost reproduktoru (řekněme v deseti krocích). Uživatel tedy zadává citlivé údaje, první aplikace na pozadí mění hlasitost reproduktoru a nic víc nedělá. Uživatel nic nepozná. Druhá aplikace, která je spuštěna taky na pozadí zase snímá hlasitost reproduktoru přes svůj vlastní broadcast receiver, který snímá jakoukoliv změnu hlasitosti reproduktoru. Díky této zakrývací metodě je druhá z aplikací schopna získat podle předem stanovených intervalů nastavování hlasitosti reproduktoru jednotlivé číslice 0-9. Tato data pak může klidně šířit dál internetem a nikdo to ani nepostřehne.

Samozřejmě mějme na paměti, že uživatel musí obě aplikace nainstalovat a tím opět souhlasit s povolením využívat systémové zdroje, o které aplikace při instalaci žádá. Nicméně v tomto případě i uživatelská obezřetnost může znamenat ztrátu citlivých údajů, protože kdo by podezřívá aplikaci, která např. je určena k přehrávání hudby, že bude zneužívat změnu hlasitosti reproduktoru k „*vysílání*“ signálu další aplikaci. Nicméně pravděpodobnost, že by uživatel měl nainstalované zrovna tyto dvě aplikace od stejného vývojáře, které jsou schopny spolu komunikovat takovým důmyslným způsobem, je velice nízká, ale toto riziko tady je.

### 3.7 Má význam tzv. „root“ zařízení?

Android uživatelé se často setkávají s tzv. root zařízením. Root je pojmenování pro systémové super uživatele, který má nejvyšší možné systémové oprávnění. Může tedy např. měnit soubory na systémovém oddílu apod. Tím, že uživatelé „rootnou“ zařízen, tak získají možnost spouštět aplikace jako super uživatel. Na většinu aplikací to nemá žádný vliv, jsou ale i aplikace, které bez rootu nemají téměř význam. S rootem ale např. pomocí aplikace *root explorer* můžeme odinstalovat původně přeinstalované aplikace od operátora či výrobce. Toto má význam pro nadšence, kteří chtějí dostat z chytrého zařízení doslova všechno a rádi by vyzkoušeli i jinou ROMku, než kterou poskytuje výrobce. Super uživatel pak může např. dělat i kompletní zálohy aplikací pomocí *titanium backup*, dělat screenshoty obrazovky, mazat na systémovém oddíle apod. [9]

Z hlediska bezpečnosti není „root“ zařízení vůbec doporučován. Některé aplikace poté mohou dostat přístup do systémových dat bez vědomí uživatele. Ten si sice může navolit přes aplikaci „superuser“, které aplikace mohou využívat „root“ přístup, ale není vždy zaručeno, že to bude platit pouze pro ty vybrané. Může dojít ke ztrátě informací, ba dokonce i ke znehodnocení chytrého zařízení. V případě, že by se na daném chytrém zařízení nacházela již aplikace, která by měla škodlivý kód spustitelný pouze v případě „root“ zařízení, uživatel to opět vůbec nemusí postřehnout a tato aplikace může nadělat velkou škodu a nemusí být vůbec odhalena.

## 4 ZPŮSOBY OPTIMALIZACE PRO RŮZNÁ ZAŘÍZENÍ A TYPY NA PLATFORMĚ ANDROID

Android díky linuxovému jádru umožňuje unikátní hardwarovou přenositelnost a uživatelé mohou libovolně měnit různé aspekty tohoto OS. Také vývojáři nejsou omezováni a mohou vytvářet různé nadstavby uživatelského rozhraní a využívat všechny funkce zařízení, které jsou v OS Android dostupné.

Nicméně tato volnost se začala negativně projevovat a to zejména vinou fragmentace jednotlivých verzí OS a tím došlo k postupné degradaci platformy a uživatelské přívětivosti, která představovala velké riziko pro samotný Google ve snižování oblíbenosti tohoto OS. Společnost Google se rozhodla na toto riziko zareagovat a vydalo sadu tzv. doporučených konvencí *Android Design Guidelines*<sup>6</sup>. Také jsou postupně vydávány a upravovány již existující verze tzv. *Support Libraries*, neboli knihovny zajišťující podporu starším verzím systému. Mají za úkol zajistit konzistenci komponent přidaných v nových verzích Androidu i pro jejich starší verze, které je původně neobsahovaly. Programátor je může do svého projektu naimportovat a s jejich pomocí zajistit stejný vzhled a funkčnost na verzích systému, pro které vyvíjí, ovšem mnohdy za cenu vyšší časové náročnosti vývoje. [12][14]

### 4.1 Android Design Guidelines

Samotný Google doporučuje, aby si každý nový Android vývojář přečetl před programováním své první aplikace tyto doporučené konvence, aby nedocházelo zbytečně k nespokojenosti uživatelů a aplikace kvůli tomu nedostávala za design a uživatelskou přívětivost špatné hodnocení. Všeobecně mezi Android vývojáři panují dvě rčení: „*Jsi na Androidu, ne na iOS, webu, ani na hodině výtvarné výchovy.*“ a „*Udělej aplikaci tak, aby se dala ovládat nose.*“ [13]

Často můžeme na Google Play vidět aplikace, které nedodržují konvence a standardy doporučené Googlem a ve většině případů mají i špatné hodnocení a stahovatelnost, přestože funkčnost aplikace je na výborné úrovni. Toto se děje z důvodu, že uživatelé jsou zvyklí na jisté ovládací prvky a defaultní design komponent dané platformy. V případě,

---

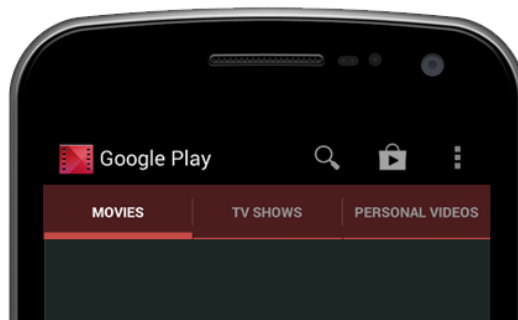
<sup>6</sup> Více na - <http://developer.android.com/design/index.html>

že daná komponenta, byť převzatá z jiné platformy, dělá na Androidu něco trochu jiného, působí na uživatele zmateným dojmem a často si na danou věc nezvyknou. Raději potom nainstalují konkurenční aplikaci. Proto konfrontace s iOS ve rčení z předchozího odstavce není náhodná. Na iOS vypadají některé komponenty stejně, ale mohou se chovat odlišně nebo jsou umístěny tam, kde by je Android uživatel neočekával. Na Obr. 7 můžeme vidět komponentu *bottom tab bar* používanou na iOS pro zobrazení možností na dané obrazovce.



Obr. 7. iOS bottom tab bar [4]

Na Obr. 8 zase můžeme vidět, jakým způsobem přistupuje k možnostem na dané obrazovce Android, který se snaží integrovat možnosti volby do horní lišty zvané *Action Bar* a *Action Tab* (mezi kterými může uživatel tzv. swipovat, nebo-li přejíždět ze strany na stranu prstem mezi prvky). Android též umožňuje použít *bottom tab bar*, jak můžeme vidět na Obr. 7, ale nedoporučuje to. [14]



Obr. 8. Android action bar včetně action tabs [4]

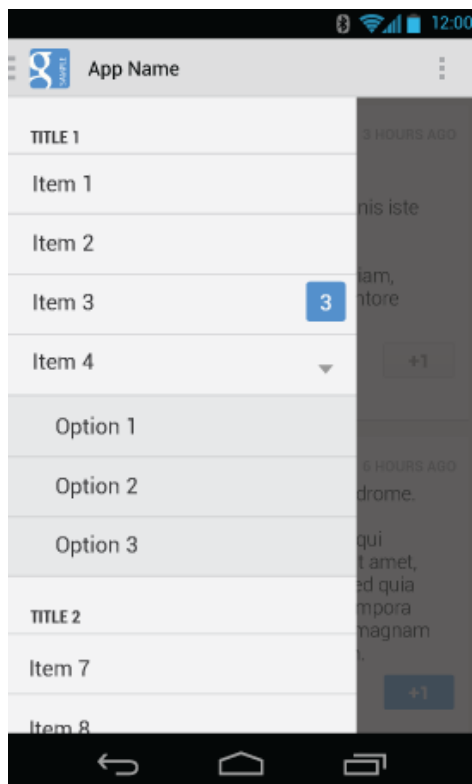
V nadcházejícím Obr. 9 lze vidět základní rozdíly mezi návrhem designu pro iOS a Android právě podle *guidelines* jednotlivých platforem.



Obr. 9. Přehled rozdílů v guidelines mezi iOS (vlevo) a Androidem (vpravo) [23]

Často se stává, že aplikace je naprogramována pomocí hybridního přístupu a nastylována nejprve pro iOS, ale následně pro Android je už řešena pouze její funkčnost, ale nikoliv design podle *guidelines*. Vzhled a ovládání poté nejsou uživatelsky zcela přívětivé. [9]

Rozdílů mezi jednotlivými komponentami a funkčností jednotlivých platforem je celá řada, nicméně postupem vývoje nových verzí jednotlivých OS se některé z nich postupně ucelují a fungují více méně podobně na každé z jednotlivých platforem. Pro příklad můžeme uvést, že začíná být téměř standardem, že každá větší aplikace obsahuje výsuvné menu s dalšími položkami (tzv. *sliding menu* nebo *navigation drawer*), viz. Obr. 10.



Obr. 10. Sliding Menu

## 4.2 Optimalizace různých velikostí a hustoty obrazovek

Od uvedení Androidu na trh je to již téměř 6 let. Od té doby šel vývoj dopředu a Android se dnes nepoužívá pouze na telefonech, jak bylo původně zamýšleno, ale i na mnohem větších zařízeních. Zařízení, které používají Android, mají nejen různé velikosti obrazovek, ale i různou hustotu pixelů. Pro upřesnění můžeme uvést několik ne tak běžných zařízení s Androidem: chytré hodinky, chytré brýle, domácí systémy (ledničky, termostaty atd.), systémy v autech (Ford), fotoaparáty, a dokonce i interaktivní zrcadla. [24]

Uživatelé očekávají, že aplikace bude fungovat korektně na všech různých typech obrazovek, a také doufají v její optimalizaci pro větší zařízení a využití volného prostoru.

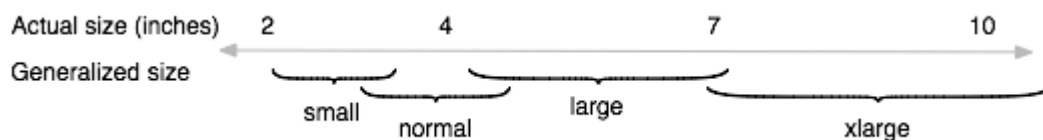
Verze Androidu 1.6 přidala nové funkce, které pomáhají k přizpůsobení různým rozměrům a rozlišením obrazovek. V dalších verzích byly tyto funkce ještě rozšířeny. Pokud je aplikace kompilována pro verzi systému Android 1.6 a vyšší, je automaticky předpokládáno, že obsluhu různých velikostí obrazovky provádí programátor sám a tím pádem systém nezapne *tzv. režim compatibility* (ten má za následek roztáhnutí rozvržení aplikace a tím pádem pro větší obrazovky dochází k jejich menší čitelnosti, zejména u obrázků, které mohou být rozmazané). [10]

#### 4.2.1 Prvky ovlivňující optimalizaci

- **Velikost obrazovky**

Je to skutečná fyzická velikost, udává velikost úhlopříčky v palcích. Android seskupuje všechny aktuálně dostupné velikosti displeje do 4 kategorií:

- small
- normal
- large
- extra large



Obr. 11. Pohled Android kategorizace velikosti displeje v palcích [25]

Pro rozvrhnutí komponent se v Androidu používá souborů .xml. Pokud se vzhled aplikace liší pro různé velikosti, je nutné pro správnou optimalizaci definovat zvlášť rozložení pro několik .xml souborů, jak můžeme vidět v Obr. 12.

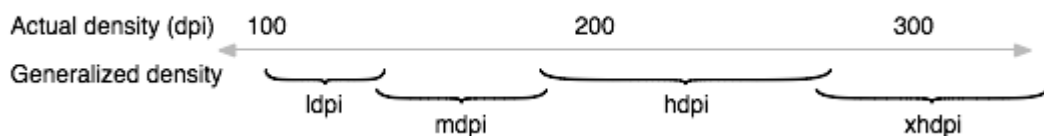
```
res/layout/my_layout.xml           // layout pro „normal“ obrazovku
res/layout-small/my_layout.xml     // layout pro „small“ obrazovku
res/layout-large/my_layout.xml     // layout pro „large“ obrazovku
res/layout-xlarge/my_layout.xml    // layout pro „extra large“ obrazovku
res/layout-xlarge-land/my_layout.xml // layout pro „extra large“
v orientaci na šířku
```

Obr. 12. Rozložení layoutů pro různé velikosti obrazovek

- **Hustota obrazovky**

Obvykle označováno jako dpi, neboli počet pixelů na čtvereční palec. Android seskupuje všechny aktuální dostupné hustoty displeje do 4 kategorií:

- low - ldpi
- medium - mdpi
- high - hdpi
- extra high – xhdpi



Obr. 13. Pohled Android kategorizace hustoty obrazovky [25]

- **Orientace**

Jsou dvě možné použití zařízení a to na výšku (portrait) či na šířku (landscape)

- **Rozlišení**

Udává celkový počet fyzických pixelů na displeji. Přidáním elementu `<support-screens>` do manifestu můžeme explicitně určit aplikaci, které rozměry obrazovek podporuje, a které ne (tzn. aplikace nepracuje s fyzickými rozměry v pixelech, ale v dp, viz. další odrážka). Ty, které nepodporuje, obslouží režim kompatibility, který je zmíněn v předchozím odstavci 4.2.

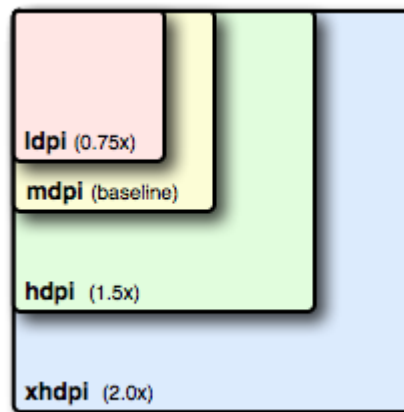
```
<support-screens
    android:anyDensity="true"
    android:largeScreens="true"
    android:normalScreens="true"
    android:smallScreens="true" />
```

Obr. 14. Element `<support-screens>`

- **Pixely nezávislé na hustotě obrazovky (dp)**

Vývojáři systému Android vytvořili tzv. *density independent pixel (dip nebo dp)*. Jedná se o virtuální jednotku, která se mapuje v měřítku 1:1 na reálné pixely obrazovky s hustotou bodů 160dpi a následně se škálují na základě tohoto poměru (tato hustota obrazovky odpovídá obrazovce HVGA a androidí skupině pro hustotu

- medium). Jednotka dp (dip) je tedy definována matematickým vztahem:  $1 dp = 160 px/dpi$ . Z tohoto vztahu plyne, že koresponduje s fyzickou velikostí displeje. Tato jednotka se pak na každém zařízení přepočítává zpět na pixely. Takže pokud se  $50 dp = 50 px$  na obrazovce s hustotou 160 dpi, na obrazovce s hustotou 240 dpi bude platit, že  $50 dp = 75 px$ . Výhoda těchto měrných jednotek pak spočívá v tom, že skutečné rozměry zůstávají stejné, takže mezi velikostí 50 dp při hustotě bodů 160 dpi a 50 dp při hustotě 240 dpi není žádný viditelný rozdíl. [13]
- **Obrázky** – U obrázků je optimalizace řešena modifikátory ve složce *resources*, protože klasické bitmapy (PNG, JPG, GIF) nejsou prakticky škálovatelné. Pro každou hustotu ldpi, mdpi, hdpi a xhdpi bychom měli poskytnout obrázek v jiném rozlišení, a to v poměru 3:4:6:8. Tedy, když má obrázek pro ldpi rozlišení 36\*36, pro mdpi bude 48\*48, pro hdpi 72\*72 a xhdpi 96\*96 px. Když nezadáme obrázky pro všechny hustoty, měla by se použít ta „nejbližší“. Hustoty pixelů mezi jednotlivými typy zařízení jsou samozřejmě i jiné hodnoty, než které jsou vyhovující pro poměr 3:4:6:8. Nicméně, Android daný obrázek zvětší tak, aby dané hustotě odpovídal. [10][13][25]



Obr. 15. Velikost bitmapy pro různé hustoty [25]

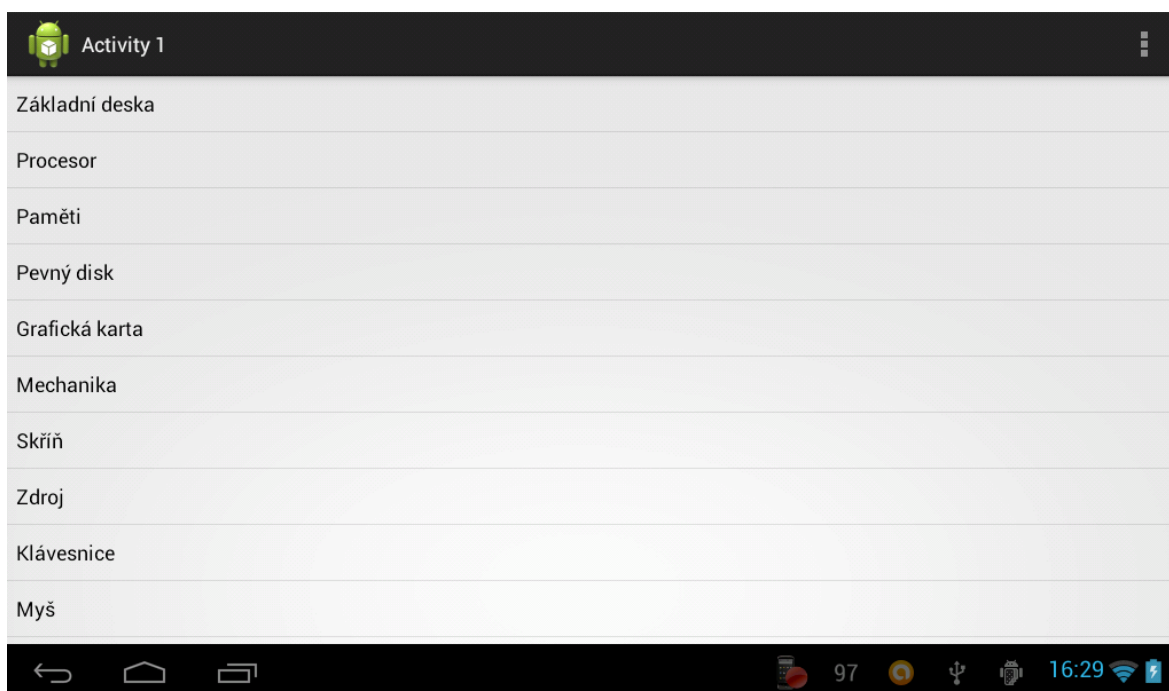
### 4.3 Tablety a větší zařízení

V roce 2011 vyšla verze Android 3.0<sup>7</sup>, která jako první oficiálně podporuje zařízení s většími obrazovkami, než jaké mají tradiční telefony, pro které byly navrženy starší verze

<sup>7</sup> Verze Android 3.0 oficiálně podporuje pouze tablety a větší zařízení a nebyla určena pro klasické telefony

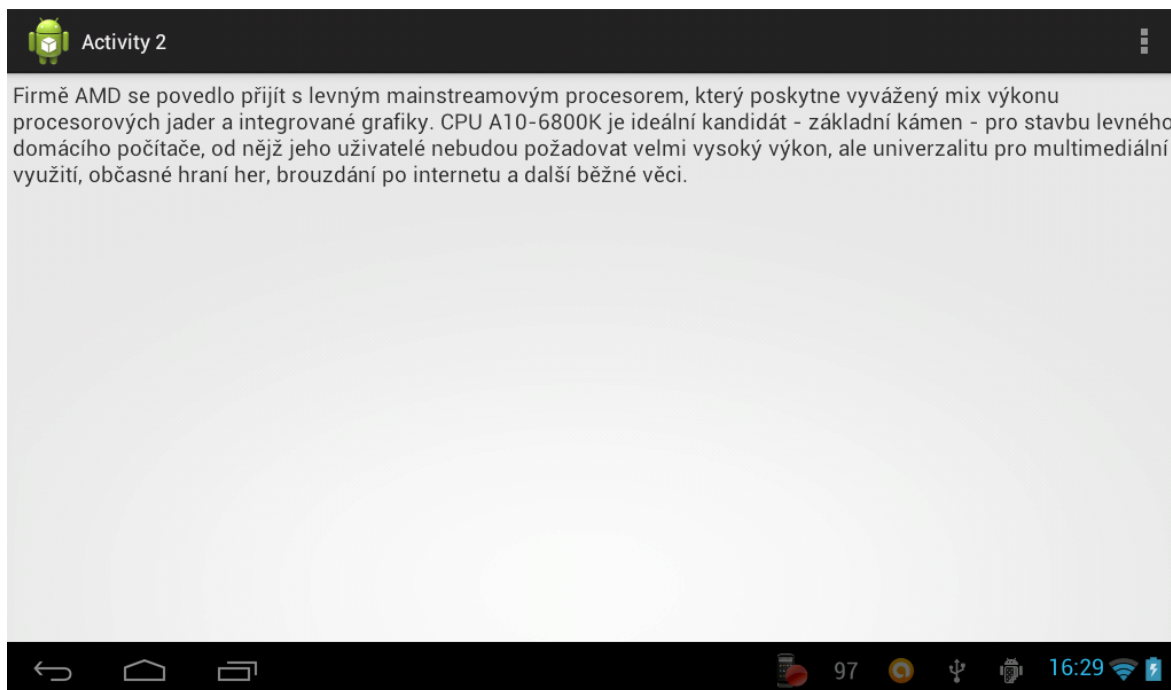
systemu. Verze Android 4.0 uvolněna o půl roku později sjednotila specifickou technologii uživatelského rozhraní verze 3.0 a tradiční technologie systému Android.

Původní UI před verzí 3.0 zaměřené na telefony umožňuje bez problémů obsluhovat i tablety způsobem, kdy se všechny navržené prvky UI zvětší tak, aby zabíraly potřebné místo. System tedy tuto úlohu zvládl, nicméně jeho použití na tabletech se projevilo jako nepraktické, protože docházelo ke zbytečnému plýtvání volného místa na obrazovce, viz následující Obr. 16, který zobrazuje pouze seznam nabídek, které po kliknutí na položku spustí další aktivitu obsahující detailní informace položky.



Obr. 16. Ukázka aktivity se seznamem nabídek na 7" tabletu

Na následujícím Obr. 17 je již zobrazena další aktivita obsahující detail položky „Procesor“ z Obr. 16



Obr. 17. Ukázka aktivity s detailem položky z předchozího obrázku na 7" tabletu

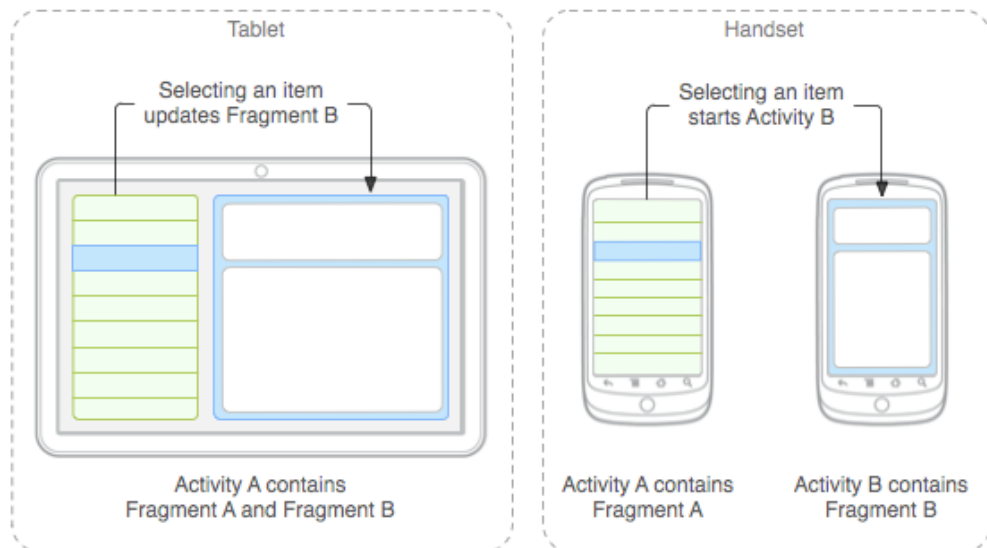
Zatímco na telefonu je použití nabídek vhodné, v tabletu již vhodné být nemusí, protože obrazovka tabletu nabízí více volného místa k zobrazení příslušných položek přímo na obrazovce. „Moderní“ verze systému Android jsou navrženy tak, aby zachovaly vzhled a funkce klasických verzí a současně nabízely aplikacím rozsáhlé možnosti využití volného místa na obrazovce. Tento způsob optimalizace se dělá v Androidu pomocí fragmentů, které jsou popsány v následující podkapitole.

### 4.3.1 Fragmenty

Fragmenty nejsou widgety, jako například objekty typu Button nebo TextView. Fragmenty ale nejsou ani kontejnery, jako například objekty LinearLayout nebo RelativeLayout. A fragmenty nejsou ani aktivity. Místo toho fragmenty slouží k agregaci widgetů a kontejnerů. Fragmenty lze poté umisťovat do aktivit (někdy i více než jeden fragment), jejich počet záleží podle rozměru obrazovek (tablety, televize, atd.). Lze tedy říci, že fragmenty jsou něco jako pod-aktivity.

Při použití fragmentů je každá oddělená komponenta uživatelského rozhraní umístěna do svého fragmentu. Aktivity, kterých se tato implementace týká, pak na základě velikosti obrazovky určují, která komponenta bude do fragmentu umístěna a která nikoliv. V následujícím Obr. 18 je v levé části vyobrazen návrh aplikace pro tablet. Ten je definován pouze jednou aktivitou obsahující dva fragmenty, kdy výběrem položky

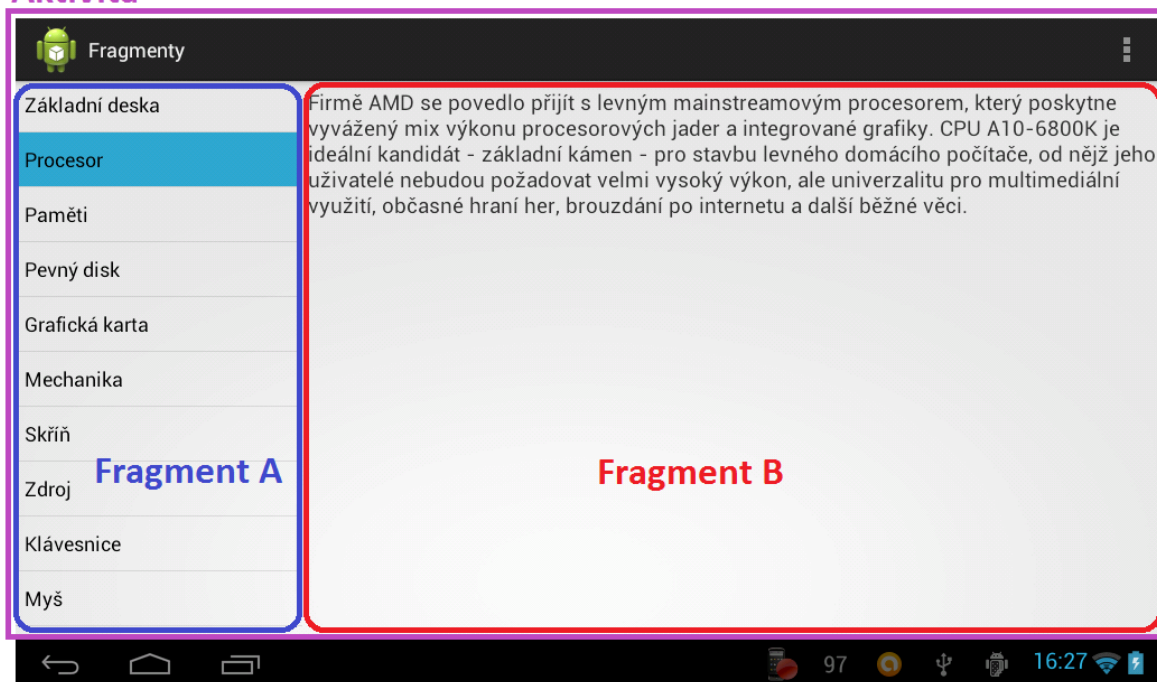
v seznamu fragmentu A je aktualizován detail té vybrané položky v pravém fragmentu B. V pravé části obrázku je vyobrazeno řešení pro telefon, kdy po vybrání příslušné nabídky ve fragmentu A (v aktivitě A) se otevře nová aktivita B s daným obsahem ve fragmentu B. [26][27]



Obr. 18. Příklad použití dvou UI komponent definovaných uvnitř fragmentů [26]

Na Obr. 19 můžeme vidět názornou ukázkou vyobrazení stejné aplikace, která byla ukázána v předchozí kapitole, ale již pomocí fragmentů, kdy rozložení hlavní aktivity je stejné jako v levé části Obr. 18. Aplikace se tedy skládá z jedné aktivity a dvou fragmentů.

## Aktivita

Obr. 19. Ukázka aplikace s využitím fragmentů na 7<sup>č</sup> tabletu

## 5 ZPŮSOBY ONLINE REZERVACE A MOŽNOSTI PLATEB PŘES MOBILNÍ APLIKACI

### 5.1 Online rezervace

Rezervace přes mobilní a webové aplikace probíhají pomocí síťové architektury klient-server, kdy klient (aplikace s GUI) žádá o službu jiný program zvaný server (velice často pracuje na serveru a může se jednat např. o přístup k databázi). Na server je vyslán požadavek pomocí webové služby a server vrátí odpověď.



Obr. 20. Architektura klient-server

V případě rezervačního systému může probíhat několik stejných požadavků na server v jednu dobu a server musí být schopen odpovědět a správně zareagovat na vzniklou situaci. Např. požadavek na rezervaci lístků do kina na stejné číslo sedadla může dojít z několika různých míst v republice současně. Server musí být schopen zaregistrovat a zapsat do databáze pouze první požadavek na rezervaci a dalším dát vědět, že požadované číslo sedadla je již obsazeno a k rezervaci nedošlo.

### 5.2 Mobilní platby

Mobilní platby mohou být nazývány také jako mobilní peníze, mobilní transakce nebo také jako mobilní peněženky. Jedná se o platby, při kterých je použit mobilní telefon

k uskutečnění bezhotovostních elektronických transakcí. Mobilní telefon za pomoci dalších technologií zaúčtuje částku transakce na úkor telekomunikačního nebo jiného účtu zákazníka. Výhodou oproti platbám přes internet je právě dostupnost mobilního zařízení, neboť prakticky každý disponuje mobilním telefonem a má ho stále u sebe. Výhodou oproti klasickému PC je tedy jeho dostupnost. Mobilní platby lze využít k platbě za širokou škálu služeb a zboží, např.:

- Hudba, video, vyzváněcí tóny, hry a další digitální produkty
- Doprava (MHD, vlak), parkování
- Knihy, časopisy, vstupné atd.

V současné době existuje několik hlavních směrů v mobilních platbách:

- Premium SMS
- Přímé zpoplatnění u operátora
- Mobilní webové platby

### 5.2.1 Premium SMS

V České Republice jsou prémiové SMS podporované všemi operátory. Jedná se o klasické SMS ovšem s vyšší tarifací, neboť jsou účtovány tzv. prémiovou cenou. V ČR se cena takové SMS pohybuje v rozmezí od 10-198,- Kč. Uživatel nakupující službu nebo zboží odešle SMS v definovaném formátu na definované číslo a obratem obdrží speciální kód, který použije pro získání přístupu k požadovanému zboží. Např. dobítí imaginárního kreditu na webových stránkách, pomocí kterých může koupit zboží. [28]

Tab. 5. Přehled premium SMS pro Českou Republiku [28]

| Cena (s DPH) | Cena (bez DPH) | Operátoři              | Zisk |
|--------------|----------------|------------------------|------|
| 198.0 (CZK)  | 163.64 (CZK)   | O2, T-Mobile, Vodafone | 55%  |
| 99.0 (CZK)   | 81.82 (CZK)    | O2, T-Mobile, Vodafone | 50%  |
| 79.0 (CZK)   | 65.29 (CZK)    | O2, T-Mobile, Vodafone | 45%  |
| 50.0 (CZK)   | 41.32 (CZK)    | O2, T-Mobile, Vodafone | 45%  |
| 30.0 (CZK)   | 24.79 (CZK)    | O2, T-Mobile, Vodafone | 40%  |
| 20.0 (CZK)   | 16.53 (CZK)    | O2, T-Mobile, Vodafone | 40%  |

| Cena (s DPH) | Cena (bez DPH) | Operátoři              | Zisk |
|--------------|----------------|------------------------|------|
| 10.0 (CZK)   | 8.26 (CZK)     | O2, T-Mobile, Vodafone | 35%  |

V Tab. 5 můžeme vidět přehled pro jednotlivé prémiové SMS na území České Republiky. Pro subjekty poskytující služby či prodej zboží je zisk v rozmezí 35-55% v závislosti na tarifaci jednotlivých SMS. Z uvedených údajů tedy vyplývá, že celkový zisk není vysoký, protože velkou část z ceny prémiové SMS si ponechávají sami operátoři. Platba je tedy spíše vhodná k placení drobného zboží či služby.

Nevýhody Premium SMS:

- Poměrně malá spolehlivost – platební transakce může selhat v případě ztráty/nedoručení sms
- Relativně nízká rychlost – odesílání a následné přijmutí sms může trvat relativně dlouho (záleží zejména na vytíženosti mobilní sítě). Potvrzení od obchodníka může trvat delší dobu, ale zákazníci však nejsou ochotni čekat déle, než několik sekund.
- Zabezpečení – zprávy nejsou šifrovány a např. na Androidu je může číst kterákoliv aplikace, má-li k tomu svolení uživatele
- Vysoké poplatky – jak je popsáno výše, velká část placené částky připadne operátorovi a jeho agregátorům, kteří platbu pro operátora zprostředkovávají

### 5.2.2 Přímé zpoplatnění u operátora

K přímému zpoplatnění u operátora dochází, když uživatel zaplatí obchodníkovi ze svého mobilního telefonu způsobem, že placená částka se zákazníkovi strhává přímo z kreditu, nebo je mu připočtena k jeho pravidelnému měsíčnímu vyúčtování v případě tarifu. Tato služba je zprostředkovávána v ČR všemi operátory.

Princip je jednoduchý: zákazník poskytne obchodníkovi své telefonní číslo, obchodník následně vyšle žádost o autorizaci pomocí SMS ve tvaru: „Obchodník XY žádá o autorizaci nákupu zboží YZ v ceně XYZ Kč“, zákazník poté potvrdí SMS zprávou zadáním PIN kódu a časově omezeného hesla a částka je mu stržena tak, jak je popsáno v předcházejícím odstavci.

Výhody:

- Zabezpečení – dvojitá autorizace jako prevence zneužití

- Pohodlí – není potřeba žádné registrace nebo instalace dalšího softwaru do zařízení
- Rychlost – transakce může být dokončena v několika sekundách

### 5.2.3 Mobilní webové platby

K zpracování mobilní webové platby je potřeba, aby zákazník použil pro platbu internetové stránky nebo speciální aplikace nainstalované do telefonu. Komunikace se serverem, kde je zpracovávána platba, probíhá pomocí WAP protokolu nebo přímo pomocí připojení přes WiFi, popř. mobilní internet. V současné době existují dva hlavní způsoby mobilních webových plateb, které jsou popsány v následujících bodech.

#### 5.2.3.1 Platební karty

Používání platebních karet je v dnešní době nejrozšířenější způsob internetových plateb vůbec. Existuje velké množství druhů platebních karet, ale v ČR se setkáváme prakticky pouze se dvěma, a to s debetními a kreditními kartami, které jsou pravděpodobně vydávány dvěma největšími asociacemi: VISA a MasterCard. Rozdělení platebních karet je dále na elektronické a embosované. Elektronické, které jsou v ČR zároveň nejrozšířenější, slouží ke klasickým transakcím pro výběr z bankomatu nebo platbě v obchodě např. za potraviny. Embosované karty se dají využít ke všem druhům on-line plateb včetně platby zboží přes internet.

V ČR nejsou platby přes internet přímo zajišťovány vydavateli karet, ale tzv. zprostředkovateli. Těmito zprostředkovateli jsou partnerské banky, které pomocí systému 3D Secure zajišťují bezpečnost. Např. banky jako ČSOB, KB a eBanka využívají platební bránu GPWebPay od společnosti Global Payments, zatímco Česká spořitelna využívá svoji vlastní platební bránu. Dalšími rozšířenými platebními bránami jsou v České republice např. PayU nebo GoPay, které poskytují poměrně snadnou implementaci do webových stránek obchodníka. Průběh kompletní realizace internetové platby můžeme vidět na Obr. 21.



Obr. 21. Kompletní cyklus internetové platby pomocí platební brány PayU [29]

Informace o kartě jsou přenášeny zabezpečeným protokolem HTTPS přímo bance. Pomocí systému 3D-Secure jsou tedy citlivé údaje o kartě poskytovány pouze bance, která si ještě před potvrzením platby vyžádá ověřující kód, který obvykle odešle na mobilní telefon kupujícího pomocí SMS. Banka pak sama kontaktuje obchodníka (přes platební bránu), zda transakce proběhla úspěšně, či nikoliv. Výhodou může být pro obchodníka to, že embosovanou platební kartou disponuje prakticky každý, nevýhodou pak poplatky zprostředkovatelům, které se pohybují od 2-4% za každou platbu a mnohdy je třeba ještě zaplatit počáteční poplatek. [29]

### 5.2.3.2 Elektronické peněženky

Elektronickou peněženku si můžeme představit něco jako bankovní účet. Je to online platební systém, který pracuje v reálném čase a podporuje i velice malé platby. Protože se jedná o online systém, jsou platby rychlejší a také levnější, než systémy popsány výše. Elektronická peněženka není přímo svázána s nějakým dalším bankovním účtem, ale může být dobytá odkudkoliv, tzn. z jakéhokoliv bankovního účtu, z jiné elektronické peněženky. Funkce el. peněženky tedy spočívá v určitém způsobu převodu peněz do peněženky, se kterými můžeme poté disponovat jako s reálnými penězi a po uskutečnění platby je prostředník sám převede danému obchodníkovi. V ČR jsou nejrozšířenější mobilní peněženky: PayPal, GoPay, PayPay, PaySec, Google Wallet, a dále třeba Mobito, které si získává čím dál větší oblibu. Nejvýznamnějším prvkem elektronických peněženek je jejich bezpečnost, na které se podílí několik dále popsaných faktorů. Uživatel si většinou do

peněženky nepřevádí všechny své peníze, ale pouze určitou část. Tím je eliminováno riziko, že uživatel přijde o všechny své peníze v případě jejího zneužití. Dalším faktorem je neposkytování citlivých údajů a zneužitelných dat při platbě. Lze také nastavit limit platby, kdy po jeho překročení je vyžadováno heslo k odemčení peněženky, popř. jsou potřeba další údaje poskytnuté třetí stranou. Do elektronické peněženky lze integrovat i platební a jiné bezkontaktní karty. Uživatel může přidat údaje o svých kartách a ty jsou poté emulovány přímo v peněžence. Při platbě si uživatel může vybrat, kterou kartou či zda chce platit kartou nebo kreditem přímo v aplikaci. Takto funguje např. Google Wallet.

V současné době existují dva způsoby použití elektronických peněženek. První způsobem je implementace komunikace s danou elektronickou peněženkou přímo do mobilní aplikace přes poskytované API. Přes tuto mobilní aplikaci pak může uživatel nakupovat zboží přímo v reálném čase, např. MHD jízdenky. Druhým způsobem je tzv. „*Tap&Go*“, který je založen na technologii NFC. Telefonem, který je vybaven touto technologií lze fyzicky u obchodníka, který poskytuje platby pomocí NFC, zaplatit jakýkoliv nákup pomocí elektronické peněženky jednoduchým přiložením telefonu k NFC modulu. Funguje to úplně stejně jako bezkontaktní platební karty, které jsou taky vybaveny NFC terminálem. Lze tedy tímto způsobem již bez problému platit i u nás v ČR. Platby přes mobilní telefon jsou ovšem oproti bezkontaktním platebním kartám mnohem bezpečnější, protože telefon je aktivní pouze při platbě, zatímco bezkontaktní platební karty jsou aktivní stále a může je číst prakticky kdokoliv, pokud disponuje NFC modulem a dostane se do vzdálenosti kratší než 20cm. [29][30][31]

Hlavní výhodou oproti předchozím platbám pomocí platebních karet jsou prakticky nulové náklady pro obchodníky za přijímání plateb.

## **II. PRAKTICKÁ ČÁST**

## 6 ANALÝZA

Na trhu je již k dispozici několik aplikací podobného zaměření, nicméně žádná z dosud vytvořených nedosáhla takového pozitivního ohlasu, aby nově přicházející aplikace s podobným zaměřením nepřevyšovaly svou kvalitou ty stávající. Cílem analýzy je porovnat dosavadní řešení na trhu, nalezení nedostatků a získání základní představy k vytvoření vlastního návrhu aplikace a její následné implementace. Protože byla zvolena platforma Android, zaměřil jsem se tedy pouze na aplikace dostupné pro tuto platformu.

### 6.1 Analýza již existujících řešení

Vybrané aplikace byly staženy a nainstalovány z oficiálního úložiště aplikací pro Android z Google Play a odzkoušeny na mobilním telefonu LG Nexus 4, tabletech Samsung Nexus 10 a Prestigio 5770D. Bylo nezbytné stanovit podmínky pro výběr aplikací:

- neplacená aplikace
- Android minimálně 2.3.3 a vyšší
- vzhled uživatelského rozhraní

#### Galeria Krakowska

Galeria Krakowska<sup>8</sup> je na první pohled v Google Play dobře vypadající aplikace, jejíž počet stažení se blíží 50 tisícům. Jedná se o poměrně známou aplikaci polského nákupního centra ve městě Krakov. Aplikace je udržována (poslední aktualizace 14. 1. 2014), nicméně její funkčnost tomu neodpovídá. Přestože má aplikace 4 hvězdičky, trpí řadou nedostatků. Prvním z nich je určitě design, přestože je navrhnut na míru, vypadá spíše jako hybridní mobilní aplikace s designem pro iOS a je pouze překlopená, aby fungovala i pro Android. Ovšem moc nefunguje, aplikace jako taková nejde ukončit klasickým způsobem, dialogové okno se při aktualizaci samo také neukončí a točí se stále dokola, přestože jsou již načtena aktuální data. I přes tyto nedostatky poskytuje aplikace přehled aktuálních slev a událostí, které se v nejbližší době uskuteční v centru. Nejsou implementovány žádné notifikace a design je uzamčen v módu na výšku. Dalším nedostatkem z mého pohledu je „*splash screen*“, který není Googlem doporučován a dle mého názoru zde naprosto

---

<sup>8</sup> K dispozici na <https://play.google.com/store/apps/details?id=pl.amistad.android.galeriakrakowska>

postrádá smysl, neboť trvá 5 vteřin a na pozadí ani neproběhne pokus o aktualizaci dat a pouze zobrazuje logo nákupního centra. Zda chce uživatel stáhnout aktuální data, se uživatele zeptá až po ukončení „*splash screenu*“. Dalším nedostatkem je určitě nemožnost jakéhokoliv nastavení. Aplikace se celkově jeví spíše jako špatný vtip, než funkčně užitečný pomocník a marketingový nástroj. OC obsahuje i kino, bohužel v aplikaci o něm není ani zmínka.



Obr. 22. Galeria Krakowska

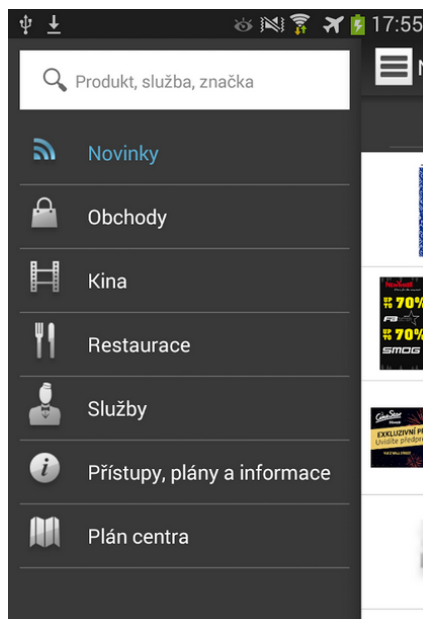
### Centrum Černý Most, Centrum Chodov

Jedná se prakticky o dvě různé aplikace, které jsou ale jinak úplně stejné, liší se pouze detaily jednotlivých obchodů obchodních center a orientační mapou. Vzhled aplikací Centrum Černý Most<sup>9</sup> a Centrum Chodov<sup>10</sup> je oproti předchozí aplikaci na lepší úrovni, a přestože se opět jedná o hybridní aplikaci, vzhled je částečně napodoben jako nativní Android aplikace a vypadá opravdu dobře. Ovšem i zde můžeme narazit na řadu problémů. Na telefonu aplikace běží bez problémů, ale není vůbec přizpůsobena tabletům a větším rozlišením. Obrázky na kvalitnějším displeji jsou rozmazané a např. úvodní „*splash screen*“ je neforemně roztáhlý na tabletech a vůbec nevypadá dobře. Během proklikávání aplikace 2x nepochopitelně spadla. Jako největší nedostatek hodnotím přehled kina, ke

<sup>9</sup> K dispozici na <https://play.google.com/store/apps/details?id=com.unibail.cernymost2>

<sup>10</sup> K dispozici na <https://play.google.com/store/apps/details?id=com.unibail.chodov2>

kterému, přestože podle popisu a screenů v Google Play je implementováno přímo do aplikace, v aplikaci nejsou žádná data i přes jejich aktualizace. Aplikace opět neposkytuje žádné nastavení ani notifikace a občas se lehce zasekne. I přes tyto nedostatky se jedná o dobrou aplikaci, která plní svůj účel a vypadá velice dobře a dokonce obsahuje možnosti sdílení obsahu přes sociální sítě atd.



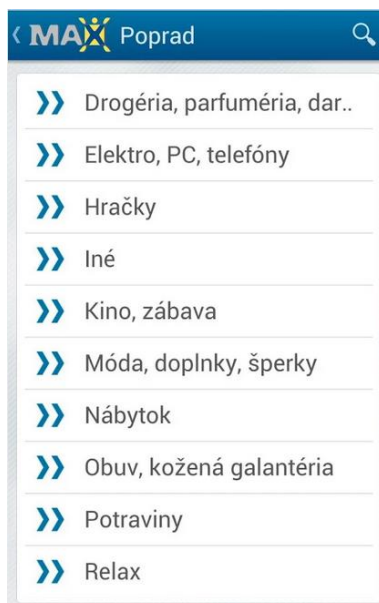
Obr. 23. Centrum Černý Most

### Max do vrécka

Jak už název napovídá, Max do vrécka<sup>11</sup> je slovenskou aplikaci pro zábavné obchodní centra Max na Slovensku. Jedná se o nativní aplikaci, která funguje plynule. Obsah aplikace je rozdílný pro různá obchodní centra, ale design a rozmístění ovládacích prvků a komponent je jednotný. Aplikace poskytuje údaje o událostech, slevách, dále informace o jednotlivých obchodech a mapu centra. Vše funguje bez problémů a v aplikaci je obsaženo jak nastavení, tak i notifikace pro vybrané OC. Jako hlavní nedostatek vidím to, že chybí jakékoliv aktuální informace z kina. Pro kino je uveden pouze informativní popis o kinu a cenách jednotlivých vstupenek, který najdeme v přehledu prodejen daného OC. Aplikace neobsahuje možnosti sdílení na sociálních sítích apod., ale zase funguje hezky i na tabletu a při otočení displeje „na výšku“.

---

<sup>11</sup> K dispozici na <https://play.google.com/store/apps/details?id=sk.originals.zocmax>



Obr. 24. Menu aplikace

Max do vrécka

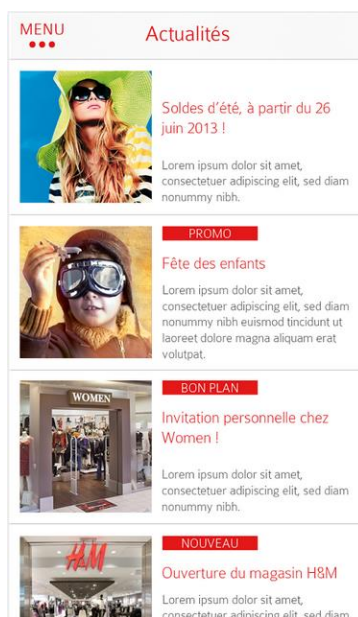
## Geric

Další aplikací, kterou jsem vybral je aplikace francouzského obchodní centra Geric<sup>12</sup>, které svou aplikaci uvolnilo teprve 11. 4. 2014 a je tedy absolutní novinkou. V době psaní této práce měla aplikace okolo 2 tisíc stažení s hodnocením 4 hvězdiček. Na první pohled se jedná o pěkně zpracovanou aplikaci, na které je vidět, že design byl prioritou. Nicméně, opět se jedná o hybridní HTML5 aplikaci, která se svou uživatelskou přívětivostí nemůže rovnat aplikacím nativním. V tomto případě jsem byl šokován reakcí mobilní aplikace na posuv prstu po obrazovce při tzv. scrollování, které jsou strnulé a neplynulé. Aplikace obsahuje výsuvné menu (tzv. sliding menu), které je ovšem implementováno jaké statické a nereaguje tzv. swipe gesto zprava doleva, když chce uživatel toto menu zavřít, ale musí kliknout na ikonu „MENU“ vpravo nahoře, která je z části překrytá hranou displeje. Aplikace je uzamčená pouze do módu na výšku a ne zcela přizpůsobená větším obrazovkám. Obrázky sice rozmazané nejsou, ale vzhled na 10“ tabletu vypadá u některých vybraných položek až komicky. Po designové stránce je aplikace ale na velmi vysoké úrovni oproti ostatním zmiňovaným aplikacím a určitě plní úlohu marketingového nástroje. Nejsou zde ovšem opět implementovány notifikace a jsou znát jisté

---

<sup>12</sup> K dispozici na <https://play.google.com/store/apps/details?id=com.myapphone.android.geric>

programátorské nedodělky, které aplikaci brzdí a nevyvolávají úplně příjemný pocit při jejím ovládní. Nejsou implementovány žádné možnosti sdílení.



Obr. 25. Přehled novinek  
v aplikaci Geric

### Zlaté jablko

Tato aplikace je opravdovou novinkou, na kterou jsem narazil při psaní této práce a rozhodl se ji přidat jako nejnovější aplikaci. Byla vydána dne 22. 4. 2014 a prozatím má pouze pod 50 stažení. Opět se jedná o hybridní HTML5 aplikaci, která obsahuje některé prvky z jiných platforem. Na první pohled aplikace vypadá zajímavě a také se chová tak, jak člověk očekává. Nicméně tato aplikace trpí řadou nedostatků. Podporuje pouze online režim, tzn., že veškerý obsah se stahuje podle jednotlivých požadavků uživatele. Nic není uloženo v mobilním zařízení a tím vzniká velká nevýhoda oproti předchozím popsaným, protože v aplikaci si uživatel může prohlížet jakákoliv data pouze, pokud je připojen k internetu. Obchodní centrum obsahuje i kino a aplikace dokáže zobrazit jeho aktuální program a nabízí i možnost rezervace. Při rezervaci jsme ovšem přesměrováni do nativního prohlížeče mobilního zařízení a rezervace probíhá stejně jako pomocí počítače. Chybí také možnost sdílení novinek a filmů. Na tabletu je aplikace velice nesourodá, obrázky zdeformované a rozmazané, protože nejsou přizpůsobené na větší velikosti obrazovek a rozlišení.

## 6.2 Hodnocení

Analýza různorodých aplikací na trhu ukázala nedostatečně kladený důraz na vzhled a kvalitu uživatelského rozhraní a celkově přinesla rozporuplné pocity o úrovni některých aplikací. Pokud pomineme funkčnost a rychlost aplikace, tak kvalitně zpracované uživatelské rozhraní má v dnešní době rozhodující význam při stahování a následném používání aplikace uživateli. Velkým zklamáním bylo vynechání aktuálního programu kina ve všech zmiňovaných aplikacích, kromě Zlatého jablka a nefunkčního programu u aplikace Centrum Černý Most. Dalším zklamáním pro mě byla u 4 z 5 testovaných aplikací chybějící podpora větších velikostí obrazovek, vynechání notifikací a možnost jakéhokoliv nastavení v aplikaci, které ovšem může souviset právě s notifikacemi. Žádná z aplikací nedokáže kontrolovat na pozadí v pravidelných intervalech vzniklé události v centrech, jako jsou třeba novinky nebo např. změna prodejen, restaurací atd. Otestování aplikací přineslo mnoho ponaučení, která se následně projevila v návrhu a implementaci. V Tab. 6 můžeme vidět porovnání jednotlivých aplikací, podle jednotlivých funkcí a vlastního pocitu z daných aplikací.

Tab. 6. Porovnání již existujících aplikací

| Funkce aplikace                              | Galeria<br>Krakowska | Centrum<br>Černý Most | Max do<br>vrécka | Geric | Zlaté<br>jablko |
|--|----------------------|-----------------------|------------------|-------|-----------------|
| Design podle Guidelines                      | -                    | •                     | •                | -     | -               |
| Schopnost pracovat offline                   | •                    | •                     | •                | •     | -               |
| Zobrazuje novinky, akce                      | •                    | •                     | •                | •     | •               |
| Schopnost notifikovat                        | -                    | -                     | •                | -     | -               |
| Zobrazuje detaily obchodů,<br>detaily centra | •                    | •                     | •                | •     | •               |
| Obsahuje program kina                        | -                    | •                     | -                | -     | •               |
| Možnost rezervace<br>vstupenek do kina       | -                    | -                     | -                | -     | -               |
| Sdílení obsahu (akcí ...)                    | -                    | •                     | -                | -     | -               |
| Schopnost pracovat                           | -                    | -                     | •                | -     | -               |

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| v „landscape módu“  |   |   |   |   |   |
| Přizpůsobeno tabletům   | - | - | • | - | - |
| Nastavení aplikace  | - | - | - | - | - |
| Automatické stahování dat na pozadí (novinky, obchody, program kina...) | - | - | - | - | - |
| Uživatelská přívětivost   | - | • | • | • | - |
| Reakce aplikace na požadavky  | - | • | - | • | • |

V tabulce je také zahrnuto osobní hodnocení každé aplikace. Toto hodnocení probíhalo z čistě osobního pocitu a týká se řádků: Uživatelská přívětivost, reakce aplikace na požadavky. Hodnocení bylo následovné, pokud jsem byl spokojen s přívětivostí nebo reakcí aplikace na požadavky, udělil jsem jí bod.

## 7 POŽADAVKY APLIKACE

Tato práce je zaměřena na tvorbu mobilní aplikace určené primárně k podpoře marketingu na dalším informačním kanálu obchodních center. Mobilní aplikace je určena spíše k udržení návštěvnosti, než k jejímu markantnímu zvýšení. Jejím hlavním úkolem je poskytnout potencionálnímu návštěvníkovi aktuální informace o celkovém dění v obchodním centru. V této kapitole se proto pokusíme popsat reálné požadavky na takovou aplikaci.

### 7.1 Specifika požadavků

Velmi důležitým specifikem mobilní aplikace je fakt, že velmi často se nejedná o hlavní, ale spíše o jakýsi doplňkový informační kanál. To se velmi často projevuje neochotou firem platit náklady na vývoj těchto aplikací. Můžeme zcela s určitostí říct, že každé obchodní centrum již dnes vlastní webové stránky, na kterých poskytuje uživateli detailní informace nejen o centru, ale i o akcích, obchodech a službách. Z tohoto důvodu se může jevit jako nejideálnější varianta vytvoření webové aplikace s responzivním designem pro přístup mobilních zařízení. Webová aplikace má ovšem své hranice a nedokáže uživateli nabídnout stejné funkce jako aplikace mobilní (viz. MOBILNÍ APLIKACE A JEJICH VÝVOJ) a také je třeba přistupovat přes prohlížeč vyplněním příslušné adresy. Často se také setkáváme s tím, že vedení firem není dostatečně technicky informováno, často má také nereálné požadavky nebo nesmyslná kritéria hodnocení.

Je třeba se dostatečně zamyslet nad tím, proč a jakým způsobem má být aplikace vyvíjena a vnímána mezi uživateli, aby nedocházelo k jejich nespokojenosti. Mezi hlavní specifika, která se musejí vyřešit před vývojem, je, mimo stanovení typu mobilní aplikace, zejména stanovení jazyka a cílové skupiny, kterým se poté přizpůsobují jednotlivé prvky uvnitř aplikace (tj. vzhled, rozmístění komponent, orientace v aplikaci). Pro obchodní centrum je třeba určit následující specifika

- jazyk aplikace
- potenciální uživatelé
- softwarové požadavky
- hardwarové požadavky
- zdroj dat

### 7.1.1 Jazyk aplikace

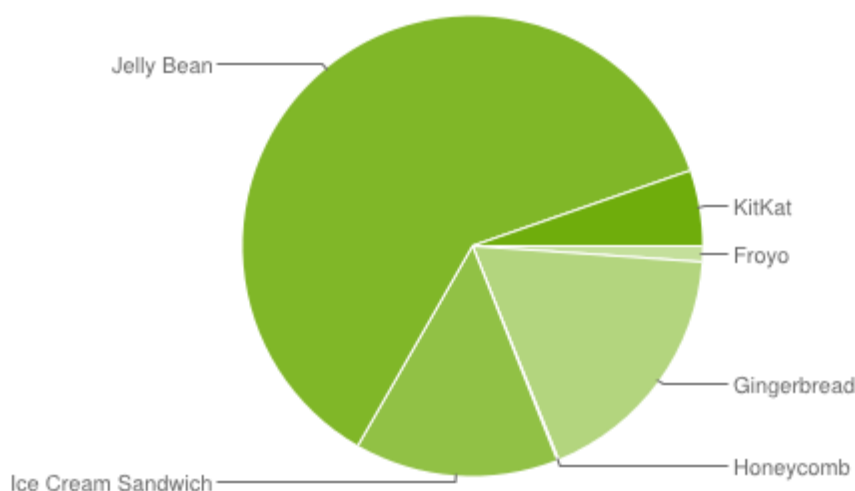
V souladu se zadáním má moje aplikace sloužit klientům obchodních center. Můžeme předpokládat, že její určení je primárně pro Českou Republiku a tak se jako jazyk aplikace jeví čeština. Nicméně není vyloučeno, že si aplikaci může nainstalovat i cizinec, který má v oblibě dané obchodní centrum. Proto jsem zvolil jako defaultní jazyk aplikace angličtinu, a pro lidi, kteří mají v mobilním zařízení nastavený český jazyk, se zobrazí kompletní český překlad.

### 7.1.2 Potenciální uživatelé

Cílová skupina pro mobilní aplikaci obchodních center je choulostivou záležitostí, protože se nedá přesně určit. Podle slov ředitele OZC Zlaté Jablko a.s. p. Ing. Kratochvíla, které mi byly sděleny při osobní schůzce: „*Je pravda, že v obchodních centrech nakupují více ženy, nicméně podle našeho průzkumu je návštěvnost pohlaví všech věkových kategorií prakticky vyvážená*“, jsem usoudil, že nejlepší bude stanovit neutrální cílovou skupinu v rozmezí 12 – 60 let.

### 7.1.3 Softwarové požadavky

Aby mohlo mobilní aplikaci využít co největší množství uživatelů Android, je třeba se zaměřit na verze systému a posoudit jejich jednotlivé zastoupení na trhu, které můžeme vidět na Obr. 26.



Obr. 26. Podíl Android verzí na trhu [12]

Na základě Obr. 26 jsem se rozhodl vyvíjet pro verze Gingerbread (2.3.3) a vyšší. Aplikace tak bude dostupná pro téměř 99% uživatelů, kteří navštíví Google Play. Ve své aplikaci tedy nepočítám s verzí Froyo 2.2, pro které již nejsou podporovány všechny knihovny, které jsou při vývoji využity.

#### **7.1.4 Hardwarové požadavky**

Mobilní aplikace musí být funkční na různých velikostech a hustotách obrazovek.

#### **7.1.5 Zdroj dat**

Jedním z prvních kroků, které musíme při návrhu mobilní aplikace používající internetové připojení udělat, je určit zdroj dat. Obchodní centrum, které má zřízeny internetové stránky má určitě povolenou i jejich správu a přístup k datům, kde zadává aktuální údaje, popř. mění stávající. V rámci této práce je počítáno, že pomocí webových služeb se na tato data lze dotazovat. Jako zdroj dat můžeme tedy považovat poskytnuté webové služby obchodního centra, pomocí kterých si aplikace je schopna aktualizovat a vyměňovat údaje.

### **7.2 Požadovaná funkcionalita**

V následujících podkapitolách jsou popsány reálné požadavky na aplikaci, které byly stanoveny nejmenovaným obchodním centrem v České Republice. Toto obchodní centrum obsahuje také kino. Reálné požadavky obchodního centra se téměř shodují s nalezenými nedostatky již existujících aplikací, které jsou popsány v předchozí kapitole.

#### **7.2.1 Základní popis**

Mobilní aplikace obchodního centra slouží uživatelům k procházení aktuálního harmonogramu akcí, slev a novinek. Také jsou obsaženy základní informace o poloze, historii a kontaktech. Dále si uživatelé mohou prohlédnout jak seznamy obchodů nacházejících se v centru, tak i restaurací a kaváren. Aplikace umožňuje uživatelům také prohlížení aktuálního harmonogramu filmů a novinek z oblasti filmového průmyslu a samotného kina. Mohou si zobrazit týdenní program kina, prohlédnout si informace o daném filmu a případně i provést rezervaci požadovaných sedadel. Provedené rezervace jsou přehledně řazeny v aplikaci. Možnost přehledu provedených rezervací pomáhá uživateli vyhnout se problémům způsobeným zapomenutím na danou akci, jako je třeba nevyzvednutí vstupenek před začátkem filmu.

### 7.2.2 Účel

- Uživatel musí mít možnost prohlédnout si aktuální seznam akcí, novinek a slev v rámci OC včetně týdenního programu kina. Dále jsou k dispozici i informace o OC včetně historie a kontaktů.
- Aplikace musí umět provádění automatických aktualizací dat na pozadí pomocí webových služeb, pokud jsou povoleny uživatelem v nastavení. Tím budou uživateli poskytnuty vždy aktuální údaje o dění v OC včetně aktuálního seznamu obchodů a restaurací.
- V případě zjištění nových akcí, novinek nebo slev v OC, je vyvolána interní notifikace, která upozorní uživatele na vzniklou událost.
- Uživatel musí mít možnost rezervace vstupenek na film, aniž by musel být přítomen v areálu kina.
- Aplikace musí uchovávat informace o dostupných a nedostupných (již vypršených) vstupenkách.
- Je třeba dbát na jednoduchost a uživatelskou přívětivost, proto se musí postupovat podle Android Design Guidelines

### 7.2.3 Rozsah

- Správa vstupenek – aplikace musí umožňovat správu vstupenek, jako je přidání či odebrání rezervované vstupenky ze seznamu vstupenek
- Vyhledání filmů a volných vstupenek – musí být zajištěno vyhledávání volných sedadel a zobrazovat již rezervované sedadla u daného filmu v daný čas
- Rezervace vstupenek – uživateli musí být umožněno zarezervování vstupenek. Dále uživatel u zarezervované vstupenky vidí datum a čas, kdy začíná promítání filmu.

### 7.2.4 Uživatelé

- Uživatel aplikace je zákazník, který má k dispozici zobrazovací zařízení (mobilní telefon, tablet,...) a prostřednictvím mobilní aplikace a Internetu si může zobrazit aktuální data a sdílet podle vlastního uvážení na sociálních sítích atd.
- Uživatel si vybere film z programu filmů na následující týden a zarezervuje si vstupenku na film, pokud jsou ještě dostupná nějaká sedadla.

- Uživatel může vytvořit rezervaci na film bez jakékoliv registrace. Jako identifikační údaj na pokladně pak slouží unikátní číselný symbol označený jako ID vstupenky.

#### **7.2.5 Omezující podmínky**

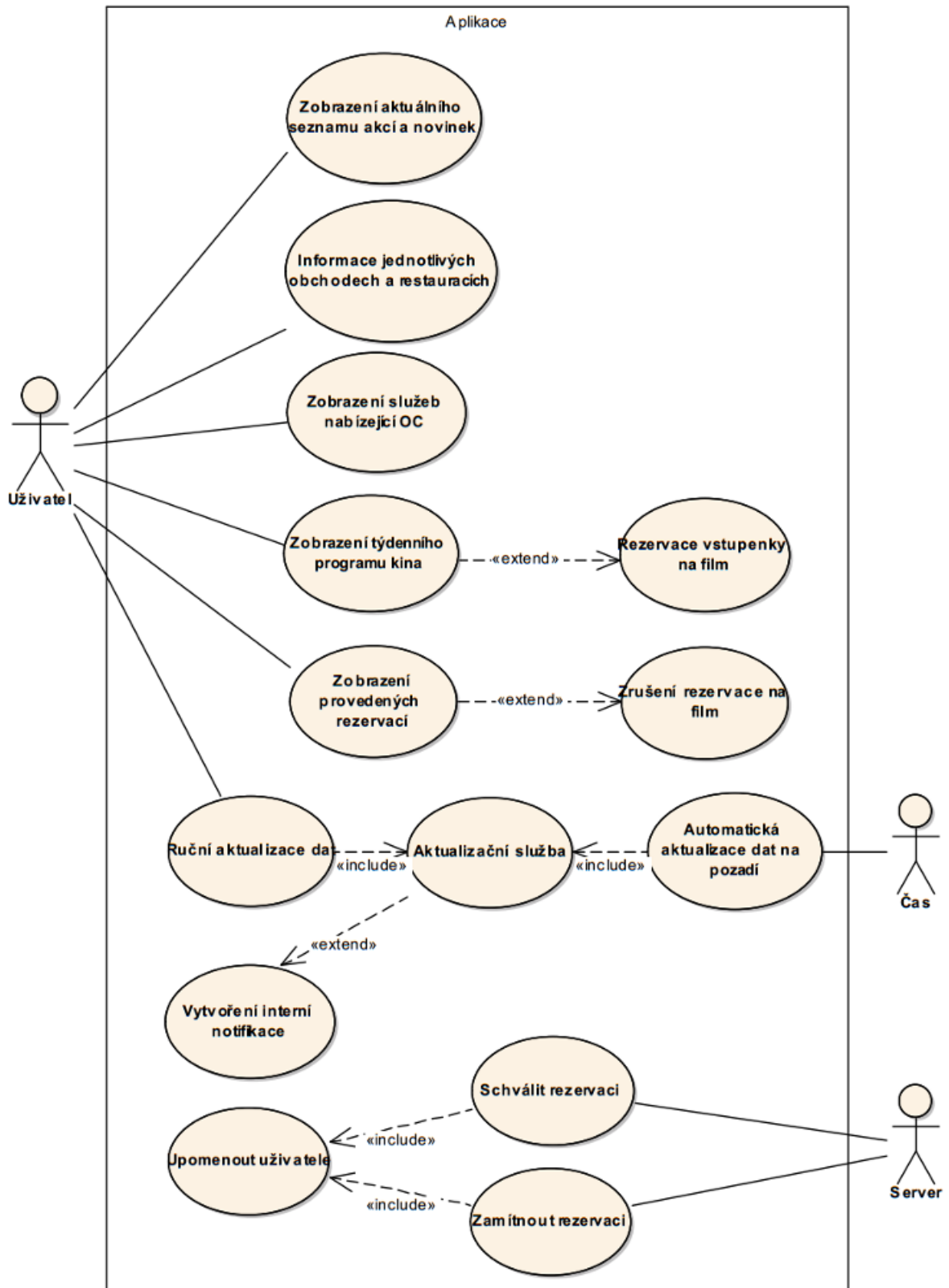
- Omezující podmínky – aplikace vyžaduje připojení k internetu při rezervaci vstupenek a při ručním stahování aktuálních dat. Mobilní aplikace se připojí k serveru obchodního centra vždy, když se změní stav připojení mobilního zařízení (např. když uživatel se připojí do domácí WiFi sítě, tak se automaticky vyvolá kontrola aktuálních dat, je-li povolena v nastavení)
- Verze Android OS – aplikace cílí na co nejvyšší možný počet uživatelů, aplikace musí běžet na Android 2.3.3 a novější
- Aplikace musí být schopna pracovat i v offline režimu a poskytovat všechny údaje stáhnuté při posledním updatu včetně vlastních zarezervovaných vstupenek do kina.

## **8 NÁVRH PROTOTYPU APLIKACE**

Většina funkcí poskytovaných v aplikaci byla již vyložena v předchozí kapitole. Tato kapitola je zaměřena na vytvoření a popis prototypu aplikace.

### **8.1 Use-case diagram**

Formální přehled je popsán v příloženém use-case diagramu na Obr. 27, kde každý use-case reprezentuje možné chování uživatele v aplikaci.



Obr. 27. Use-case diagram mobilní aplikace

Z obrázku můžeme vidět, že v aplikaci figurují tři aktéři. Jeden aktivní a dva pasivní. Prvním a jediným aktivním aktérem je uživatel, který vykonává většinu případů užití.

Většina případů užití se vztahuje k prohlížení dat uložených v aplikaci, ale uživatel může také zarezervovat, či zrušit rezervaci vstupenek do kina nebo provést manuální aktualizaci dat.

Druhým je pak čas, který má na starosti pouze jednu funkci, a tou je aktualizace dat na pozadí. Jedná se o to, že v každý předem definovaný časový interval je vyslán na pozadí pomocí webové služby dotaz na server obchodního centra, který vrátí nějaká data. Pokud došlo ke změně, jsou data roztříděna a uložena do databáze. V tomto případě může dojít i k promazání starších novinek (může být k dispozici max. 20 z důvodu šetření místa na disku – řazeno od nejnovějších). Jedná se také o určité ulehčení, kdy uživatel nemusí mazat jednotlivé položky manuálně. Pokud uživatel není připojen k internetu, odloží se tento krok a aktualizace se vyvolá při následující změně připojení.

Třetím a zároveň posledním aktérem je server kina, který má za úkol schválit nebo zamítnout požadavek při rezervaci vstupenek. Schválí rezervaci, pokud dané sedadlo, či sedadla jsou volná a zapíše číslo lístku do databáze. K zamítnutí rezervace dojde tehdy, kdy během vytváření rezervace si daná místa rezervoval již někdo jiný a jsou tedy obsazena. V obou dvou případech poskytne uživateli zprávu s daným výsledkem.

## 8.2 Uživatelské rozhraní

Při návrhu uživatelského rozhraní neboli GUI, jsem se držel všeobecných zásad, které popisují Android Design Guidelines a také toho, že GUI mobilní aplikace by mělo být založeno na jednoduchosti a přímočarosti. Mobilní aplikace by sama o sobě měla řešit nějaký problém a být přehledná a intuitivní. Skici, neboli mock-ups jsem vytvořil v programu NinjaMock<sup>13</sup>, který je zdarma a lze pracovat online přímo v okně prohlížeče. Na následujících podkapitolách jsou znázorněny všechny důležité obrazovky aplikace.

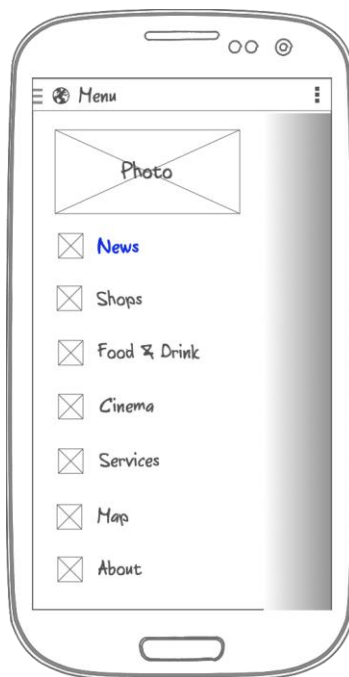
### 8.2.1 Hlavní aktivita, menu a fragment „News“

Hlavní aktivita je tvořena jednou komponentou, která se nazývá „*Navigation Drawer*“. Tato komponenta je složena ze dvou prvků, první je panel (obvykle seznam - *ListView*), který zobrazuje hlavní volby (menu) aplikace na levém okraji obrazovky (viz. Obr. 28). Po většinu času je schován, ale při přejetí prstem (*swipe*) z levého okraje obrazovky nebo

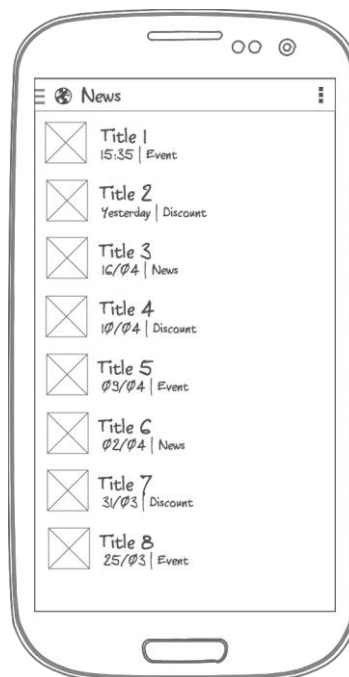
---

<sup>13</sup> Dostupné na: <http://www.ninjamock.com/>

kliknutím na ikonu aplikace ho lze odkrýt. Druhý prvek je reprezentován tzv. Frame Layoutem, do kterého se dynamicky nahrávají data (vždy nějaký fragment) v závislosti na vybrané položce v menu. Tento druhý prvek je vyobrazen na Obr. 29, který zobrazuje stisknutou položku „News“ v menu a obsahuje seznam novinek v obchodním centru. Při otevření aplikace je defaultně zobrazena právě tato položka „News“.



Obr. 28. Hlavní menu aplikace



Obr. 29. Seznam novinek (první položka menu)

### 8.2.2 Fragment ‚Shops‘

Druhou položkou menu je položka reprezentující seznam obchodů. Jedná se o fragment, který obsahuje další dva fragmenty. Každý z nich zobrazuje data podle jiných kritérií, první (A/Z) zobrazuje rovnou sestupně všechny obchody a druhý zase poskytuje kategorie všech obchodů, které jsou v obchodním centru. Kliknutím na příslušný obchod v Obr. 30 dojde k zobrazení příslušných informací o daném obchodu (otevírací doba, sortiment, odkaz na oficiální stránky atd.). Kliknutím na kategorii podle Obr. 31 dojde k přechodu do další obrazovky, kde si uživatel může vybrat příslušný obchod z dané kategorie.



Obr. 30. Sestupné třídění obchodů

Obr. 31. Zobrazení kategorií všech obchodů v centru

Položka v menu „Food & Drink“ (Jídlo a pití) obsahuje naprosto totožné rozmístění prvků, jako položka „Shops“. Z tohoto důvodu není v této kapitole znázorněna.

### 8.2.3 Fragment „Services“

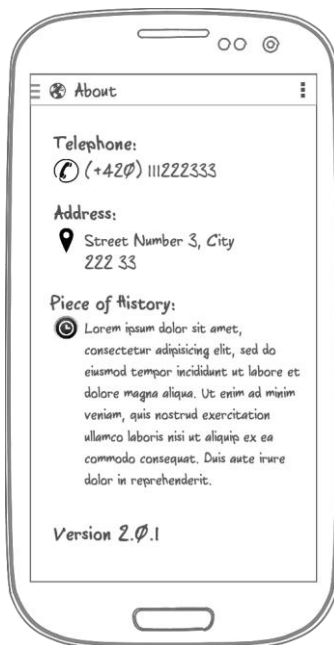
Tato položka menu poskytuje uživateli přehledné informace o dostupných veřejných službách, kterých může využít. Na Obr. 32 si můžeme všimnout tří šipek, pomocí kterých lze data zobrazit, či schovat.



Obr. 32. Fragment ‚Services‘

#### 8.2.4 Fragment ‚Map‘ a ‚About‘

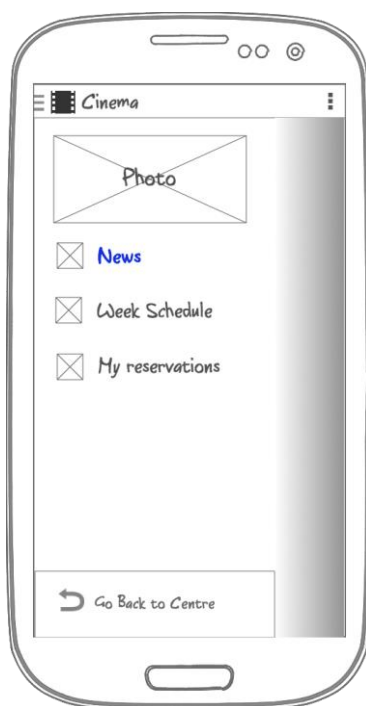
Položka ‚Map‘ je v aplikaci reprezentována obrázkem fiktivního obchodního centra. V případě reálného nasazení aplikace by zde byla umístěna 3D mapa obchodního centra včetně popisu jednotlivých obchodů a restaurací. V položce ‚About‘ můžeme nalézt informace o centru a také kontakt (viz. Obr. 33).



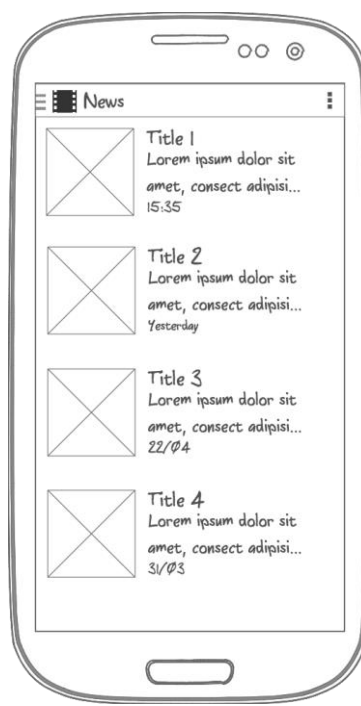
Obr. 33. Fragment ‚About‘

### 8.2.5 Aktivita ‚Cinema‘

Rozhodl jsem se navrhnout rozvrh kina včetně rezervace vstupenek jako takovou pod aplikaci aplikace obchodního centra. Při stisku položky ‚Cinema‘ v menu dojde k otevření nové aktivity, která má lehce pozměněný design a vlastní ‚*Navigation Drawer*‘ (menu). Nicméně jedná se o prakticky stejné rozložení jako pro hlavní aktivitu obchodního centra s tím rozdílem, že pomocí tlačítka ‚Go Back to Centre‘ se uživatel může dostat zpět z této podaplikace do aplikace OC. Na Obr. 34 je opět výsuvné menu, kdy kliknutím na kteroukoliv položku dojde ke změně fragmentu na pravé straně (defaultní fragment při spuštění je ‚News‘). Na Obr. 35 je tedy seznam novinek pro kino. V případě kliknutí na položku se otevře novinka v nové obrazovce (aktivitě).



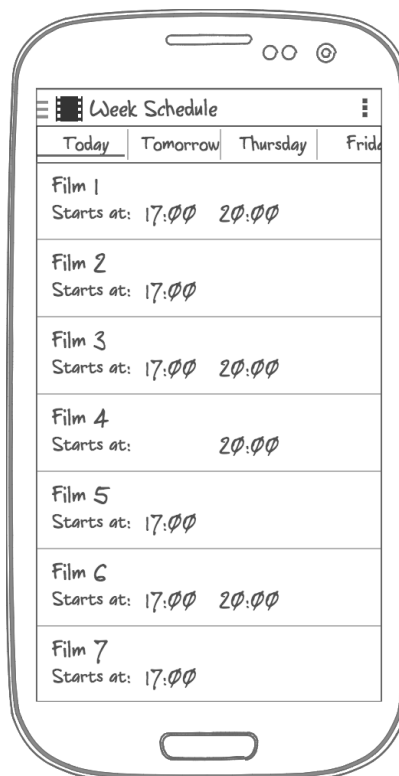
Obr. 34. Menu kina



Obr. 35. Seznam novinek z kina

#### 8.2.5.1 Fragment ‚Week Schedule‘

V tomto fragmentu se nachází týdenní harmonogram kina. Pro každý den je navržena jedna položka tzv. ‚Tab‘, mezi kterými lze přecházet klikem na příslušný tab nebo swipováním.

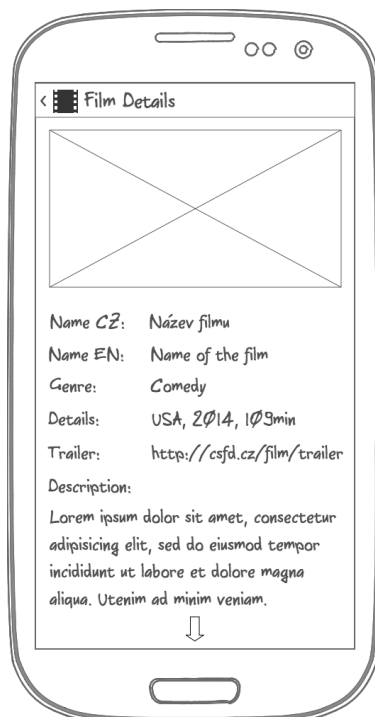


Obr. 36. Týdenní harmonogram filmů

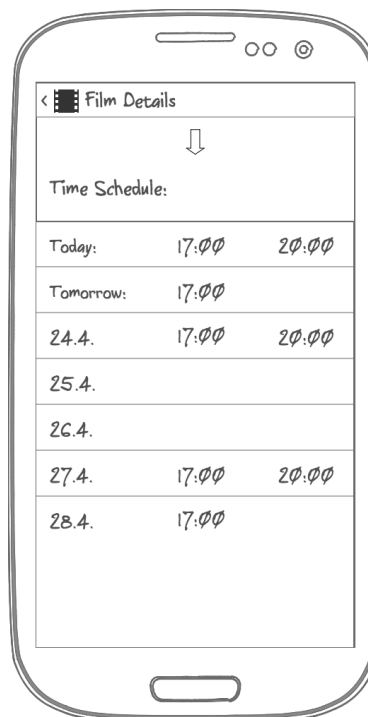
Kliknutím do příslušné položky kteréhokoliv „*ListView*“ dojde k přesměrování na další obrazovku (aktivitu) daného filmu „*Film Details*“ s příslušnými informacemi o daném filmu včetně promítacích časů.

#### 8.2.5.2 Aktivita „*Film Details*“

Tato aktivita obsahuje detail daného filmu včetně obrázku, odkazu na trailer a časů promítání.



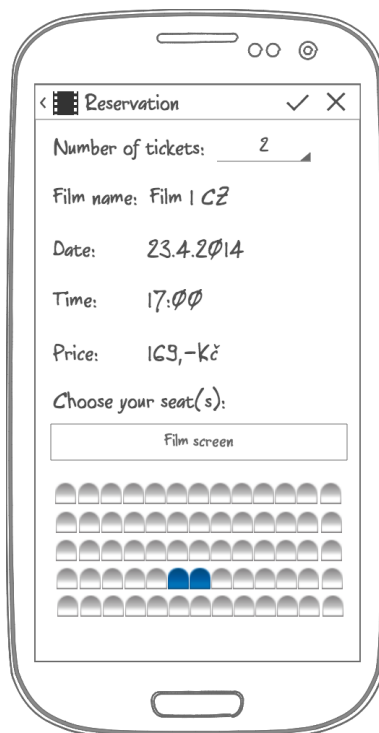
Obr. 37. Detaily filmu 1

Obr. 38. Detaily filmu 2:  
Časy promítání filmu

Kliknutím na čas promítání se uživatel dostane do další pod aktivity, kde si může zarezervovat vstupenky na daný čas. (viz. další podkapitola 8.2.5.3)

### 8.2.5.3 Aktivita ‚Reservation‘

Tato pod aktivita slouží k rezervaci vstupenky (popř. vstupenek), jak můžeme vidět na Obr. 39. Lze zarezervovat až 4 vstupenky.

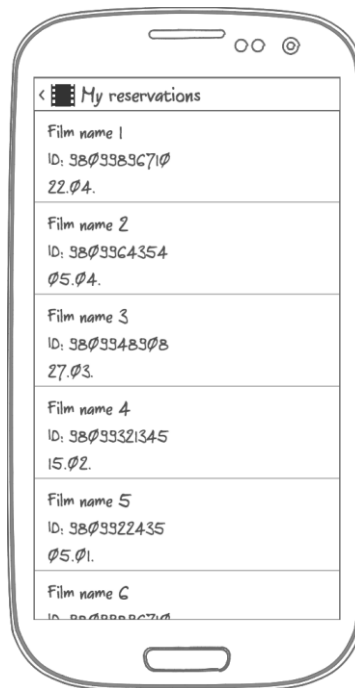


Obr. 39. Rezervace vstupenek

Při odeslání požadavku na server se zkontroluje, zda jsou sedadla ještě volná a podá uživateli zprávu, zda rezervace dopadla úspěšně nebo sedadla jsou již obsazená.

#### 8.2.5.4 *Fragment ,My Reservations‘*

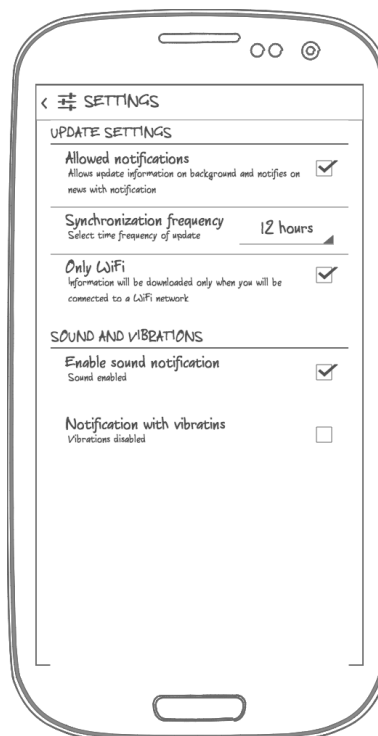
Tento fragment je další položkou v menu kina. Poskytuje seznam uskutečněných rezervací.



Obr. 40. Seznam rezervací

### 8.2.6 Aktivita ‚Settings‘

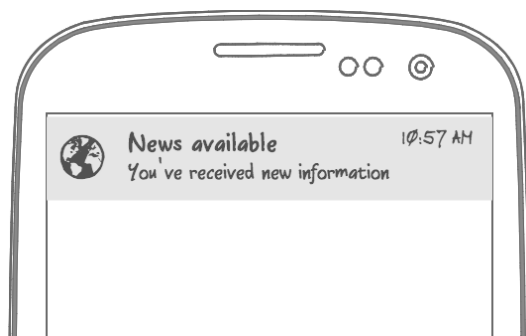
Pod třemi tečkami u většiny screenů se skrývá nastavení, pomocí kterého můžeme povolit automatické stahování dat na pozadí, frekvenci stahování a zvuky a vibrace při upozornění.



Obr. 41. Nastavení

### 8.2.7 Notifikace

Pro případ aktualizace novinek obchodního centra na pozadí jsou přichystané interní notifikace, které se zobrazí uživateli v závislosti na počtu vzniklých novinek podle Obr. 42 nebo Obr. 43.



Obr. 42. Notifikace pro jednu novinku



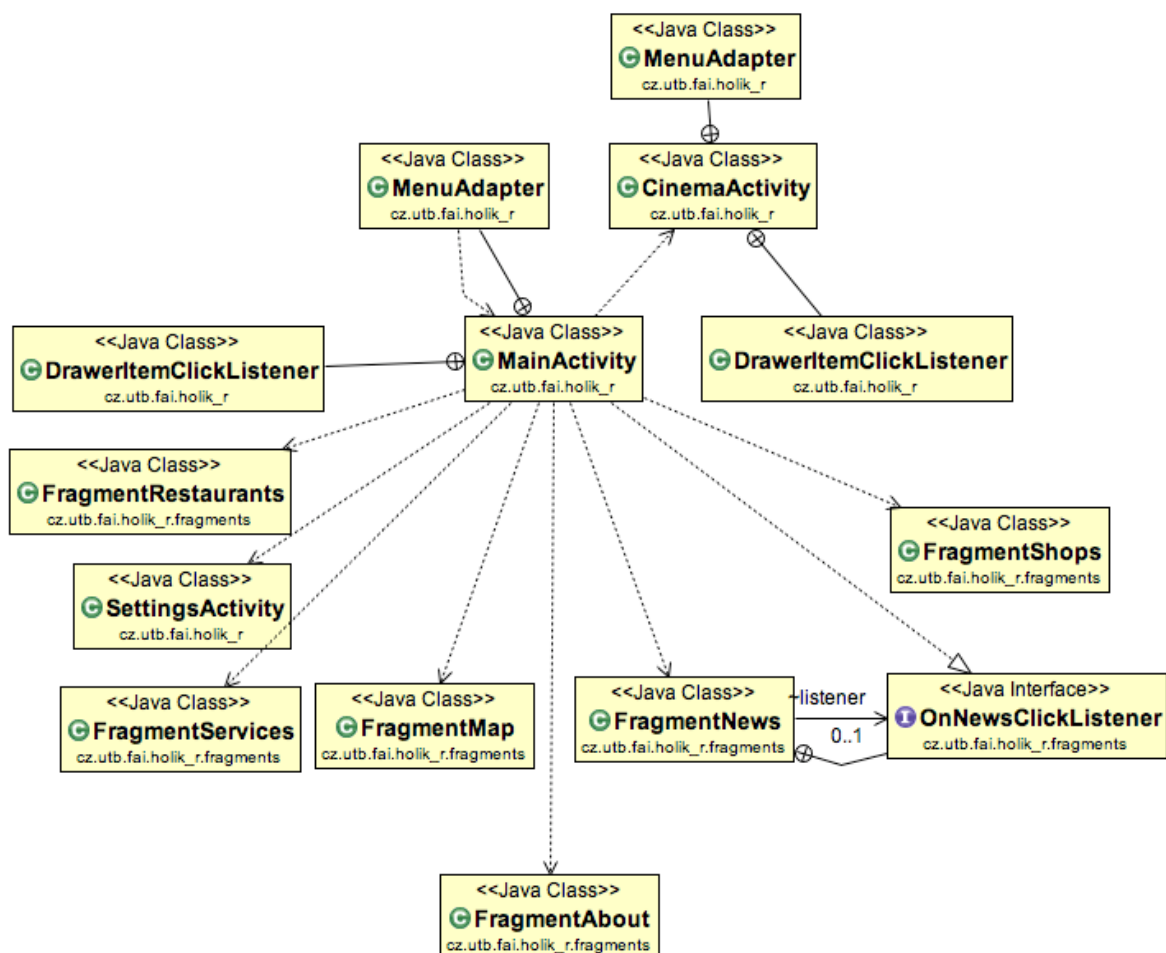
Obr. 43. Notifikace pro více novinek

## 9 NÁVRH ARCHITEKTURY

Rozvržení vnitřní struktury systému je podle dobrých zásad z programování rozvrženo do jednotlivých Java balíčků. Každý balíček reprezentuje významově podobné třídy. Konvencí pro název hlavního balíčku je obrácená webová adresa, která v našem případě vypadá následovně: *cz.utb.fai.holik\_r*. Název tohoto balíčku slouží také jako jednoznačný identifikátor aplikace, kterou chceme nabízet přes oficiální obchod Google Play. Tento balíček obsahuje všechny aktivity používané v aplikaci. Dalšími balíčky jsou: *.adapters*, *.classes*, *.databases*, *.fragments*, *.receivers* a *.services*.

### 9.1 Class diagramy

- **MainActivity class diagram**, který můžeme vidět na Obr. 44, zobrazuje přehled závislosti tříd Android aplikace pro hlavní aktivitu (úvodní obrazovku).

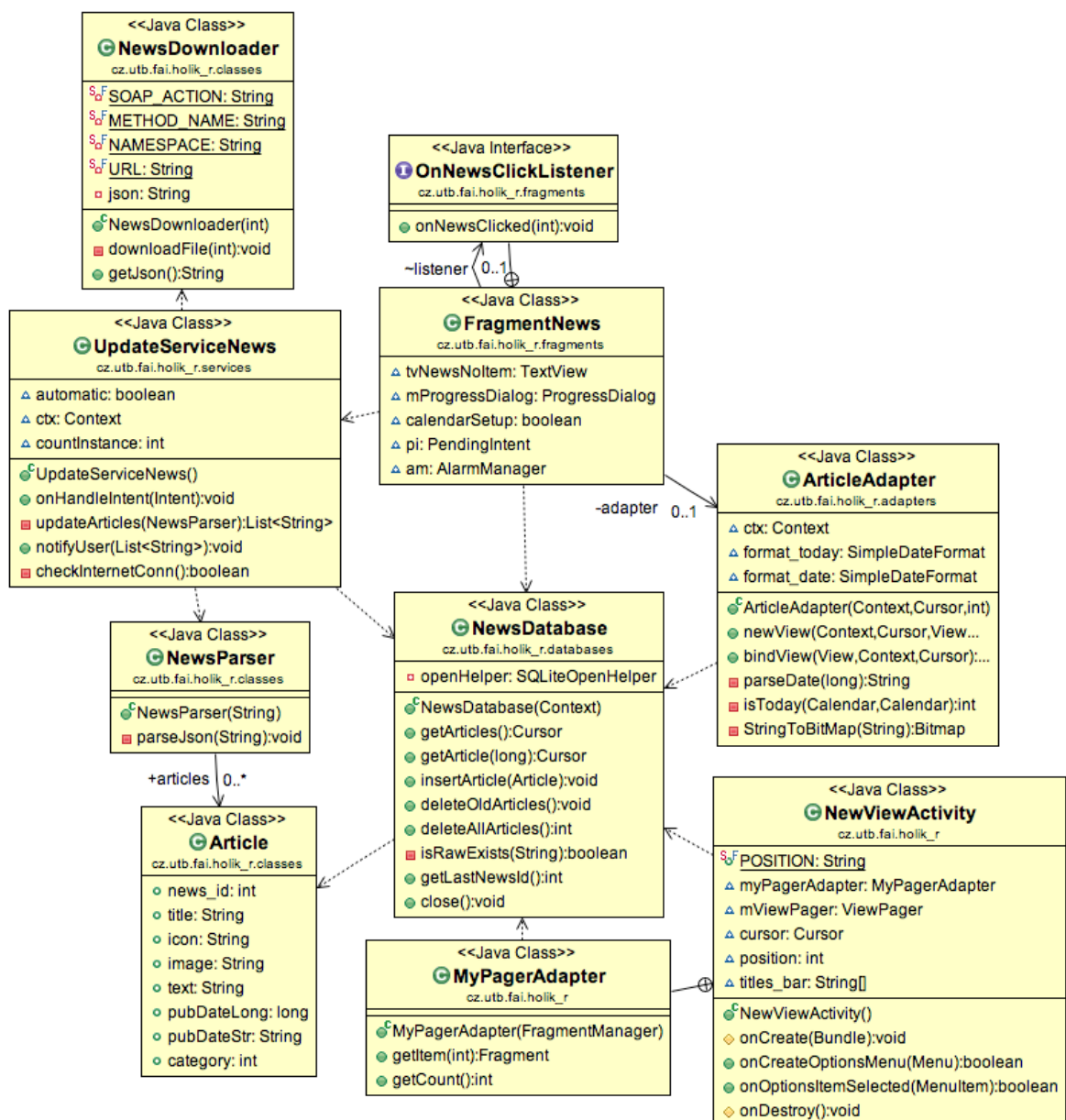


Obr. 44. Diagram hlavních tříd hlavní aktivity aplikace

Záměrně jsou vynechány všechny proměnné a metody kvůli šetření místem. Diagram popisuje závislost mezi hlavní aktivitou a uživatelským rozhraním

jednotlivých tříd na hlavní obrazovce. Třídy FragmentNews, FragmentShops, FragmentRestaurants, CinemaActivity, FragmentServices, FragmentMap a FragmentAbout utvářejí uživatelské rozhraní pro jednotlivé položky menu, které jsou reprezentovány právě těmito Java třídami. Ostatní třídy v diagramu slouží pro správnou funkčnost uživatelského rozhraní, jako např. výsuvné menu. SettingsActivity tvoří samostatnou aktivitu, která slouží pro preferované nastavení aplikace.

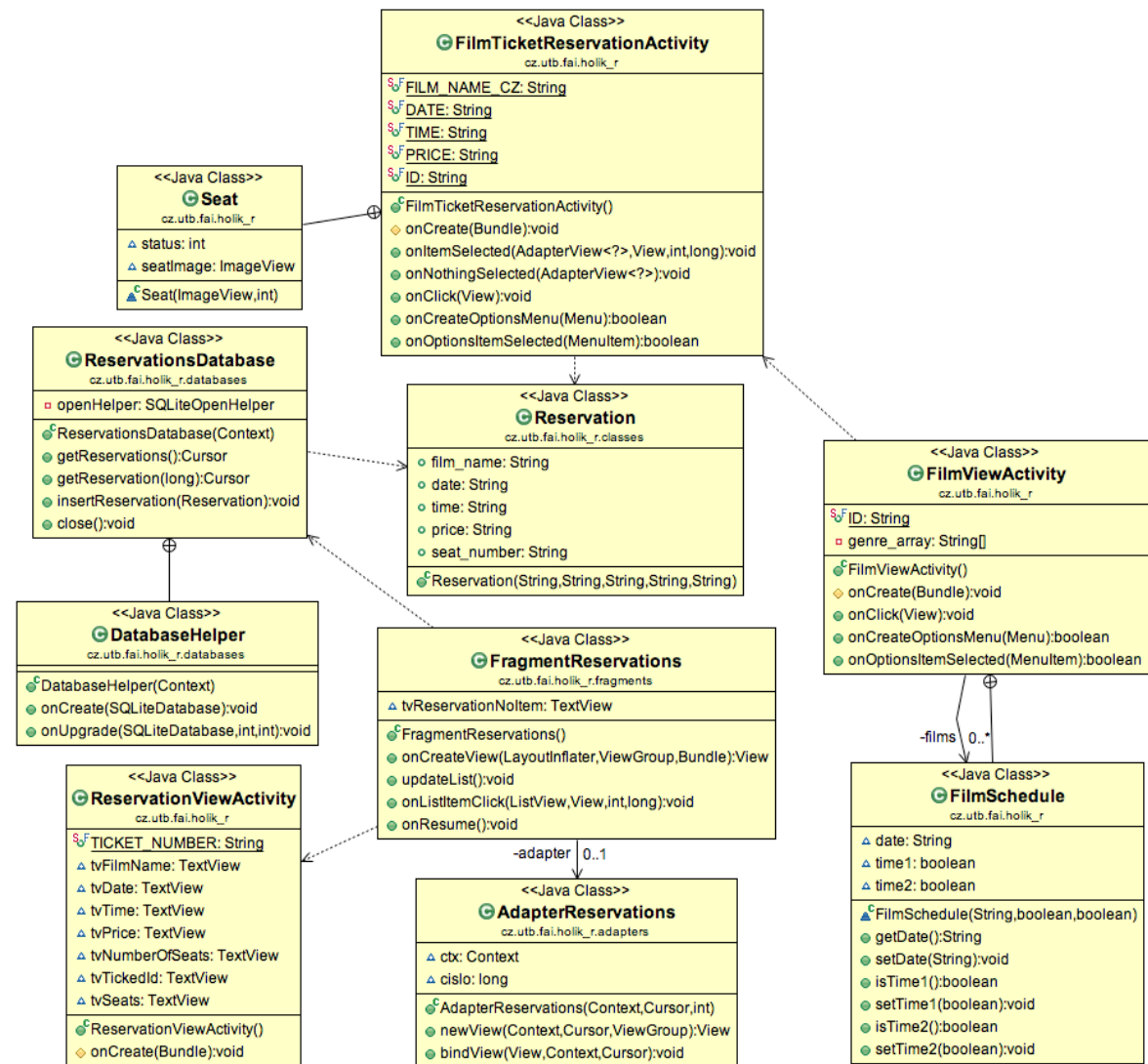
- **NewsFragment class diagram**



Obr. 45. Class diagram třídy NewsFragment a její závislosti na používaných třídách

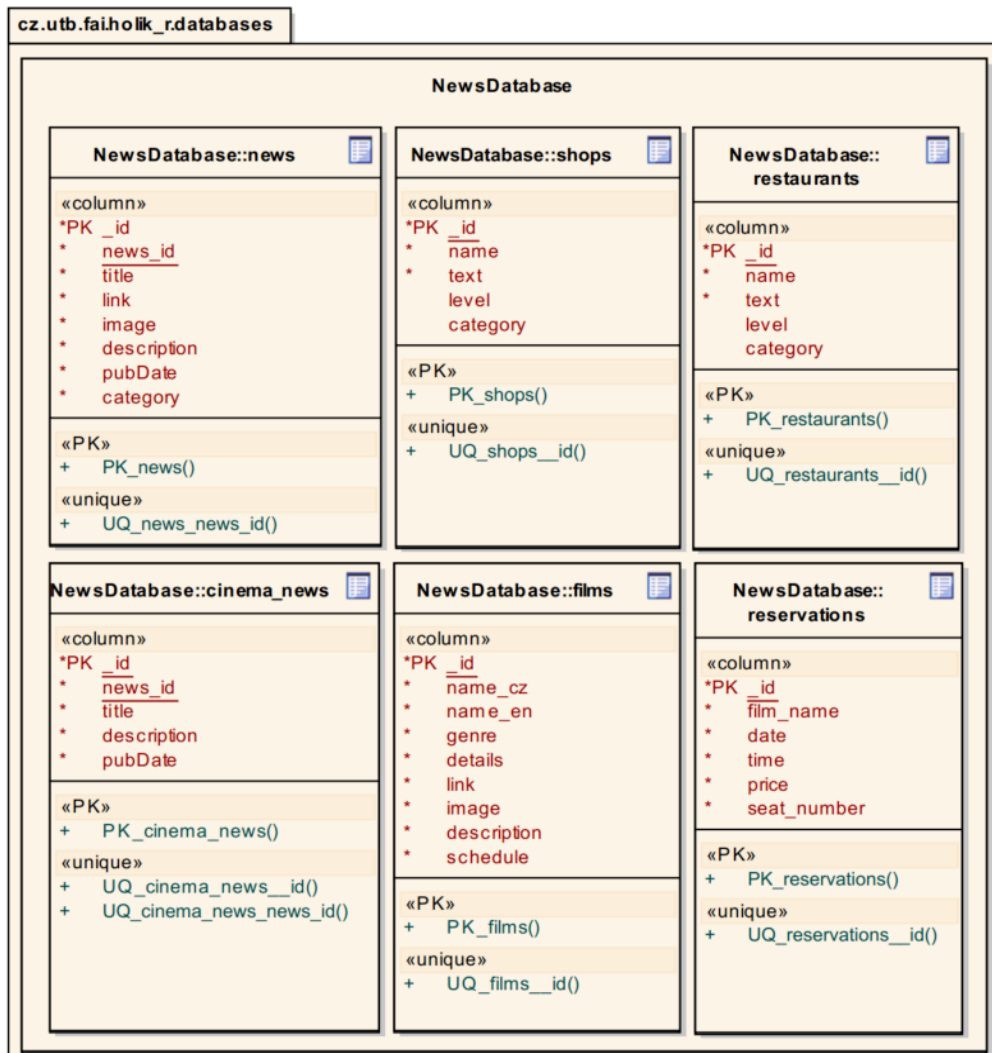
Na Obr. 45 můžeme vidět class diagram, který reprezentuje třídu NewsFragment a její závislosti na dalších třídách, které jsou použity k zobrazení novinek obchodního centra. Opět jsou u některých tříd vynechány statické, veřejné a privátní proměnné a metody.

- **Class diagram Reservation** - rezervace vstupenky do kina



Obr. 46. Class diagram popisující závislost tříd potřebných k rezervaci lístku do kina

- **Databáze** – pro ukládání dat v aplikaci byla vytvořena databáze s celkem šesti tabulkami, viz. Obr. 47.



Obr. 47. Diagram databáze uvnitř Android aplikace

## 10 IMPLEMENTACE

Mobilní aplikace byla vytvořena za pomoci open-source vývojového prostředí Eclipse s integrovaným ADT pluginem. Verze Android OS je 2.3.3 a vyšší, to nám zaručuje bezproblémovou funkčnost přibližně na 99% zařízeních používající tento OS. Dále byla použita externí knihovna kSOAP 2 verze 2.6.4 pro komunikaci s webovou službou, která je dostupná na adrese: <http://oc.asp2.cz/NewsService.asmx>. Tato webová služba obsahuje vygenerované testovací data, pomocí kterých lze nasimulovat podmínky reálného obchodního centra. Webová služba bude dostupná do konce kalendářního roku 2014.

Po shromáždění dat z předchozích kapitol jsem získal potřebné informace k implementaci požadovaných funkcí do mobilní aplikace. V následujících kapitolách jsou rozebrány některé podstatné body implementace

### 10.1 Získávání dat

Téměř všechny informace jsou získávány pomocí již zmíněné webové služby. Veškerá komunikace s Internetem musí na platformě Android probíhat mimo hlavní vlákno aplikace, protože ve hlavním vlákne běží celé UI a kdyby nějaká akce trvala více, než několik vteřin, systém Android by aplikaci násilně ukončil. V naší aplikaci jsou http požadavky zpracovávány přes vedlejší vlákno pomocí návrhového vzoru „*Messenger*“. Dotazy na webovou službu jsou zprostředkovány pomocí knihovny kSOAP. Ukázkové použití knihovny kSOAP můžeme vidět na ukázce Kód 1.

```
public class NewsDownloader {  
  
    private static final String SOAP_ACTION =  
    "http://oc.asp2.cz/News/GetNews";  
    private static final String METHOD_NAME = "GetNews";  
    private static final String NAMESPACE =  
    "http://oc.asp2.cz/News";  
    private static final String URL =  
    "http://oc.asp2.cz/NewsService.asmx";  
  
    private String json = null;  
  
    public NewsDownloader(int lastNewsId) {  
        downloadFile(lastNewsId);  
    }  
  
    private void downloadFile(int lastNewsId) {  
        SoapObject Request = new SoapObject(NAMESPACE, METHOD_NAME);  
        Request.addProperty("id", lastNewsId);  
        SoapSerializationEnvelope soapEnvelope = new
```

```
SoapSerializationEnvelope (SoapEnvelope.VER11);
    soapEnvelope.dotNet = true;
    soapEnvelope.setOutputSoapObject (Request);

    AndroidHttpTransport aht = new AndroidHttpTransport (URL);

    aht.debug = true;
    aht.call (SOAP_ACTION, soapEnvelope);
    SoapPrimitive resultString =
(SoapPrimitive) soapEnvelope.getResponse ();
    json = resultString.toString ();
}
}
```

Kód 1. Ukázka knihovny kSOAP

Vracená data ze serveru jsou ve tvaru JSON. Ukázkový JSON parser pro novinky obchodního centra je na ukázce Kód 2.

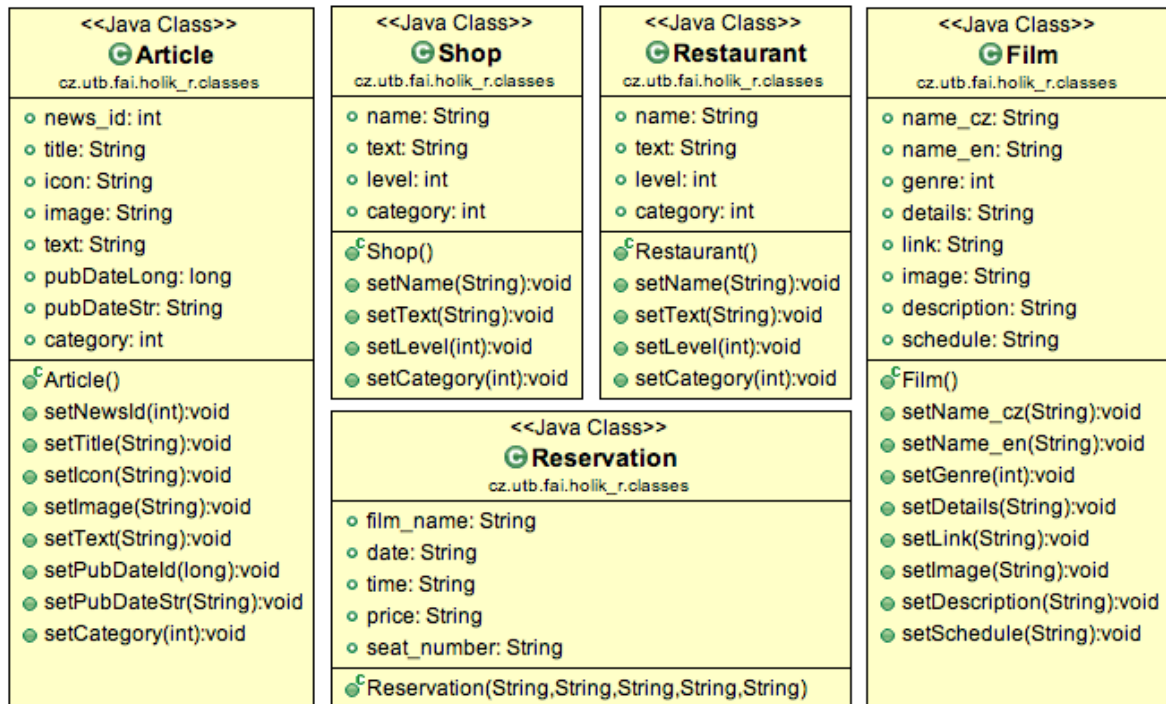
```
private void parseJson (String jsonStr) {
    // vytvoření instance JSONObject
    JSONObject jsonObj = new JSONObject (jsonStr);
    // vytvoření instance JSONArray pro řetězec s konstantou NEWS
    JSONArray jArr = jsonObj.getJSONArray (NEWS);
    // cyklus, ve kterém se prochází jednotlivé prvky pole
    for (int i=0; i< jArr.length (); i++) {

        JSONObject obj = jArr.getJSONObject (i);
        // získání hodnot z JSON pole pro každou iteraci
        int news_id = obj.getInt (ID);
        String title = obj.getString (TITLE);
        String icon = obj.getString (ICON);
        String image = obj.getString (IMAGE);
        String text = obj.getString (TEXT);
        // převedení času ze stringu do formátu long
        String pubdateStr = obj.getString (PUBDATE);
        int category = obj.getInt (CATEGORY);

        .
        .
        .
        // uložení hodnot do instance entity Article
    }
}
```

Kód 2. Ukázka parsování JSON na platformě Android

Pro práci s daty jsem vytvořil tzv. entity class – tedy třídy reprezentující objekty, se kterými je pracováno. Jejich obsahem je pouze konstruktor, atributy a get, set metody. Pro tuto aplikaci využívám celkem pět těchto tříd. Jedná se přesně o třídy viz. Obr. 48.



Obr. 48. Jednotlivé entity třídy používané v aplikaci

## 10.2 Ukládání dat

Údaje se v Androidu dají ukládat dvěma různými způsoby. Tou nejběžnější je uložení hodnot do relační databáze. Druhý způsob je uložení údajů do tzv. *Shared Preferences*, které fungují na principu *Key-Value* a data jsou uložena v databázi dlouhodobě.

### 10.2.1 Databáze v Androidu

Data jsou stažena přes webovou službu a parsována (viz. předchozí kapitola 10.1) do tzv. objektů entity tříd, které slouží jako vstupní argumenty metody `insert()` v databázi aplikace. Systém Android obsahuje SQLite databázi, která slouží k ukládání dlouhodobých údajů. Jejím zástupcem je třída `SQLiteOpenHelper`, kterou stačí zdědit a překrýt metodu `onCreate()`, která se zavolá při prvním používání aplikace a vytvoří tabulku. Dále je zde metoda `onUpgrade()`, která se volá v případě změny verze databáze na vyšší (tedy při její aktualizaci). Příklad definování databáze můžeme vidět na Kódu 3.

```
static class DatabaseHelper extends SQLiteOpenHelper {
    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TB_NAME + " ("
```

```
+ COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
+ COLUMN_NEWS_ID + " INTEGER,"
+ COLUMN_TITLE + " TEXT NOT NULL,"
+ COLUMN_ICON + " TEXT NOT NULL,"
+ COLUMN_IMAGE + " TEXT NOT NULL,"
+ COLUMN_TEXT + " TEXT NOT NULL,"
+ COLUMN_PUBDATE + " TEXT NOT NULL,"
+ COLUMN_CATEGORY + " INTEGER"
+ ");");

    Log.i("Database", "DB news created!");
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TB_NAME);
    onCreate(db);
}
}
```

Kód 3. Ukázka třídy SQLiteOpenHelper, vytváření a aktualizace tabulek v databázi na Android OS

Pro použití v aplikaci je nutné na instanci třídy *DatabaseHelper* zavolat metodu *getReadableDatabase()* pro čtení nebo *setReadableDatabase()* pro zápis údajů. Po dokončení práce s databází je třeba zavolat metodu *close()*. Veškeré třídy reprezentující databáze jsem umístil do balíčku *.databases*. SQL příkazy je možno psát dvěma různými způsoby. První způsob je použití metody *rawQuery()*, kde se zapisuje příkaz v surové podobě. Druhé a elegantnější řešení je použití metody, do které se pouze dosadí jednotlivé části příkazu jako výčet požadovaných sloupců, podmínky, řazení a jiné. Samotný příkaz se pak generuje automaticky. Nesmírnou výhodou je, že takto se vyhneme chybám, které jsou často způsobeny při surovém zápisu metodou *rawQuery()*. Pro vkládání hodnot do databáze se používá instance třídy *ContentValues*, do které se přidávají údaje typu sloupec-hodnota. Instance této třídy je pak předána jako součást metody *insert*.

```
public void insertArticle(Article article){

    SQLiteDatabase db = openHelper.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(COLUMN_NEWS_ID, article.news_id);
    values.put(COLUMN_TITLE, article.title);
    values.put(COLUMN_ICON, article.icon);
    values.put(COLUMN_IMAGE, article.image);
    values.put(COLUMN_TEXT, article.text);
    values.put(COLUMN_PUBDATE, article.pubDateLong + "");
    values.put(COLUMN_CATEGORY, article.category);
}
```

```
db.insert(TB_NAME, null, values);

db.close();
}
```

Kód 4. Ukázka metody insert v SQLite databázi

Pro čtení dat je potřeba využít instanci *Cursor*, která reprezentuje výsledky daného dotazu. (viz. Kód 5) S touto instancí pak můžeme dále pracovat a získávat, či filtrovat data, jak potřebujeme.

```
public Cursor getArticle(long id){
    SQLiteDatabase db = openHelper.getReadableDatabase();
    String[] selectionArgs = {String.valueOf(id)};
    String[] columns = {COLUMN_ID, COLUMN_TITLE, COLUMN_IMAGE,
    COLUMN_TEXT, COLUMN_CATEGORY};
    return db.query(TB_NAME, columns, COLUMN_ID + "= ?",
    selectionArgs, null, null, null);
}
```

Kód 5. Ukázka query v SQLite databázi

## 10.2.2 Shared Preferences

Jak již bylo zmíněno výše, *Shared Preferences* jsou druhou variantou ukládání dlouhodobých dat. Využívá se zejména pro uchování nastavení aplikace. Poskytuje tzv. posluchače událostí, které reagují na změnu uvnitř *Shared Preferences* a tím může aplikace měnit a přizpůsobovat své chování za běhu. Já jsem je použil ve třídě *SettingsActivity*, která slouží právě k uchovávání nastavení pro celou aplikaci. Ukázku vložení dat do *Shared Preferences* může vypadat následovně, viz. Kód 6.

```
public static void setAllowedNotification(Context ctx, boolean
value){
    SharedPreferences sharedPref =
    PreferenceManager.getDefaultSharedPreferences(ctx);
    Editor editor = sharedPref.edit();
    editor.putBoolean(ALLOW_NOTIFICATION, value);
    editor.commit();
}
```

Kód 6. Ukázka vkládání dat do Shared Preferences

Před uložením údajů musíme vytvořit novou instanci třídy *SharedPreferences*, a protože se jedná o návrhový vzor *Singleton*, je třeba získat odkaz na instanci pomocí statické metody *.getDefaultSharedPreferences()*. Poté vytvoříme instanci třídy *Editor()* pomocí instance na *SharedPreferences()* a metody *edit()*. Nyní můžeme uložit hodnoty, které se

opět ukládají způsobem key-value. Nesmíme zapomenout potvrdit tuto změnu metodou `commit()`.

Pro získání dat ze *SharedPreferences* je potřeba zavolat jednu ze statických metod třídy *PreferenceManager.getDefaultSharedPreferences(context)* viz. kód 7. Abychom získali danou hodnotu, musíme použít klíč, podle kterého je hodnota identifikována. Pokud tento klíč neexistuje, je vrácena hodnota ve druhém atributu metody např. *getBoolean(key, value)*, tak, jak můžeme vidět v kódu 7.

```
public static boolean getAllowNotification(Context ctx) {
    return
    PreferenceManager.getDefaultSharedPreferences(ctx).getBoolean(ALLOW_NOTIFICATION, true);
}
```

Kód 7. Ukázka získání dat ze Shared Preferences

### 10.3 Automatické stahování dat

Platforma Android umožňuje stahování dat na pozadí, aniž by to uživatel postřehl. S touto možností je třeba zacházet opatrně, protože při příliš častém intervalu a velké náročnosti zpracování dat může mít velký vliv na úbytek energie v baterii. V aplikaci používám automatické stahování dat na pozadí pro stahování novinek obchodního centra a aktualizace jeho obchodů, restaurací a programu kina. Frekvenci stahování lze nastavit v nastavení aplikace.

#### 10.3.1 Alarm manager a pending intent

Opakující události v Androidu reprezentuje třída *AlarmManager*, pomocí které můžeme spouštět nějakou akci pravidelně třeba každý den v určitou hodinu. Její metoda `.setRepeating(int type, long triggerAtMillis, long intervalMillis, PendingIntent operation)` přebírá následující argumenty:

- `type` – jedná se o 4 konstanty: `ELAPSED_REALTIME`, `ELAPSED_REALTIME_WAKEUP`, `RTC` a `RTC_WAKEUP`. Rozdíl mezi `ELAPSED_REALTIME` a `RTC` je v tom, že u prvního jmenovaného se čas předaný v „*triggerAtMillis*“ počítá u naboování zařízení, zatímco `RTC` přijímá Unixovou časovou známku – počet milisekund od půlnoci 1.1.1970. Rozdíl mezi `WAKEUP` a bez `WAKEUP` je ten, že s `WAKEUP` dojde k akci vždy v přesně předanou dobu a pokud je potřeba, zařízení je probuzeno ze spánku. Zatímco druhá

varianta počká, až se mobilní zařízení vzbudí ze spánku a poté provede příslušnou akci.

- `triggerAtMillis` – časová známka známku, kdy se akce poprvé spustí.
- `intervalMillis` – časové rozmezí pro jednotlivé intervaly opakování akce (v milisekundách).
- `operation` – reprezentuje samotnou akci. Třída `PendingIntent` je podobná klasickému `Intentu`, která navíc nese informaci o tom, jakou metodou se má spustit (`startActivity()`, `startService()` nebo `sendBroadcast()`). Právě na `PendingIntent` závisí, která akce se spustí při sepnutí `AlarmManageru`.

Možná implementace `AlarmManageru` je uvedena na ukázce kódu 8.

```
Intent intent = new Intent(this, UpdateServiceNews.class);
intent.putExtra(UpdateServiceNews.AUTOMATIC_SERVICE, true);

// inicializace Pending Intentu
pi = PendingIntent.getService(this, 0, intent, 0);

// inicializace alarm manageru
am = (AlarmManager) this.getSystemService(Context.ALARM_SERVICE);
// nastavení periody alarm manageru, AlarmManager.RTC = pokud je
telefon v úsporném režimu, necháme ho spát a aktualizaci
provedeme až když se vzbudí, jinak se AlarmManager zapne každých
60 min
am.setRepeating(AlarmManager.RTC, calendar.getTimeInMillis(),
1000*60*60, pi);
```

Kód 8. Ukázka implementace `AlarmManageru`

Pokud chceme daný `AlarmManager` zrušit, stačí na něm zavolat metodu `cancel()` se stejným `PendingIntent` v argumentu, se kterým byl vytvořen. Pokud se tak nestane, `AlarmManager` se bude neustále opakovat. Ve výsledné aplikaci této práce je `AlarmManager` v pravidelných intervalech (definovaných v nastavení) používán ke spouštění další Java třídy, která kontroluje aktuální data na serveru.

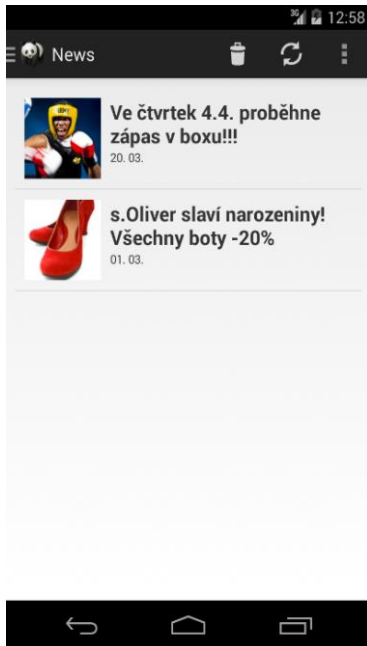
## 11 VÝSLEDNÁ ANDROID APLIKACE

Ze srovnání již existujících aplikací v kapitole 6. jsme zjistili řadu nedostatků, které naše mobilní aplikace dokáže zvládnout lépe, dokonce byly implementovány některé funkce, které žádná ze srovnávaných aplikací neobsahuje. Podařilo se implementovat všechny stanovené požadavky z kapitoly 7. Aplikace pro svůj plnohodnotný běh potřebuje alespoň občasné připojení k internetu. Mezi jednou z hlavních funkcí je schopnost automaticky kontrolovat a stahovat vniklé novinky a akce v obchodním centru na pozadí aplikace, aniž by o tom uživatel přímo věděl a případně uživatele upozornit na stažení aktuálních dat.

Na následující podkapitole jsou prezentovány jednotlivé hlavní obrazovky aplikace, které jsou rozvrženy podle návrhu v kapitole 8.

### 11.1 Ukázka aplikace

Při prvním spuštění mobilní aplikace dojde automaticky ke stažení aktuálních dat reprezentující novinky, pokud je zařízení připojeno k internetu. Uživatel může zvolit i ruční aktualizaci v horní části panelu akcí kliknutím na příslušnou ikonu, kdykoliv uzná za vhodné. Prvotní obrazovka, která je zobrazena po spuštění aplikace jsou novinky, viz. Obr. 49. Pokud žádné novinky ještě nejsou uloženy v databázi, zobrazí se textová zpráva, že databáze neobsahuje žádná data a uživatel by měl aktualizovat.



Obr. 49. Přehled novinek



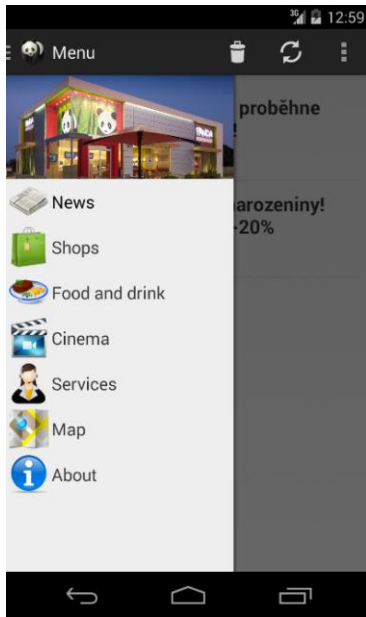
Obr. 50. Detail novinky



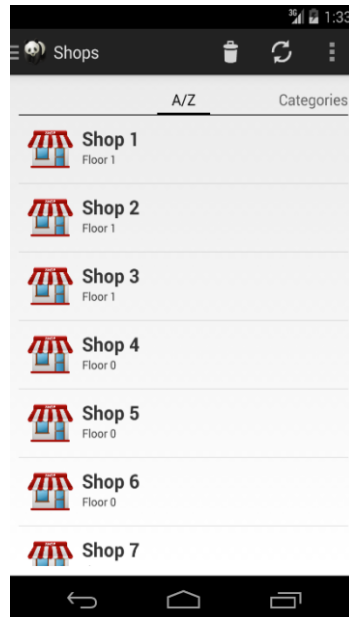
Obr. 51. Swipe gesto mezi novinkami

Obr. 50 obsahuje detail pro danou novinku a Obr. 51 symbolizuje tzv. gesto swipe mezi jednotlivými novinkami. Není tedy nutné v případě více novinek vracet se zpět do seznamu novinek, ale lze rovnou „swipovat“ mezi nimi.

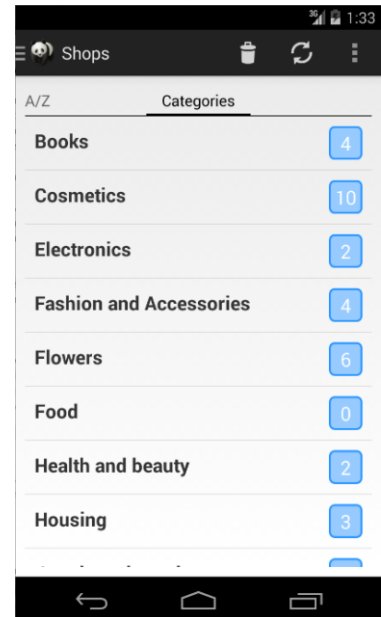
Následující Obr. 52 zobrazuje hlavní menu aplikace, které je tzv. výsuvné. Po většinu času ho není vidět, ale při přejetí prstem zleva doprava nebo kliknutím na ikonu pandy dojde k jeho zobrazení. Při kliknutí na kteroukoliv položku v menu, menu zmizí a v obsahové části se zobrazí požadovaná data.



Obr. 52. Hlavní menu aplikace



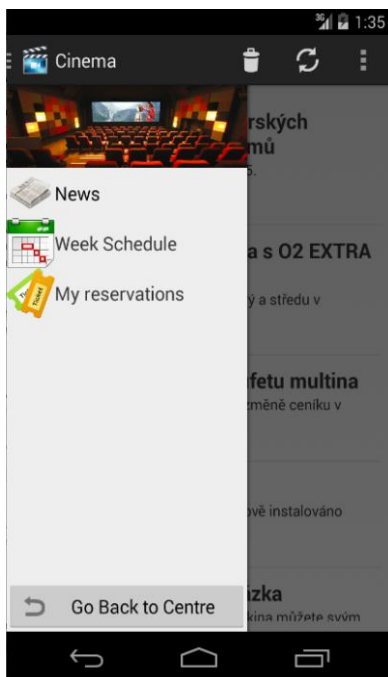
Obr. 53. Přehled obchodů podle abecedy



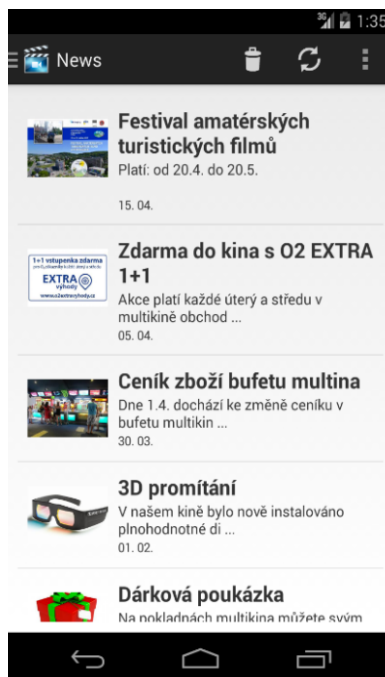
Obr. 54. Přehled kategorií obchodů

Obr. 53 a Obr. 54 zobrazují v prvním případě seznam všech obchodů podle abecedy, druhý zmiňovaný obrázek zase představuje přehled jednotlivých kategorií a počet obchodů, které do dané kategorie patří. Mezi oběma obrazovkami je možno opět přejíždět pomocí „*swipe*“ gesta.

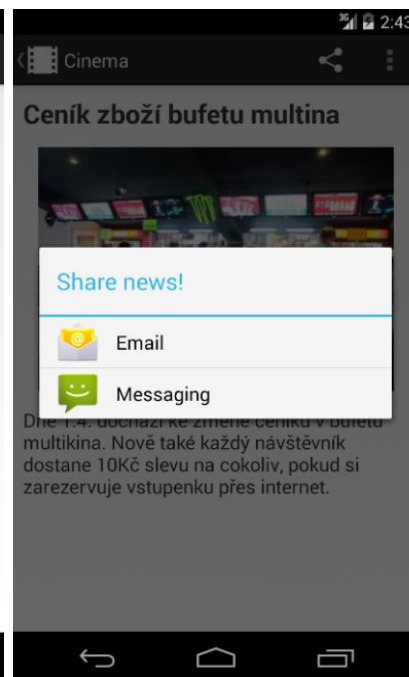
Další důležitou položkou výsuvného menu je kino, které se otevře ve vlastní obrazovce a obsahuje lehce pozměněný design a dokonce i vlastní menu, viz. Obr. 55.



Obr. 55. Menu kina



Obr. 56. Přehled novinek kina



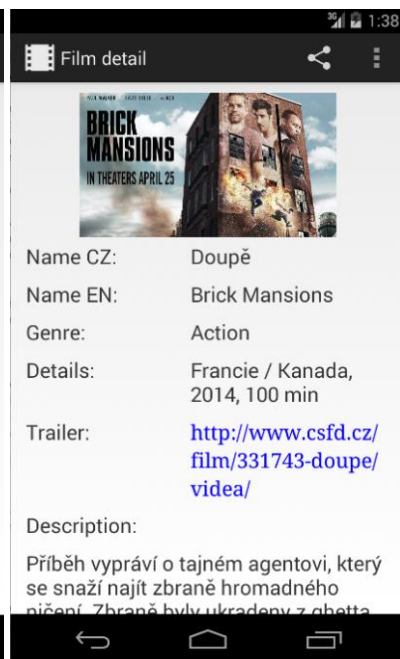
Obr. 57. Možnosti sdílení obsahu

Přehled novinek kina je navržen ve stejném rozvržení, jako přehled novinek obchodního centra, viz. Obr. 56. Stejně tak i detail jednotlivých novinek je konstruován podle stejné šablony. Další funkce, která je implementována pro textový obsah u detailu novinek obchodního centra, kina i u detailu jednotlivých filmů, je možnost sdílení dat pomocí dalších aplikací, které jsou schopny přijmout textový řetězec. Tyto aplikace jsou na každém zařízení odlišné a zobrazí se jako seznam, stejně jako na Obr. 57, podle toho, které aplikace má uživatel nainstalovány. Tento obsah je možno sdílet i přes aplikace sociálních sítí, jako je Facebook, Twitter atd.

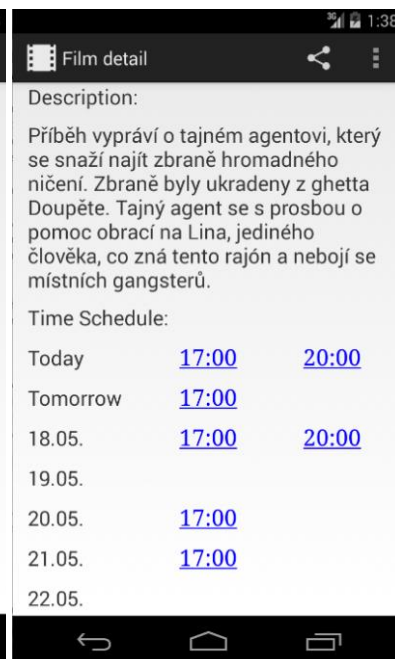
Druhou položkou výsuvného menu kina je aktuální program kina na následující týden, viz. Obr. 58. Mezi jednotlivými dny se dá opět „swipovat“.. Při kliknutí na kterýkoliv film dojde k otevření nové obrazovky s detailem filmu, jak je ukázáno na následujícím Obr. 59 a Obr. 60. Jako detaily pro daný film se zobrazuje obrázek, název filmu v českém i anglickém jazyku, žánr, příslušné detaily filmu, odkaz na trailer a následný popis.



Obr. 58. Týdenní program kina

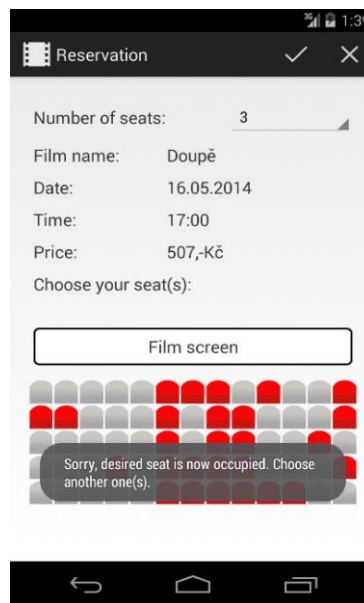
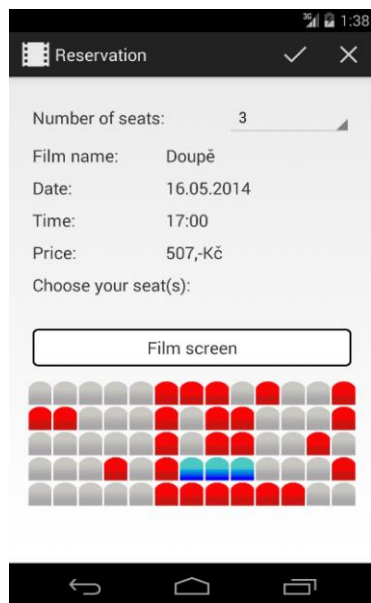


Obr. 59. Film detail 1



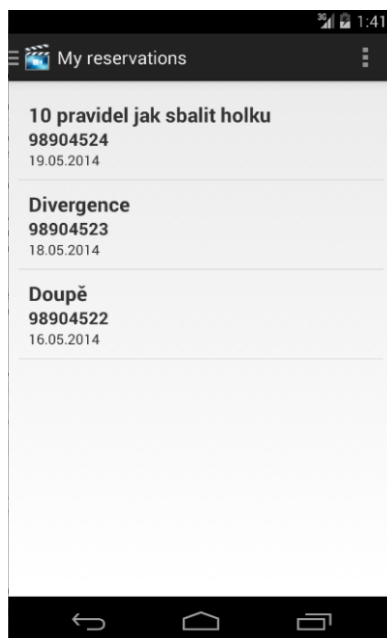
Obr. 60. Film detail 2

Na již zmiňovaném Obr. 60 je zobrazen aktuální harmonogram promítání daného filmu. Po kliknutí na příslušný čas v daném řádku reprezentující datum, dojde k přesměrování na další obrazovku, na které je možno rezervovat vstupenky na tento film pro daný datum a čas. Obrazovka pro rezervaci filmu je na Obr. 61. Aplikace je taky ošetřena pro případ, že by si v daný okamžik zarezervoval lístek/lístky někdo těsně před námi. V tom případě aplikace upozorní uživatele na vzniklou situaci a požádá ho o další pokus viz. Obr. 62.



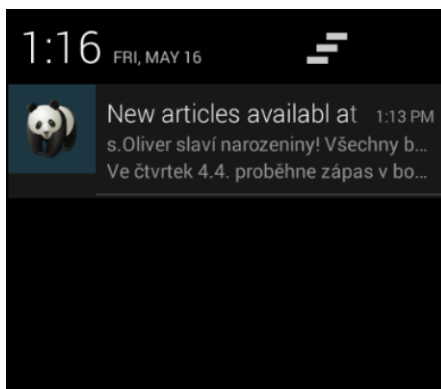
Obr. 61. Obrazovka rezervace Obr. 62. Zamítnutí rezervace

Pro přehled uskutečněných rezervací slouží položka v menu kina s označením „*Moje rezervace*“, viz. Obr. 63. Detail dané rezervace se otevře po kliknutí na položku v seznamu rezervací a je vyobrazena na Obr. 64.



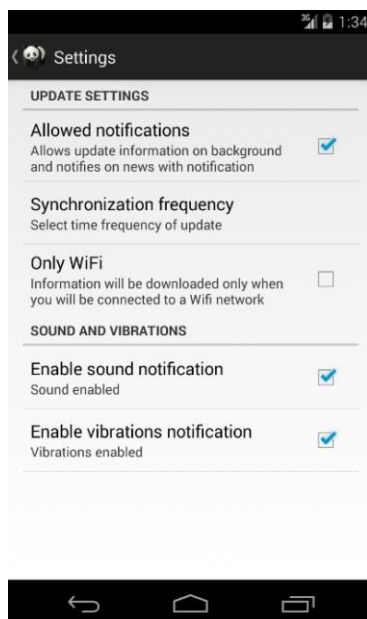
Obr. 63. Přehled rezervací Obr. 64. Detail rezervace

Při stažení aktuálních novinek na pozadí dojde k vyvolání interní notifikace a tím k upozornění uživatele na nově vzniklé události. Notifikační zpráva je uživateli zobrazena v panelu akcí. Po jejím rozvinutí vypadá tak, jak je uvedeno na Obr. 65. Po kliknutí na notifikační zprávu dojde k otevření aplikace.

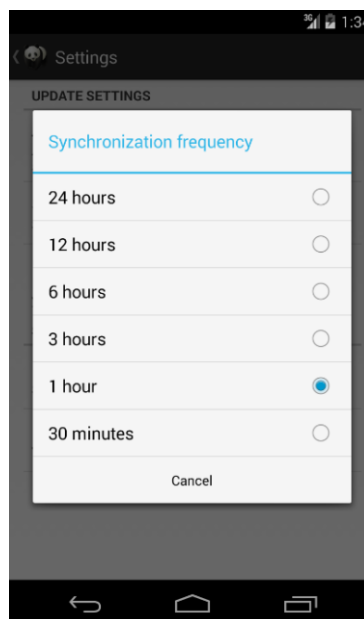


Obr. 65. Vyvolaná notifikace

Velice důležitá obrazovka v aplikaci je nastavení pro celou aplikaci. Nastavení je prakticky celou dobu skryto pod třemi tečkami vpravo nahoře u většiny obrazovek. Tato obrazovka slouží k individuálnímu nastavení stahování. Uživatel má na výběr, zda si bude aplikaci aktualizovat ručně, nebo ji nechá kontrolovat nové data na pozadí v určitém časovém intervalu. Obrazovka s nastavením je na Obr. 66 a frekvence kontroly novinek je vedle na Obr. 67.



Obr. 66. Nastavení



Obr. 67. Frekvence synchronizace aktualizací

## 11.2 Možnosti rozšíření

Vytvořená aplikace jako taková je schopna plnit reálné marketingové a informativní požadavky. Nicméně by se dala rozšířit o některé další funkce, které byly mimo rozsah této práce, ale uživatelé by je mohli přivítat. V případě reálného nasazení by se určitě jednalo o návaznost rezervačního systému vstupenek do kina přes některou platební bránu fungující pro daný subjekt, která je již využívána na daném serveru při přístupu přes webové rozhraní. Další funkcí, která by se mohla hodit při orientaci ve velkých nákupních centrech, je určitě vnitřní navigační systém, který je schopen navést uživatele do míst centra, kam potřebuje. Jako poslední funkci, kterou zde uvedu, je speciální elektronická peněženka centra, vhodná pro věrné zákazníky, kteří mohou např. získat různé slevy při platbě pomocí této peněženky, popř. jiné formy benefitů pomáhajících k udržení stávajících zákazníků.

## ZÁVĚR

Hlavním cílem této práce bylo vytvořit mobilní aplikaci určenou pro klienty obchodních center. Důraz byl kladen na pochopení, jakým způsobem lze mobilní aplikace vyvíjet, dále na analýzu operačního systému Android, jeho bezpečnosti a možností optimalizace pro různé velikosti a rozlišení obrazovek, a také přehled možností rezervace a plateb pomocí mobilního zařízení. Poté bylo úkolem s využitím těchto informací navrhnout a implementovat softwarové vybavení pro funkčnost aplikace.

Myslím, že tyto cíle práce byly splněny. V teoretické části byly podrobně popsány metodiky vývoje mobilních aplikací, analýza operačního systému Android, také jeho bezpečnost a optimalizace pro různé velikosti zařízení a hustoty obrazovek. Byly také představeny možnosti rezervace a plateb přes mobilní aplikaci.

Praktickým výsledkem je pak funkční mobilní aplikace, jež odpovídá vytvořenému zadání, požadavkům na aplikaci a jejímu návrhu. Mobilní aplikace byla vyvíjena programovacím jazykem Java pro Android a ve vývojovém prostředí Eclipse pro zařízení s operačním systémem Android 2.3.3 a vyšší, který dovoluje nasazení na přibližně 99% všech zařízení s Androidem. Celá aplikace je napsána tak, aby ji bylo možné v budoucnu rozšířit. Vhodné by bylo v případě reálného nasazení implementovat např. funkci, která umožní zákazníkovi zarezervovaný lístek do kina rovnou i zaplatit, popř. vnitřní navigaci v případě velkých obchodních center, která usnadní uživateli snazší orientaci uvnitř velkého objektu. Také by určitě nebylo od věci věnovat více času designu aplikace pro správnou volbu kombinace barev, možnosti stínování a různých grafických přechodů.

**SEZNAM POUŽITÉ LITERATURY**

- [1] GARTHER INC. *Smartphone Sales Grew 44.5 Percent in Second Quarter of 2013 and Exceed Feature Phone Sales for First* [online]. 2013 [cit. 2014-01-31]. Dostupné z: <http://www.gartner.com/newsroom/id/2573415>
- [2] KYPTA, Tomáš. *Vyvíjíme pro Android - úvod* [online]. 2011 [cit. 2014-01-31]. Dostupné z: <http://www.abclinuxu.cz/clanky/vyvijime-pro-android-uvod>
- [3] *Android Security Overview* [online]. 2013 [cit. 2014-01-31]. Dostupné z: <http://source.android.com/devices/tech/>
- [4] GOOGLE INC. *Android Applications Fundamental* [online]. 2013 [cit. 2014-01-31]. Dostupné z: <http://developer.android.com/guide/components/fundamentals.html>
- [5] *Android Security Overview. Android Open Source* [online]. 2011 [cit. 2014-01-31]. Dostupné z: <http://source.android.com/devices/tech/>
- [6] LUTONSKÝ, Marek. *O sirotcích s Androidem – MobilMania.cz* [online]. 2013 [cit. 2014-01-31]. Dostupné z: <http://www.mobilmania.cz/clanky/o-sirotcich-s-androidem/sc-3-a-1317610/default.aspx>
- [7] *Mobilní skandál, program od výrobce sleduje možná i váš mobil – idnes.cz* [online]. 2013 [cit. 2014-01-31]. Dostupné z: [http://mobil.idnes.cz/mobilni-skandal-program-od-vyrobce-sleduje-mozna-i-vas-mobil-pl5-/mob\\_tech.aspx?c=A111202\\_163821\\_mob\\_tech\\_vok](http://mobil.idnes.cz/mobilni-skandal-program-od-vyrobce-sleduje-mozna-i-vas-mobil-pl5-/mob_tech.aspx?c=A111202_163821_mob_tech_vok)
- [8] *Špehovat vás může téměř každý – idnes.cz* [online]. 2013 [cit. 2014-01-31]. Dostupné z: [http://mobil.idnes.cz/spehovat-vas-muze-temer-kazdy-bezpecnostni-dira-u-telefonu-htc-pvo-/mob\\_tech.aspx?c=A111006\\_003055\\_mob\\_tech\\_ham](http://mobil.idnes.cz/spehovat-vas-muze-temer-kazdy-bezpecnostni-dira-u-telefonu-htc-pvo-/mob_tech.aspx?c=A111006_003055_mob_tech_ham)
- [9] JOHNSON, Jeremy. *Android vs iOS: Differences in UI Patterns and Design* [online]. 2013 [cit. 2014-02-10]. Dostupné z: <http://www.slideshare.net/jeremy/android-vs-iphone-differences-in-ui-patterns-and-design>
- [10] ALLEN, Grant. *Android 4: průvodce programováním mobilních aplikací. 1. Vyd. Překlad Jakub Mužík. Brno: Computer Press, 2013, 656s. ISBN 978-80-251-3782-6.* [cit. 2014-01-31].

- [11] HILDENBRAND, Jerry. *Android App Permissions – How Google Gets it Right* [cit. 2014-01-31]. Dostupné z: <http://www.androidcentral.com/android-permissions-privacy-security>
- [12] GOOGLE INC. *Dashboard* [online]. 2014 [cit. 2014-01-31]. Dostupné z: [http://developer.android.com/about/dashboards/index.html?utm\\_content=buffer7a7fe&utm\\_medium=social&utm\\_source=twitter.com&utm\\_campaign=buffer](http://developer.android.com/about/dashboards/index.html?utm_content=buffer7a7fe&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer)
- [13] KONEČNÝ, Matěj. Zdroják.cz: *Vyvíjíme pro Android: Stylování a design*. Zdroják.cz [online]. 2013 [cit. 2014-02-03]. Dostupné z: <http://www.zdrojak.cz/clanky/vyvijime-pro-android-stylovani-a-design/>
- [14] GOOGLE INC. *Android Design Guidelines: Pure Android* [online]. 2014 [cit. 2014-02-03]. Dostupné z: <http://developer.android.com/design/patterns/pure-android.html>
- [15] SVOBODA, Štěpán. IBA.CZ. *Mobilní aplikace: HTML 5 versus nativní řešení* [online]. 2013, 27.11.2013 [cit. 2014-02-03]. Dostupné z: <https://www.ibacz.eu/blog/-/blogs/mobilni-aplikace-%E2%80%93-html-5-versus-nativni-reseni>
- [16] Mautilus [online]. 2014 [cit. 2014-02-04]. Dostupné z: <http://www.mautilus.com/cs/knowhow/vyvoj-mobilnich-aplikaci/>
- [17] HUJER, Martin. ZDROJAK.CZ. *Mobilní vývoj: Cordova a Sencha Touch aneb mobilní aplikace pomocí webových technologií* [online]. 2014, 15.1.2014 [cit. 2014-02-03]. Dostupné z: <http://www.zdrojak.cz/clanky/cordova-sencha-touch-mobilni-aplikace/>
- [18] COWARD, Jim. TECH.PRO. *When to Go Native, Mobile Web or Cross-Platform/Hybrid* [online]. 2013 [cit. 2014-02-04]. Dostupné z: <http://tech.pro/blog/1355/when-to-go-native-mobile-web-or-cross-platformhybrid>
- [19] Android Developers. GOOGLE INC. *Activity* [online]. 2014 [cit. 2014-02-04]. Dostupné z: <http://developer.android.com/reference/android/app/Activity.html>
- [20] CARROLL, Dave. *Native, HTML5, or Hybrid: Understanding Your Mobile Application Development*. Developerforce [online]. 2013, 7.11.2013 [cit. 2014-02-04]. Dostupné z: [http://wiki.developerforce.com/page/Native,\\_HTML5,\\_or\\_Hybrid:\\_Understanding\\_Your\\_Mobile\\_Application\\_Development\\_Options](http://wiki.developerforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options)

- [21] GOOGLE INC. Google Play [online]. 2014 [cit. 2014-02-04]. Dostupné z: <https://play.google.com/store>
- [22] MAIORCA, Davide, Davide ARIU, Luigi MELONI, Iginio CORONA, Radovan HOLÍK a Giorgio GIACINTO. PATTERN RECOGNITION AND APPLICATIONS LAB. Tell Me Why (You Obfuscate): Analysis and Assessment of Android Obfuscation Mechanisms for Antivirus Evasion. Cagliari, 2013. [cit. 2014-02-08]
- [23] PATEL, Pratik. THE APP ENTREPRENEUR. IOS And Android Design Guidelines [online]. 2013 [cit. 2014-02-10]. Dostupné z: <http://theappentrepreneur.com/ios-android-design-guidelines>
- [24] PETROVAN, Bogdan. Android Everywhere: 10 Types of Devices That Is Making Better. Android Authority [online]. 2012 [cit. 2014-02-12]. Dostupné z: <http://www.androidauthority.com/android-everywhere-10-types-of-devices-that-android-is-making-better-57012/>
- [25] Supporting Multiple Screens. GOOGLE INC. Developers [online]. 2014 [cit. 2014-02-12]. Dostupné z: [http://developer.android.com/guide/practices/screens\\_support.html](http://developer.android.com/guide/practices/screens_support.html)
- [26] GOOGLE INC. Fragments [online]. 2013 [cit. 2014-02-16]. Dostupné z: <http://developer.android.com/guide/components/fragments.html>
- [27] AZZOLA, Francesco. JAVA CODE GEEKS. Android Fragment Lifecycle: multiple screen support [online]. 2013 [cit. 2014-02-16]. Dostupné z: <http://www.javacodegeeks.com/2013/06/android-fragment-lifecycle-multiple-screen-support.html>
- [28] Mobile Payments in Czech Republic. Fortumo [online]. [cit. 2014-03-20]. Dostupné z: <https://fortumo.com/countries/czech-republic>
- [29] PayU: Jak fungují online platby [online]. [cit. 2014-03-21]. Dostupné z: <http://www.payu.cz/jak-funguji-line-platby>
- [30] ŠMIGALA, Andrej. Možnosti využití technologie NFC na mobilních platformách založených na OS Android. Brno, 2013. Dostupné z: [http://is.muni.cz/th/359954/fi\\_b/bp-xsmigala.pdf](http://is.muni.cz/th/359954/fi_b/bp-xsmigala.pdf). Bakalářská práce. Masarykova Univerzita.

- [31] ORAZMEDOV, Timur. Bezkontaktní platby - NFC. Brno, 2013. Dostupné z: [http://is.bivs.cz/th/12110/bivs\\_m/Diplomova\\_prace\\_Bezkontaktni\\_platby\\_NFC\\_Timur\\_Orazmedov\\_2MK-ITM.pdf](http://is.bivs.cz/th/12110/bivs_m/Diplomova_prace_Bezkontaktni_platby_NFC_Timur_Orazmedov_2MK-ITM.pdf). Diplomová práce. Masarykova Univerzita.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

|     |                                    |
|-----|------------------------------------|
| APP | Application                        |
| GPS | Global positioning System          |
| IDE | Integrated Development Environment |
| JVM | Java Virtual Machine               |
| NFC | Near Field Communication           |
| OC  | Obchodní centrum                   |
| OHA | Open Handset Alliance              |
| OS  | Operating System                   |
| OTA | Over The Air                       |
| ROM | Read Only Memory                   |
| SMS | Short Message Service              |
| UI  | User Interface                     |

## SEZNAM OBRÁZKŮ

|   |    |
|---|----|
| Obr. 1. Přehled možných mobilních aplikací a jejich hlavních kladů [20] .....         | 14 |
| Obr. 2. Celosvětový podíl operačních systémů na smartphonech za 2013 Q2 [1] .....     | 20 |
| Obr. 3. Podíl Android verzí na trhu [12].....   | 22 |
| Obr. 4. Životní cyklus aktivity [19] .....  | 24 |
| Obr. 5. Povolení, pomocí kterých aplikace žádá k používání systémových zdrojů .....   | 29 |
| Obr. 6. Příklad povolení vyžadující aplikace před instalací v Google Play [21].....   | 30 |
| Obr. 7. iOS bottom tab bar [4] .....  | 35 |
| Obr. 8. Android action bar včetně action tabs [4].....                                | 36 |
| Obr. 9. Přehled rozdílů v guidelines mezi iOS (vlevo) a Androidem (vpravo) [23] ..... | 37 |
| Obr. 10. Sliding Menu .....   | 38 |
| Obr. 11. Pohled Android kategorizace velikosti displeje v palcích [25] .....          | 39 |
| Obr. 12. Rozložení layoutů pro různé velikosti obrazovek .....                        | 39 |
| Obr. 13. Pohled Android kategorizace hustoty obrazovky [25] .....                     | 40 |
| Obr. 14. Element <support-screens> .....  | 40 |
| Obr. 15. Velikost bitmapy pro různé hustoty [25].....                                 | 41 |
| Obr. 16. Ukázka aktivity se seznamem nabídek na 7' tabletu .....                      | 42 |
| Obr. 17. Ukázka aktivity s detailem položky z předchozího obrázku na 7' tabletu.....  | 43 |
| Obr. 18. Příklad použití dvou UI komponent definovaných uvnitř fragmentů [26].....    | 44 |
| Obr. 19. Ukázka aplikace s využitím fragmentů na 7' tabletu .....                     | 45 |
| Obr. 20. Architektura klient-server .....   | 46 |
| Obr. 21. Kompletní cyklus internetové platby pomocí platební brány PayU [29] .....    | 50 |
| Obr. 22. Galeria Krakowska .....  | 54 |
| Obr. 23. Centrum Černý Most .....   | 55 |
| Obr. 24. Menu aplikace Max do vréčka.....   | 56 |
| Obr. 25. Přehled novinek v aplikaci Geric.....  | 57 |
| Obr. 26. Podíl Android verzí na trhu [12].....  | 61 |
| Obr. 27. Use-case diagram mobilní aplikace .....                                      | 66 |
| Obr. 28. Hlavní menu aplikace .....   | 68 |
| Obr. 29. Seznam novinek (první položka menu) .....                                    | 68 |
| Obr. 30. Sestupné třídění obchodů.....  | 69 |
| Obr. 31. Zobrazení kategorií všech obchodů v centru .....                             | 69 |
| Obr. 32. Fragment ‚Services‘ .....  | 70 |

|  |    |
|--|----|
| Obr. 33. Fragment ‚About‘ .....  | 70 |
| Obr. 34. Menu kina .....   | 71 |
| Obr. 35. Seznam novinek z kina .....   | 71 |
| Obr. 36. Týdenní harmonogram filmů .....   | 72 |
| Obr. 37. Detaily filmu 1 .....   | 73 |
| Obr. 38. Detaily filmu 2: Časy promítání filmu .....   | 73 |
| Obr. 39. Rezervace vstupenek .....   | 74 |
| Obr. 40. Seznam rezervací .....  | 75 |
| Obr. 41. Nastavení .....   | 75 |
| Obr. 42. Notifikace pro jednu novinku .....  | 76 |
| Obr. 43. Notifikace pro více novinek .....   | 76 |
| Obr. 44. Diagram hlavních tříd hlavní aktivity aplikace .....                                | 77 |
| Obr. 45. Class diagram třídy NewsFragment a její závislosti na používaných třídách .....     | 78 |
| Obr. 46. Class diagram popisující závislost tříd potřebných k rezervaci lístku do kina ..... | 79 |
| Obr. 47. Diagram databáze uvnitř Android aplikace .....                                      | 80 |
| Obr. 48. Jednotlivé entity třídy používané v aplikaci .....                                  | 83 |
| Obr. 49. Přehled novinek .....   | 89 |
| Obr. 50. Detail novinky .....  | 89 |
| Obr. 51. Swipe gesto mezi novinkami .....  | 89 |
| Obr. 52. Hlavní menu aplikace .....  | 90 |
| Obr. 53. Přehled obchodů podle abecedy .....   | 90 |
| Obr. 54. Přehled kategorií obchodů .....   | 90 |
| Obr. 55. Menu kina .....   | 91 |
| Obr. 56. Přehled novinek kina .....  | 91 |
| Obr. 57. Možnosti sdílení obsahu .....   | 91 |
| Obr. 58. Týdenní program kina .....  | 92 |
| Obr. 59. Film detail 1 .....   | 92 |
| Obr. 60. Film detail 2 .....   | 92 |
| Obr. 61. Obrazovka rezervace .....   | 93 |
| Obr. 62. Zamítnutí rezervace .....   | 93 |
| Obr. 63. Přehled rezervací .....   | 93 |
| Obr. 64. Detail rezervace .....  | 93 |
| Obr. 65. Vyvolaná notifikace .....   | 94 |

---

|  |    |
|--|----|
| Obr. 66. Nastavení .....                           | 94 |
| Obr. 67. Frekvence synchronizace aktualizací ..... | 94 |

**SEZNAM TABULEK**

|   |    |
|---|----|
| Tab. 1. Vlastnosti jednotlivých mobilních aplikací .....                | 18 |
| Tab. 2. Přístup k nativním prvkům jednotlivých mobilních aplikací ..... | 18 |
| Tab. 3. Interaktivní vlastnosti jednotlivých mobilních aplikací .....   | 19 |
| Tab. 4. Přehled Android verzí na trhu podle procent [12] .....          | 22 |
| Tab. 5. Přehled premium SMS pro Českou Republiku [28] .....             | 47 |
| Tab. 6. Porovnání již existujících aplikací.....                        | 58 |

**SEZNAM KÓDŮ**

|  |    |
|--|----|
| Kód 1. Ukázka knihovny kSOAP .....   | 82 |
| Kód 2. Ukázka parsování JSON na platformě Android.....   | 82 |
| Kód 3. Ukázka třídy SQLiteOpenHelper, vytváření a aktualizace tabulek v databázi<br>na Android OS..... | 84 |
| Kód 4. Ukázka metody insert v SQLite databázi.....   | 85 |
| Kód 5. Ukázka query v SQLite databázi .....  | 85 |
| Kód 6. Ukázka vkládání dat do Shared Preferences.....  | 85 |
| Kód 7. Ukázka získání dat ze Shared Preferences.....   | 86 |
| Kód 8. Ukázka implementace AlarmManageru .....   | 87 |

## SEZNAM PŘÍLOH

P I: OBSAH PŘILOŽENÉHO CD-ROM

## **PŘÍLOHA P I: OBSAH PŘILOŽENÉHO CD-ROM**

CD-ROM obsahuje následující adresáře:

- *Aplikace* – obsahuje spouštěcí .apk soubor
- *Zdrojové kódy* – obsahuje všechny zdrojové kódy aplikace
- *Fulltext.pdf* – obsahuje kompletní text této práce ve formátu PDF
- *Library* – obsahuje externí knihovnu kSOAP