

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

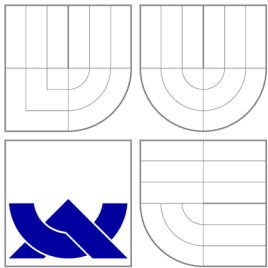
NÁSTROJ PRO VLASTNÍ TVORBU NFC ŠTÍTKŮ NA PLATFORMĚ ANDROID

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ ČERNÝ

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NÁSTROJ PRO VLASTNÍ TVORBU NFC ŠTÍTKŮ NA PLATFORMĚ ANDROID

TOOL FOR DESIGNING THE NFC TAGS ON ANDROID

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ ČERNÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. LUKÁŠ MARŠÍK

BRNO 2013

Abstrakt

Tato práce se zabývá bezdrátovou technologií NFC, její podporou na platformě Android a využitím v praxi. Popisuje návrh a implementaci aplikace, která je schopná pracovat s vlastními NFC štítky včetně jejich vytváření, upravování, mazání, a interpretace. Reálné využití NFC štítků spočívá v urychlení a automatizaci běžných úkolů a akcí, které na mobilním zařízení s operačním systémem Android často provádíme. Jedná se např. o akce, jako je změna nastavení, změna vyzváněcího profilu, ovládání hardwaru, spouštění aplikací, inicializace hovoru, odeslání zprávy, apod.

Abstract

This thesis follows up a wireless technology called NFC, its support on the Android and its use in practice. It describes the design and implementation of the application, which is able to handle own NFC tags, including their creating, editing, erasing and interpretation. The real use of NFC tags consists in speed-up and automation of common tasks and actions, which we often do on Android-powered mobile device. It could be for example: changing of settings, changing of ring profile, controlling hardware, launching applications, initializing a call, sending messages and so on.

Klíčová slova

Android, NFC, Java, tagy, úkoly, mobilní zařízení, mobilní aplikace

Keywords

Android, NFC, Java, tags, tasks, mobile devices, mobile applications

Citace

Tomáš Černý: Nástroj pro vlastní tvorbu NFC štítků na platformě Android, bakalářská práce, Brno, FIT VUT v Brně, 2013

Nástroj pro vlastní tvorbu NFC štítků na platformě Android

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Lukáše Maršíka

.....

Tomáš Černý
15. května 2013

Poděkování

Rád bych poděkoval vedoucímu této práce panu Ing. Lukáši Maršíkovi za jeho čas a rady, které mi poskytl během konzultací.

© Tomáš Černý, 2013.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	5
1.1	Současný stav	6
2	NFC	7
2.1	Co je to NFC	7
2.2	Bezkontaktní platby telefonem	8
2.3	Stav v ČR	9
2.4	Technologie NFC	9
2.5	Komunikace mezi NFC zařízeními	10
2.6	NFC tagy	10
2.7	NDEF	10
3	Android	14
3.1	Uživatelské rozhraní	14
3.2	Architektura	15
3.3	Vývoj aplikací	15
4	Android a NFC tagy	19
4.1	<i>Tag Dispatch System</i>	19
4.2	Parametry projektu pro použití NFC adaptéru	21
4.3	Filtrování pro NFC intent	22
4.4	<i>Android Application Record</i>	22
4.5	<i>Foreground Dispatch System</i>	22
4.6	Vytváření běžných typů NDEF záznamů	23
5	Návrh aplikace	24
5.1	Cílová skupina uživatelů	24
5.2	Funkce aplikace	25
5.3	Uživatelské rozhraní	26
6	Realizace aplikace	29
6.1	Dostupné prostředky pro vývoj	29
6.2	Pojmenování aplikace	29
6.3	Jazyk aplikace	29
6.4	Logo aplikace	30
6.5	Změny oproti návrhu	31
6.6	Vzhled aplikace	31
6.7	Oprávnění aplikace	32

6.8	Architektura aplikace	32
6.9	Zveřejnění aplikace	37
7	Testování a vyhodnocení aplikace	40
7.1	Testování během vývoje	40
7.2	Testování na uživateliích	40
7.3	Kapacita tagů	42
8	Závěr	45
A	Obsah CD	50
B	Plakát	51

Seznam obrázků

2.1	Možné formy <i>secure elementu</i>	8
2.2	Porovnání bezdrátových technologií	9
2.3	Porovnání běžného a kompaktního NDEF záznamu	13
3.1	Architektura <i>Androidu</i>	18
4.1	Funkce TDS	21
5.1	Zastoupení mobilních operačních systémů na trhu	25
5.2	Zastoupení verzí <i>Androidu</i> na trhu	26
5.3	Základní menu aplikace	28
5.4	Obrazovka pro vytváření a upravování operací	28
6.1	Logo aplikace	30
6.2	Porovnání původního a přepracovaného <i>action baru</i>	32
6.3	Základní menu aplikace	33
6.4	Obrazovka pro programování tagů	34
6.5	Diagram tříd aplikace	36
6.6	Diagram tříd knihovny	39
7.1	Graf výsledku testu 1	41
7.2	Graf výsledku testu 2	42
7.3	Grafy výsledků dotazníku	44

Seznam tabulek

2.1	Porovnání typů NFC tagů	11
2.2	Možné hodnoty pole TNF	12
4.1	Hodnoty pole TNF	20
4.2	Hodnoty pole RTD	20
5.1	Návrh operací pro implementaci	27
6.1	Vypuštěné či pozměněné operace	31
6.2	Použitá oprávnění aplikace	35
6.3	Způsob realizace jednotlivých operací	38
7.1	Kapacita tagu pro některé operace	43

Kapitola 1

Úvod

Tento dokument popisuje princip NFC, využití této technologie na mobilních zařízeních s operačním systémem *Android* a základní kameny potřebné pro vytvoření aplikace, která by uměla programovat vlastní NFC tagy¹, a zároveň je interpretovat.

Cílem tohoto projektu by měla být aplikace, která umožní uživatelům vytvářet vlastní tagy, které po přiložení k telefonu provedou nějakou akci. V uživatelském prostředí aplikace si uživatel vybere, které všechny akce by měl daný tag zvládat. Může si například nastavit, že daný tag vypne vyzvánění a nastaví budík na 6:00. Po potvrzení všech požadovaných akcí bude nutné přiložit k telefonu prázdný NFC tag (případně s daty, ale bude přemazán), na který se požadované akce zapíše v určitém formátu, který bude aplikace umět interpretovat. Po naprogramování stačí uživateli přiložit před spaním telefon k tagu a má zajištěno, že jej v noci nebude obtěžovat vyzvánění a bude probuzen v 6:00. Pro samotného uživatele to znamená jednorázové naprogramování tagu a jeho přiložení k telefonu před spaním, což rozhodně při dlouhodobém užívání dané akce urychluje a zefektivňuje práci. Aplikace by měla umět ovládat veškeré běžné nastavení telefonu a provádět další běžné akce.

Následující text bude strukturován podle logických celků, které bylo nutné nastudovat a zpracovat. Kapitola 2 se tedy zabývá samotnou technologií NFC, její charakteristikou a vztahem k ostatním bezdrátovým technologiím. Jsou zde také popsány NFC tagy a formát, ve kterém jsou na ně ukládána data. Krátká podkapitola je věnována bezkontaktním platbám telefonem.

Z důvodu, že cílem je napsat aplikaci pro *Android*, je nutné věnovat následující část této platformě. V kapitole 3 je popsána historie tohoto systému, jeho základní prvky a architektura, na které je postaven. Nezanedbatelná část této kapitoly se také věnuje způsobu vytváření aplikací pro tento OS.

Další je kapitola 4 popisující podporu technologie NFC na *Androidu*. Je zde popis základních prvků systému podílejících se na zpracování NFC tagů. Také jsou zde uvedeny základní předpoklady pro nastavení projektu, aby mohl programátor NFC využít a dále některé základní metody pro práci s daty naformátovanými pro tagy.

Následuje kapitola 5, která se již věnuje práci na samotné aplikaci, konkrétně jejímu návrhu. Popisuje cílovou skupinu uživatelů, pro které je aplikace určena, její primární funkce, vzhled, způsob ovládání apod.

Kapitola 6 se pak zabývá realizací tohoto návrhu. Kromě implementace základní struktury aplikace řeší také grafiku, jazyk, název a všechny ostatní aspekty, které bylo nutné promyslet. Významná část této kapitoly se věnuje i architektuře aplikace, kde jsou popsány

¹tento výraz je ekvivalentní s pojmem štítek a bude použit v následujícím textu

jednotlivé třídy a jejich význam.

V kapitole 7 je popsán způsob testování aplikace a vyvozeny závěry z výsledků, které byly získány.

Konečně kapitola 8 je věnována závěru. Zde jsou shrnuty všechny poznatky získané během práce na aplikaci a je zde zhodnocen výsledek, kterého bylo dosaženo.

1.1 Současný stav

V současné době existuje v *Google Play* obchodě několik aplikací, které nabízejí podobnou funkcionalitu. Asi nejvýraznější z nich je aplikace *NFC Task Launcher* od firmy *Tagstand*, která se zabývá vývojem aplikací využívajících NFC a prodejem NFC tagů. Aplikace je často aktualizována a přibývají jí stále nové funkce. Umožňuje programovat tagy s velkým množstvím akcí, dokáže zapisovat jak základní formát NFC tagů, který umí zpracovat jakýkoliv telefon s NFC bez jakékoliv aplikace, tak vlastní tagy, pro jejichž interpretaci je však nutné mít nainstalovanou tuto aplikaci. U některých akcí je možné naprogramovat kromě zapnutí a vypnutí odpovídající funkce i přepínání stavu.

Další aplikace nabízející podobné funkce jsou například *NFC Quick Actions*, *NFC Basic*, *NFC Smart Q*. Všechny umí v podstatě to samé, ale liší se kvalitativně, zejména s ohledem na grafické uživatelské prostředí, které je u většiny nepřívětivé, složité nebo nepřehledné. Trochu odlišnou funkci pak nabízí aplikace *AnyTAG NFC Launcher*, která si pouze zapíše identifikaci konkrétního tagu a k ní přiřadí požadované akce. Toto řešení je velice zjednodušující a jeho hlavní nedostatek je, že tagy nebudou na jiném zařízení interpretovány stejně.

Kromě těchto aplikací existují ještě aplikace přímo od výrobců telefonů, jako *Xperia SmartTags* od *Sony* a *Samsung Tectiles*. Výrobci kromě těchto aplikací nabízejí i samotné tagy, ty jsou však velice předražené.

Kapitola 2

NFC

2.1 Co je to NFC

Následující podkapitola je převzata od Doupala [21] a ze stránek *NFC Fora* [17].

Near field communication (NFC) je sada standardů bezdrátové technologie pro mobilní zařízení, která umožňuje vytvořit spojení vzájemným dotykem nebo přiblížením do krátké vzdálenosti (řadově cm). Cílem NFC není nahradit současné bezdrátové technologie, ale stát vedle nich jako rovnocenný partner s jinými možnostmi a úkoly. NFC bylo schváleno jako standard v roce 2003 a od té doby se neustále vyvíjí. V roce 2004 bylo společností *Nokia*, *Sony*, *Phillips* a dalšími založeno neziskové sdružení výrobců, vývojářů a finančních institucí s názvem *NFC Forum* (viz jejich webové stránky [15]). Nyní má přes dvě stě členů a zahrnuje další výrazné hráče jako *Broadcom*, *Docomo*, *Google*, *Intel*, *LG*, *Marvell*, *MasterCard*, *Microsoft*, *PayPal*, *Qualcomm*, *RIM*, *Samsung* nebo *Visa*. V roce 2006 vydalo *NFC Forum* technickou dokumentaci k NFC.

Mezi hlavní výhody NFC patří tyto:

Jednoduchost

Pro navázání spojení stačí přístroje přiblížit.

Všestrannost

Možnost využití v různých prostředích a situacích.

Otevřenost a standardy

Základní NFC vrstvy vycházejí z otevřených standardů.

Propojitelnost

Možnost využít pro iniciaci spojení, které pak může probíhat prostřednictvím Wi-Fi nebo Bluetooth.

Bezpečnost

NFC zařízení komunikují na krátkou vzdálenost, komunikaci je tedy obtížné zachytit.

Kompatibilita

NFC je zpětně kompatibilní s podмноžinou RFID karet.

Využití této technologie je velice široké a stále přibývají nové možnosti. Příkladem může být náhrada klíčků k vozidlu, placení parkovného, odemykání dveří, aktivace/deaktivace různých zařízení, přihlášení k počítači, přístup k citlivým datům, placení, využívání slev,

sbírání věrnostních bodů, informace o produktech a službách, reklama, nahrazení vstupenek, párování přístrojů, občanské a řidičské průkazy, pasy, vizitky a spoustu dalších.

2.2 Bezkontaktní platby telefonem

Následující podkapitola je převzata od Korba [26].

V poslední době se ve spojitosti s NFC mluví hlavně o bezkontaktních platbách pomocí mobilního telefonu. Platba je ale chápána jako operace, která vyžaduje vyšší stupeň zabezpečení. Z tohoto důvodu musí mobilní telefony pro NFC platby obsahovat kromě samotného NFC modulu i tzv. *secure element*, což je nezávislý čip, který stojí mezi NFC čipem a samotnou infrastrukturou telefonu. Jedná se o zabezpečené úložiště dat s kapacitou stovek kB. Data jsou v něm šifrována a samotný čip je chráněn bezpečnostním kódem. Je možné v něm tvořit virtuální obálky, do kterých lze vkládat jednotlivé aplikace.

Existují tři formy *secure elementu*. Nejběžnější a nejvyužívanější je *secure element* v SIM kartě. Tuto kartu vydávají operátoři, tudíž přístup je řízen jimi. Správný název pro takovou SIM kartu je *Universal Integrated Circuit Card* (UICC). Další možností je integrovaný *secure element* přímo v telefonu, což bylo použito například v telefonu *Google Nexus S*. Poslední možností je externí *secure element* například ve formě speciální microSD karty, toto řešení se ale prakticky nepoužívá. *Secure element* má velmi malou spotřebu, proto je teoreticky možné jej použít i po vybití telefonu.

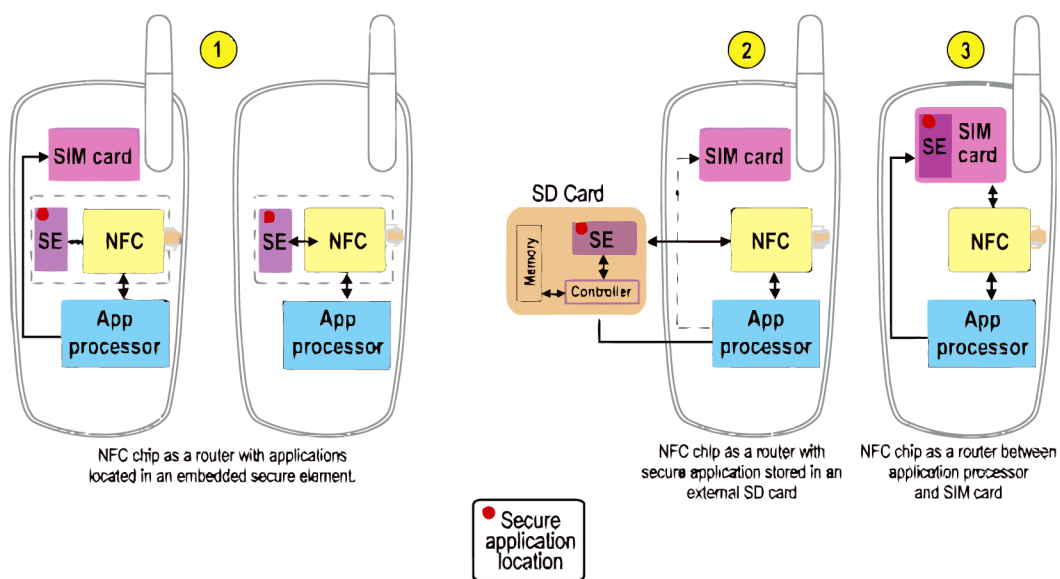


Figure 1: ① Embedded, ② SD Card-Based and ③ USIM Secure Element Solutions

Obrázek 2.1: Možné formy *secure elementu*

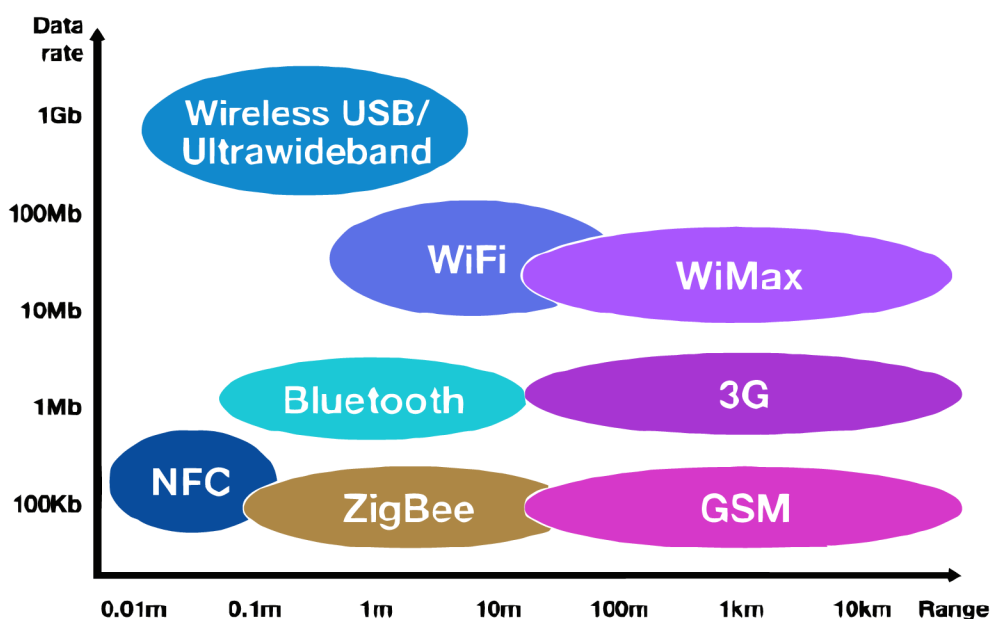
2.3 Stav v ČR

V současné době jsou NFC platby v České republice podporovány těmito společnostmi¹: *Telefonica O2*, *Komerční banka*, *Citibank Europe*, *Globus*, *Visa* a další. Za zmínku stojí také plzeňská iniciativa, díky které je v Plzni možné platit MHD pomocí NFC, a kromě toho jsou na zastávky umísťovány NFC tagy, které pomohou zjistit čas odjezdu autobusu. [21]

Velmi důležitým krokem bylo dohodnutí všech tří českých operátorů v rámci akce *NFC Forum 2012* na českém standardu pro NFC platby s názvem *Czech Wallet*. To zahrnuje využití *secure elementu* na UICC, používání centrálního hubu společného pro všechny operátory a jednotné uživatelské prostředí pomocí aplikace *Czech Wallet*. [25]

2.4 Technologie NFC

NFC využívá, na rozdíl od Wi-Fi či Bluetooth, které využívají rádiové vysílání, elektromagnetickou indukci. Jedná se v podstatě o rozšíření současných RFID karet; s jejich pasivní formou je dokonce kompatibilní. NFC je dále kompatibilní s čipy *Mifare* (marketingová značka pro RFID čipy společnosti *NXP*) i se standardem firmy *Sony* pojmenovaným *FeliCa*, který se používá hlavně v Japonsku. Komunikace probíhá na frekvenci 13,56 MHz na vzdálenost 4–20 cm. Maximální rychlost přenosu je 424 kbit/s a do budoucna se počítá s 848 kbit/s, což už ale nebude kompatibilní se všemi standardy. Na Obrázku 2.2 je porovnání technologie NFC s ostatními bezdrátovými technologiemi z hlediska rychlosti a vzdálenosti, na kterou je schopny komunikovat. [24]



Obrázek 2.2: Porovnání bezdrátových technologií [2]

¹stav na přelomu roku 2012/2013

2.5 Komunikace mezi NFC zařízeními

Následující dva odstavce jsou převzaty od Náprstka [30].

NFC komunikace může probíhat v několika režimech. Prvním z nich je komunikace mezi aktivním a pasivním prvkem, kdy na jedné straně stojí NFC čip schopný čtení i zápisu (mobilní telefon) a na straně druhé je pasivní obvod bez napájení (nejčastěji NFC tag). Aktivní prvek při komunikaci vysílá elektromagnetické vlny, které nabíjejí kondenzátor v obvodu pasivního prvku, a ten po nabití odesílá informaci zpět aktivnímu prvku. Na začátku komunikace jsou nejprve přeneseny informace o použité NFC technologii, a poté je zahájena komunikace podle odpovídajícího protokolu.

Další možností je komunikace mezi dvěma aktivními zařízeními, která má, na rozdíl od předcházejícího případu, na druhé straně také aktivní prvek s vlastním zdrojem napájení. Při komunikaci jsou zařízení na stejné úrovni — vytváří tedy malou P2P síť. Maximální rychlost komunikace je 424 kbit/s v half-duplex režimu s využitím protokolu LLCP. Příkladem je komunikace mezi dvěma mobilními telefony nebo mezi telefonem a platebním terminálem.

Poslední možností komunikace je emulace karty, která umožňuje mobilním telefonům, aby se chovaly jako NFC čipové karty.

2.6 NFC tagy

NFC tag je něco jako bezdrátová paměťová karta s malou kapacitou (jednotky kB). Tagy nejsou určeny k uložení velkého objemu dat, jako jsou fotografie či hudba, ale k uložení například webových adres, vizitek, identifikačních čísel nebo krátkých zpráv. Pro aktivaci stačí velice malý výkon (desítky μW). NFC tagy se skládají ze dvou částí — z čipu, který je složen z mnoha menších částí pro obsluhu komunikace a obsahuje paměť, a z antény. Anténa je rozměrnější část, přes kterou probíhá přenos energie a dat — je tedy limitována svou minimální velikostí (průměr alespoň 15 mm). Tagy mají často kulatý tvar ve formě samolepek, přívěšků na klíče, náramků nebo můžou mít i tvar obdélníkový ve formě plastových nebo papírových karet. Pro správnou funkci tagu je žádoucí, aby nebyl umístěn na kovovém povrchu. Existuje více typů, které se liší v použitých čípech — jejich hlavní dodavatel je firma *NXP*, což je nizozemská odnož společnosti *Phillips* (pro zajímavost má tato firma větší tržby než *Nvidia*). *NFC Forum* definovalo čtyři formáty tagů pro technologii NFC, které se liší kapacitou, přenosovými rychlostmi a možností uzamknutí tagu, aby nebylo možné měnit jeho obsah. [24]

V nedávné době představila firma *NXP* novou rodinu NFC tagů s označením *NTAG21x*, které splňují standard typu 2. Obsahují některé zajímavé novinky jako autentizaci 32bitovým heslem, 24bitový čítač přístupu, který usnadňuje analýzu využití, podpis originality, který ověřuje, zda se nejedná o kopii, a další. [3]

2.7 NDEF

Následující podkapitola je převzata z webových stránek *NFC Fora* [1].

NDEF je specifikace formátu zprávy, která je přenášena mezi dvěma aktivními NFC zařízeními, nebo mezi zařízením a NFC tagem, přičemž formát zprávy je v obou případech stejný. Jedná se o jednoduchý binární formát, který se používá pro vytvoření zprávy (mohou

	Kompatibilní tagy	Velikost paměti	Rychlost	Cena	Přístup k datům
Typ 1	<i>Innovision Topaz</i>	96 B–2 kB	106 kbit/s	nízká	R/W nebo Read-only
Typ 2	<i>NXP Mifare Ultralight</i>	48 B	106 kbit/s	nízká	R/W nebo Read-only
	<i>NXP Mifare Ultralight C</i>	144 B	106 kbit/s	nízká	R/W nebo Read-only
Typ 3	<i>Sony FeliCa</i>	1, 4, 9 kB	212 kbit/s, 424 kbit/s	vysoká	R/W nebo Read-only*
Typ 4	<i>NXP DESFire</i>	4 kB	106 kbit/s	střední	R/W nebo Read-only*
	<i>NXP SmartMX- JCOP</i>	32 kB	424 kbit/s	vysoká	R/W nebo Read-only*

* Nastaveno při výrobě

Tabulka 2.1: Porovnání typů NFC tagů [24]

být i vnořené) obsahující data jedné nebo i více aplikací. NDEF nedefinuje způsob přenosu dat — to je řešeno např. LLCP protokolem.

Funkcí NDEF je tedy zapouzdřit běžné typy dokumentů, entit včetně šifrovaných dat, XML dokumentů a fragmentů, obrázků GIF, JPEG apod. do zpráv. Je možné pracovat i s daty předem neznámé délky, čehož se dá využít při zapouzdřování dynamicky generovaného obsahu nebo pro velmi velké entity, které jsou pak ukládány zřetězené. NDEF také umožňuje agregovat více dokumentů a entit, které spolu logicky souvisejí do jediné zprávy. Malý objem dat může být zapsán pomocí kompaktní verze formátu, čímž je možné vyhnout se zbytečné režii.

NDEF zpráva se skládá z jednoho nebo více NDEF záznamů, kde každý z nich může obsahovat data běžného typu o velikost max. $2^{32} - 1$ B (pouze teoreticky). Pro přenos větších souborů je možné jednotlivé záznamy zřetěžit. Každý záznam obsahuje tři parametry:

Velikost dat

Toto pole udává velikost uložených dat v bytech. Samotné pole má označení `PAYLOAD_LENGTH` a nachází se v prvních 8 B záznamu, což umožňuje efektivně kontrolovat hranice záznamů. Jeho velikost je 1 B pro kompaktní formát (označen nastaveným SR bitem) a 4 B pro běžný formát. Nula je validní hodnota tohoto pole.

Typ dat

Toto pole slouží pro identifikaci typu dat a je označeno jako `TYPE`. Může se jednat o URI, MIME-typ a další. Díky této informaci je možné zpracovávat data příslušnou aplikací. O tom, jak bude zde uložená informace interpretována, rozhoduje pole `TNF` (*Type Name Format*). Hodnoty, kterých může toto pole nabývat, jsou shrnuty v Tabulce 2.2.

Volitelný identifikátor

Zde je uložena absolutní nebo relativní URI adresa. Použitím tohoto pole je možné odkazovat se na jiné záznamy ve stejné zprávě.

V každé zprávě je první záznam označen nastaveným MB (*Message Begin*) bitem a po-

Hodnota	Význam
0x11	Prázdný
0x01	Známy typ definovaný <i>NFC Forem</i>
0x02	MIME-typ definovaný v RFC 2046
0x03	Absolutní URI definované v RFC 3986
0x04	Externí typ definovaný <i>NFC Forem</i>
0x05	Neznámý
0x06	Nezměněný
0x07	Rezervovaný

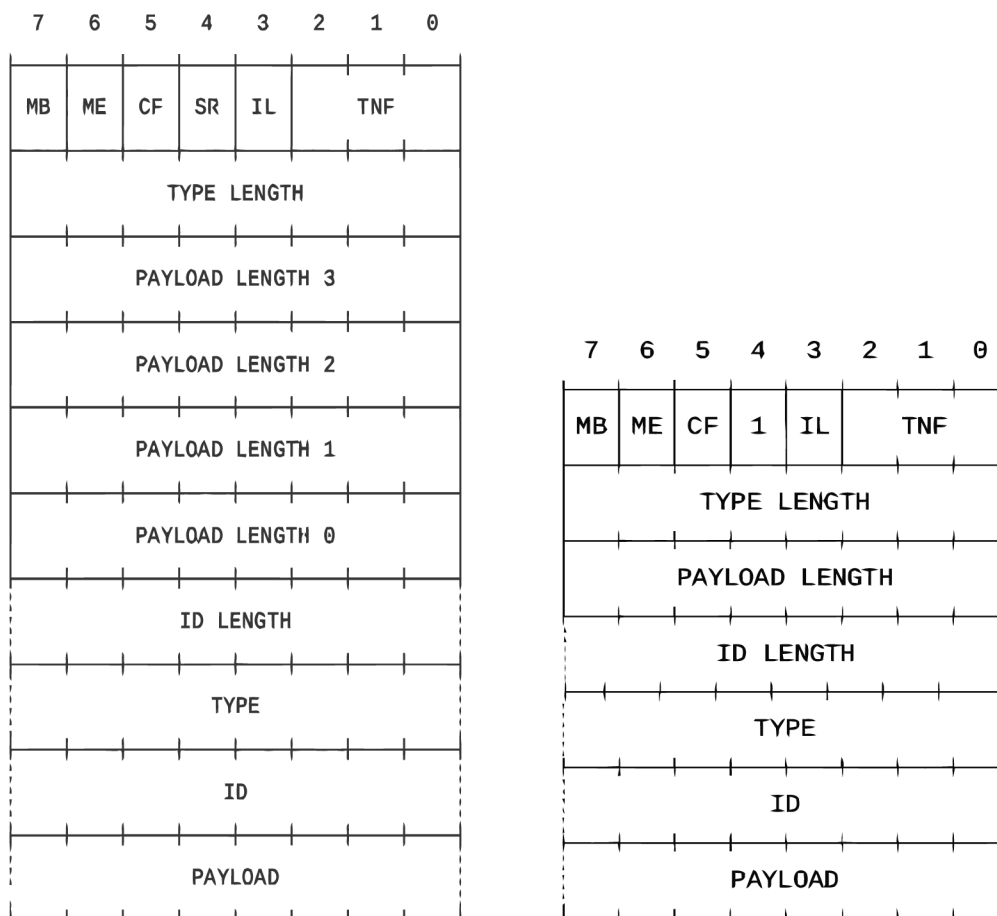
Tabulka 2.2: Možné hodnoty pole TNF [1]

slední záznam ME (*Message End*) bitem. Běžná zpráva musí obsahovat vždy alespoň jeden záznam (ten má pak nastaven MB i ME bit), pro zprávy s řetězenými záznamy jsou potřeba minimálně dva. Maximální počet záznamů ve zprávě není omezen, ale záznamy se nesmějí překrývat (ME jedné zprávy musí vždy předcházet MB zprávy následující). Jejich logické pořadí je dáno implicitně pořadím, v jakém jsou uloženy.

Pokud jsou velké objemy dat ukládány pomocí zřetězených záznamů, ty pak musí být složeny z počátečního záznamu, nula až více mezi-záznamů a z ukončujícího záznamu, přičemž všechny musí být uloženy v jediné zprávě. Typ dat musí být uložen v prvním záznamu, velikost dat v hlavičkách odpovídá velikosti jednotlivých záznamů (ne velikosti celého objemu dat). Informace o tom, že se jedná o zřetěžená data, je indikována CF bitem, toto pole tedy musí být u všech mezi-záznamů nastaveno na 1, u posledního záznamu je 0. Mezi-záznamy mají pole TNF nastaveno na 0x06, což indikuje, že se typ nemění.

NDEF využívá pro uložení dat způsob *big-endian*², přičemž nejvíce významný bit je první. Všechny NDEF parsery musí rozumět jak kompaktnímu, tak běžnému formátu NDEF. Porovnání běžného a kompaktního formátu NDEF je zobrazen na Obrázku 2.3, význam většiny polí byl vysvětlen, bit IL indikuje, zda je přítomno pole s identifikací.

²viz <http://en.wikipedia.org/wiki/Endianness>



Obrázek 2.3: Porovnání běžného a kompaktního NDEF záznamu [1]

Kapitola 3

Android

Následující kapitola je převzata od Leeho [28] a Kypty [27].

Android je operační systém založený na *Linuxu*, vyvinutý primárně pro dotykové mobilní přístroje jako smartphony, tablety, PDA nebo navigace. Původně byl vyvíjen společností *Android Inc.*, která byla v roce 2005 odkoupena společností *Google*. V roce 2007 bylo vytvořeno uskupení *Open Handset Alliance*, které zahrnovalo společnosti jako *Google*, *HTC*, *Intel*, *LG*, *Motorola*, *Nvidia*, *Qualcomm*, *Samsung*, *Texas Instruments* a další. Toto konsorcium se zabývá rozvojem otevřených standardů pro mobilní zařízení. V den jeho založení jím byl představen operační systém *Android*, což byla otevřená platforma postavená na jádře *Linuxu 2.6*. První telefon s tímto operačním systémem se začal prodávat v říjnu 2008.

Android je open-source projekt vydávaný pod licencí *Apache License*, která vyžaduje zachování prohlášení o autorských právech a zřeknutí se zodpovědnosti. Tento OS má širokou komunitu vývojářů aplikací psaných převážně v jazyku Java, které rozšiřují funkcionalitu zařízení. V současné době je dostupných více než 700 000 aplikací pro *Android* (viz Womack [34]). Tento systém ovládá zhruba 75% současného trhu s operačními systémy pro mobilní zařízení. Celkově existuje okolo 750 milionů přístrojů, které jej používají¹ (viz Page [31]).

3.1 Uživatelské rozhraní

Následující podkapitola je převzata ze stránek *Android Open Source Project* [18] a *Android Developers* [19].

Uživatelské rozhraní *Androidu* je založeno na přímé manipulaci použitím dotykových gest, která zhruba odpovídají reálným činnostem jako klepnutí, přesunutí nebo posunutí, stažení a roztažení a podobně. Systém se snaží na tyto činnosti okamžitě odpovědět plynulou reakcí, často s použitím vibrací, které uživateli poskytují zpětnou vazbu. Pro ovládání je možno rovněž využít další interní prvky zařízení, jako akcelerometr, gyroskop, proximity senzor, například pro změnu orientace displeje.

Po zapnutí přejde systém na domovskou obrazovku, což je základní navigační a informační bod, který je analogií *Plochy* na systémech PC. Na domovské obrazovce najdeme ikony, pomocí kterých spouštíme odpovídající aplikace, a dále widgety, které zobrazují aktuální obsah jako předpověď počasí, emailovou schránku nebo zprávy. Domovská obrazovka se může skládat z více stránek, mezi kterými se může uživatel pohybovat. Jedná se o velice přizpůsobivý prvek, kde je možné přidávat a odstraňovat jednotlivé elementy, a dokonce i upravovat vzhled.

¹hodnoty odpovídají přibližnému stavu v březnu roku 2013

Dalším důležitým prvkem uživatelského rozhraní je informační řádek, který se nachází úplně nahoře na obrazovce. Najdeme zde informace o zařízení jako současná aktivní připojení, vyzváněcí režim, hodiny, oznámení atd. Informační řádek je možné stáhnout dolů, což nám zpřístupňuje další informace jako dostupné updaty, přijaté zprávy, nepřijaté hovory.

3.2 Architektura

Následující podkapitola je převzata od Leeho [28].

Zdrojové kódy každé verze *Androidu* jsou dostupné pro veřejnost² až poté, co jsou vývojové verze plně vyladěny. Kód bez modifikací však bude funkční pouze na vybraných zařízeních, jako jsou přístroje řady *Nexus*. Pro ostatní zařízení je čistý *Android* upraven výrobcem pro jejich specifické účely.

Systém *Android* je postaven na pěti vrstvách. Vrstva nejbližší hardwaru je jádro, které je postaveno na jádře *Linuxu 2.6* a od verze *Ice Cream Sandwich* je použito jádro *3.x*. Využívá mnoha vlastností *Linuxu*, jako je správa paměti, sítě, procesů a zabudované ovladače. *Android* nepodporuje grafické uživatelské rozhraní *X Window System* ani úplnou sadu GNU knihoven, proto není možné přenášet linuxové aplikace na *Android* bez úprav.

Další vrstvou jsou C/C++ knihovny, které využívají různé komponenty systému. Příkladem těchto knihoven jsou například *Media Libraries* pro přehrávání video a audio souborů a zobrazování obrázků, *LibWebCore* pro internetový prohlížeč, *libc*, což je standardní knihovna jazyka C upravená pro mobilní zařízení, *OpenGL* pro vykreslování 3D grafiky a další.

Třetí vrstvou je *Android Runtime*, která obsahuje virtuální stroj *Dalvik* s JIT (*Just-in-time*) kompilací. *Dalvik* spouští *dex-code* (*Dalvik Executable*), který je obvykle překládán z *Java byte-code*. *Dalvik* využívá vlastnosti *Linuxu*, jako je správa paměti či práce s vlákny. V této vrstvě jsou také obsaženy knihovny jazyka *Java*, které se svým obsahem podobají platformě *Java Standard Edition*, ale neobsahují například knihovny pro uživatelské rozhraní (*AWT*, *Swing*), a navíc obsahují některé specifické části. Každá aplikace běží ve vlastním procesu na vlastní instanci virtuálního stroje.

Vrstva *Application Framework* je důležitá pro vývojáře, protože poskytuje přístup k velkému počtu služeb zprostředkovaných knihovnami. Základní sada zahrnuje prvky *View* pro tvorbu uživatelského rozhraní, *Content providers* ke zpřístupnění obsahu jiných aplikací, *Resource manager* k přístupu ke zdrojům, jako jsou texty a grafika, *Notification manager* pro zobrazení vlastních aplikací ve stavovém řádku a *Activity manager* pro řízení životního cyklu aplikací.

Nejvyšší vrstvou jsou pak samotné aplikace, ať už se jedná o předinstalované nebo dodatečně nainstalované. Schéma architektury *Androidu* je zobrazeno na Obrázku 3.1.

3.3 Vývoj aplikací

Pro vývoj aplikací pro *Android* se využívá *Android SDK* (*Software Development Kit*), který je volně dostupný ke stažení (viz webové stránky *Android Developers* [5]). Tento balík obsahuje API knihovny a nástroje nezbytné pro kompilaci, testování a ladění aplikací. Dále zahrnuje integrované vývojové prostředí *Eclipse*.

Aplikace sestává ze čtyř základních komponent [32]:

²viz <http://source.android.com>

Activities³

Reprezentují jednotlivé obrazovky s uživatelským rozhraním.

Services

Služby bez uživatelského rozhraní běžící dlouhodobě na pozadí.

Content Providers

Prostředky pro sdílení zdrojů mezi aplikacemi.

Broadcast Receivers

Komponenty pro přijímání broadcastových oznámení systému a pro reakci na ně.

Pro každou aplikaci je nutné vytvořit projekt, který má danou strukturu. Obsahuje zdrojové kódy a všechny další zdroje, které se v aplikaci využívají. Některé soubory jsou generovány automaticky, jiné je třeba vytvořit, pokud jsou potřeba.

Projekt obecně zahrnuje tyto soubory a složky [10]:

- **src/**

Obsahuje zdrojové soubory aktivit a ostatních tříd.

- **bin/**

Výstupní adresář pro kompilaci, najdeme zde `.apk` soubor a ostatní zkompilevané zdroje.

- **jni/**

Obsahuje zdrojové kódy vyvinuté v *Android NDK*.

- **gen/**

Obsahuje Java soubory vygenerované pomocí ADT (*Android Development Tools*).

- **assets/**

Tato složka je prázdná, je možné do ní uložit soubory, které budou beze změny zkompileovány do výsledného `.apk` souboru; například zde můžeme uložit textury a herní data při vývoji her.

- **res/**

- **anim/**

Obsahuje XML soubory, které se kompilují na objekty animace.

- **color/**

Slouží pro XML soubory popisující barvy.

- **drawable/**

Slouží pro soubory s rastrovými obrázky, *9-Patch* grafiku a pro XML soubory popisující tvary.

- **layout/**

Jsou zde XML soubory, které jsou zkompileovány na různá rozvržení obrazovky.

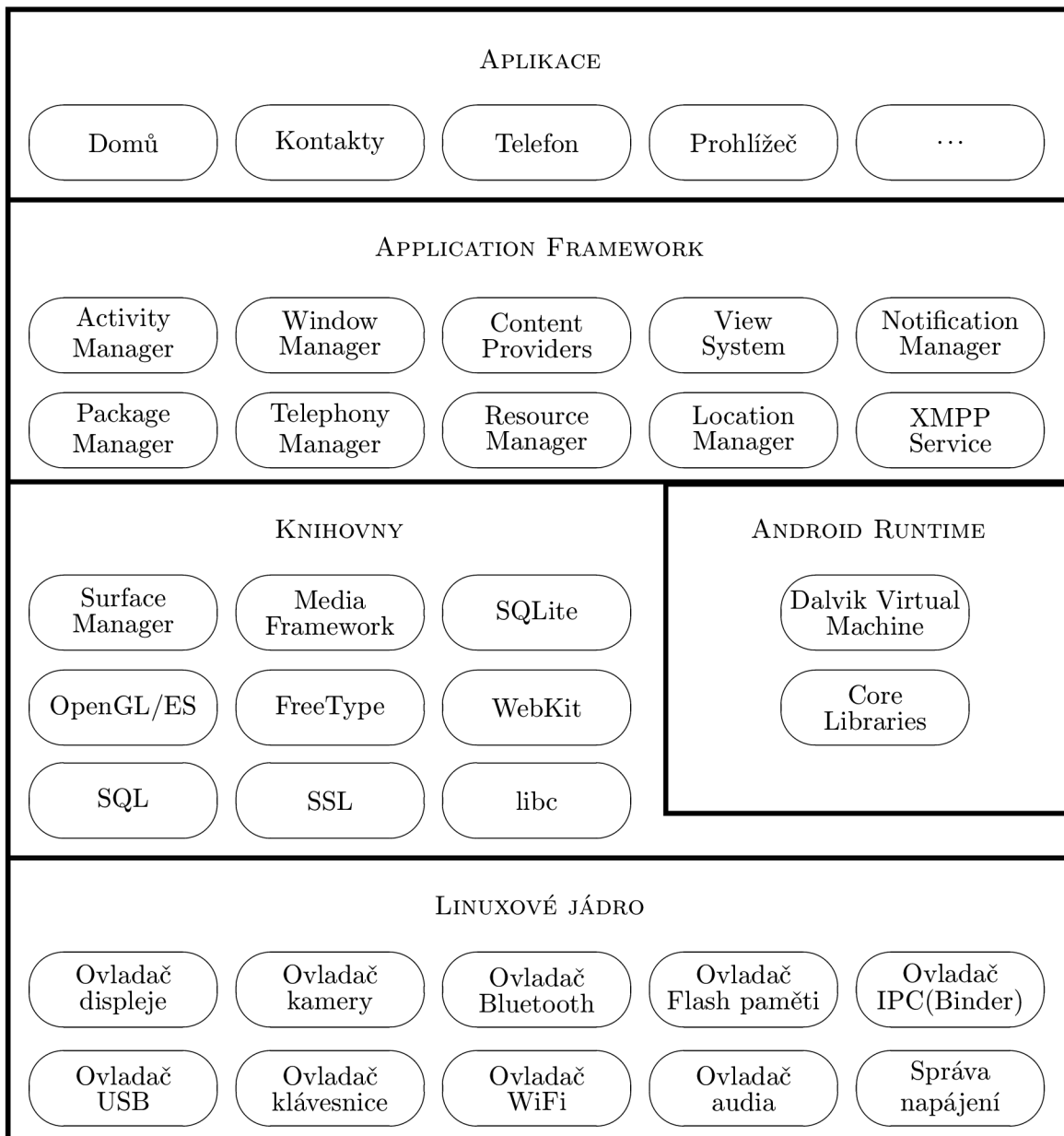
³dále bude používán pojem aktivity

- `menu/`
Tato složka je pro XML soubory, které definují menu aplikace.
 - `raw/`
Tato složka obsahuje běžné raw soubory, jediný rozdíl oproti ukládání v `assets/` je ve způsobu přístupu k těmto souborům.
 - `values/`
Slouží XML soubory, které jsou zkompileovány do různých zdrojů. Na soubory v této složce není odkazováno jménem.
 - `xml/`
Obsahuje různé další XML soubory, které slouží ke konfiguraci komponent aplikace.
- `libs/`
Obsahuje soukromé knihovny.
 - `AndroidManifest.xml`
Tento soubor popisuje aplikaci a každou z jejích komponent.
 - `project.properties`
Zde jsou uloženy informace o nastavení projektu, jako cesta pro kompilaci atd.
 - `local.properties`
Tento soubor obsahuje specifické vlastnosti místního systému potřebné pro kompilátor. *Eclipse* jej nepoužívá.
 - `ant.properties`
Obsahuje nastavení pro sestavení projektu. *Eclipse* jej nepoužívá.
 - `build.xml`
Tento soubor je potřebný pro *Ant* k sestavení aplikace.

Aplikace jsou zkompileované zdrojové soubory, které jsou společně s dalšími zdroji zabalené do ZIP archivu s příponou `.apk` — tzv. *Android package*. Tento soubor může být přímo nainstalován do zařízení s operačním systémem *Android* požadované verze. [10]

3.3.1 Vytváření knihovny

Knihovny jsou projekty obsahující zdrojový kód, na který může být odkazováno z jiných projektů. Toho se dá využít, pokud používáme společný kód pro více aplikací. Tyto projekty nemohou být přímo nainstalovány do zařízení, místo toho jsou zahrnuty do `.apk` souboru během kompilace. Co se týče struktury souborů, neliší se od běžných projektů. [10]



Obrázek 3.1: Architektura *Androidu* [7]

Kapitola 4

Android a NFC tagy

NFC tagy pro *Android* zařízení se mohou lišit svou složitostí. Jednoduché tagy nabízejí pouze čtení a zápis dat, někdy s možností uzamknutí pouze pro zápis. Složitější tagy poskytují matematické operace a mají kryptografický hardware pro autentizaci přístupu k sektorům. Data mohou být v tagu uložena v různých formátech, ale většina funkcí z *Android API* je založena na standardu NDEF. [13]

Pro *Android* existují dva hlavní případy užití při práci s NDEF:

- čtení NDEF dat z NFC tagu
- vysílání NDEF zprávy z jednoho zařízení na druhé pomocí *Android Beam* (viz stránky *Android Developers* [14])

Čtení dat je realizováno pomocí TDS (*Tag Dispatch System*), který analyzuje detekovaný tag, správně zatřídí zde uložená data a spustí aplikaci, která se o tuto kategorii dat stará. Aplikace, která chce zpracovat naskenovaný NFC tag, může definovat *intent-filter*¹ (filtr záměru) a požádat o vlastní zpracování dat. [14]

4.1 *Tag Dispatch System*

Následující kapitola je převzata z *Android Developers* [14].

V případě, že je na zařízení s *Androidem* v nastavení povoleno NFC, pak toto zařízení neustále vyhledává NFC tagy, pokud je odemčená obrazovka. Jestliže zařízení nějaký NFC tag objeví, pak je žádoucí, aby se spustila co možná nejvhodnější aplikace pro jeho zpracování, aniž by bylo nutné se ptát uživatele, jakou aplikaci chce použít. NFC pracuje na velice krátké vzdálenosti, proto by takové dotazování mohlo způsobit, že uživatel přesune zařízení dál od tagu a tím pádem zruší spojení.

Z těchto důvodů *Android* poskytuje TDS (*Tag Dispatch System*), který analyzuje NFC tagy a snaží se najít co možná nejvhodnější aplikaci. Těchto cílů dosahuje:

1. analýzou dat uložených na tagu a zjišťováním MIME-typu nebo URI, které identifikuje typ dat
2. zapouzdřením MIME-typu nebo URI a dat do intentu
3. spuštěním aplikace na základě tohoto intentu

¹pojem intent bude používán místo počestěného výrazu záměr

NDEF data jsou zapouzdřena do zprávy (`NdefMessage`), která obsahuje jeden nebo více záznamů (`NdefRecord`). *Android* podporuje i jiné typy tagů, které obsahují data v jiných formátech než NDEF, s těmito je možné pracovat pomocí tříd v *android.nfc.tech* balíku. Pro jejich použití je nutné definovat i vlastní protokol komunikace. Pro zjištění MIME-typu nebo identifikaci URI *Android* nejprve přečte první `NdefRecord` uvnitř `NdefMessage`. Tento záznam by měl mimo jiné obsahovat 3bitové pole TNF (*Type Name Format*), které značí, jakým způsobem má být interpretována hodnota pole typu (viz Tabulka 4.1). Pokud je zde uložena hodnota `TNF_WELL_KNOWN`, pak toto pole typu specifikuje RTD (*Record Type Definition*) (viz Tabulka 4.2).

Hodnota	Význam
<code>TNF_ABSOLUTE_URI</code>	URI založené na poli Typ
<code>TNF_EMPTY</code>	použije se <code>ACTION_TECH_DISCOVERED</code>
<code>TNF_EXTERNAL_TYPE</code>	URI založené na URN v poli Typ
<code>TNF_MIME_TYPE</code>	MIME-typ založený na poli Typ
<code>TNF_UNCHANGED</code>	první <code>NdefRecord</code> vadný, použije se <code>ACTION_TECH_DISCOVERED</code>
<code>TNF_UNKNOWN</code>	použije se <code>ACTION_TECH_DISCOVERED</code>
<code>TNF_WELL_KNOWN</code>	MIME-typ nebo URI závisící na RTD v poli Typ

Tabulka 4.1: Hodnoty pole TNF [14]

Hodnota	Význam
<code>RTD_ALTERNATIVE_CARRIER</code>	použije se <code>ACTION_TECH_DISCOVERED</code>
<code>RTD_HANDOVER_CARRIER</code>	použije se <code>ACTION_TECH_DISCOVERED</code>
<code>RTD_HANDOVER_REQUEST</code>	použije se <code>ACTION_TECH_DISCOVERED</code>
<code>RTD_HANDOVER_SELECT</code>	použije se <code>ACTION_TECH_DISCOVERED</code>
<code>RTD_SMART_POSTER</code>	URI založené na analýze dat
<code>RTD_TEXT</code>	MIME-typ <code>text/plain</code>
<code>RTD_URI</code>	URI založené na obsahu dat

Tabulka 4.2: Hodnoty pole RTD [14]

Pokud TDS úspěšně přiřadí MIME-typ nebo URI NDEF zprávě, pak zapouzdří tuto informaci společně s vlastními daty do intentu `ACTION_NDEF_DISCOVERED`. V některých případech se stává, že TDS nedokáže identifikovat data, protože jim nedokáže přiřadit MIME-typ nebo URI, nebo protože tag neobsahuje NDEF data. V takových případech je zapouzdřen objekt `Tag`, který obsahuje informaci o technologii tagu, společně s vlastním obsahem tagu do intentu `ACTION_TECH_DISCOVERED`.

V okamžiku, kdy je intent systémem TDS vytvořen, je zaslán aktivitě, která pro něj má filtr. Pokud je takových aktivit v zařízení více, objeví se okno pro výběr činnosti, ve kterém musí uživatel sám zvolit, kterou aplikací má být intent zpracován. TDS definuje tři typy intentů seřazené podle priority sestupně:

1. `ACTION_NDEF_DISCOVERED`

Tento intent spouští aktivitu, pokud je tag známého typu. Tento intent má největší prioritu.

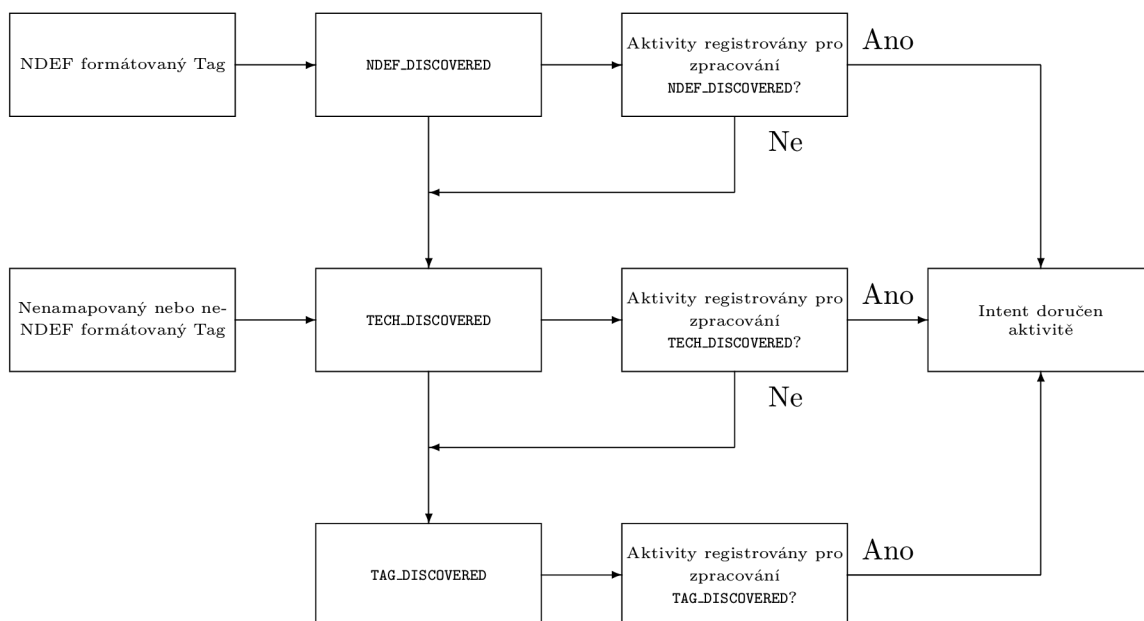
2. ACTION_Tech_DISCOVERED

Pokud žádná aplikace nereaguje na ACTION_NDEF_DISCOVERED, pak se TDS pokouší o spuštění aplikace s tímto intentem. Dalším důvodem jeho použití jsou případy, kdy není rozpoznán MIME-tyt nebo URI, nebo když tag neobsahuje NDEF data.

3. ACTION_TAG_DISCOVERED

Tento intent je použit, pokud žádná aplikace nezpracuje předcházející typy.

Funkce TDS systému je zobrazena na Obrázku 4.1. Pokud je to možné, je vhodné používat ACTION_NDEF_DISCOVERED, protože je z těchto tří typů nejvíc specifický a jeho zpracování trvá nejkratší dobu. V případě, že nemá žádná aplikace filtr pro jakýkoliv z těchto intentů, pak není provedena žádná činnost.



Obrázek 4.1: Funkce TDS [14]

4.2 Parametry projektu pro použití NFC adaptéru

Následující kapitola je převzata z *Android Developers* [14].

Aby mohla aplikace využívat NFC v zařízení, je třeba v souboru `AndroidManifest.xml` specifikovat oprávnění pomocí:

```
<uses-permission android:name="android.permission.NFC" />
```

Pokud budeme chtít aplikaci zveřejnit na *Google Play*, bude nutné aplikovat i filtr, aby ji bylo možné stáhnout pouze na zařízení s podporou NFC. Toho docílíme pomocí:

```
<uses-feature android:name="android.hardware.nfc" android:required="true" />
```

Dále je nutné nastavit minimální verzi SDK, se kterou se bude pracovat. Základní podpora NFC existuje od API verze 9, kde je však velice omezená, ideální je použít verzi minimálně 14 (odpovídá *Androidu Ice Cream Sandwich*), kde už existuje plná podpora včetně *Android Beam*.

4.3 Filtrování pro NFC intent

Následující kapitola je převzata z *Android Developers* [14].

Pro spuštění požadované aplikace po načtení NFC tagu je možné nastavit filtr pro jeden nebo i více typů intentů vytvořených pomocí TDS, nejvýhodnější je ale filtrovat pro `ACTION_NDEF_DISCOVERED`. Toho docílíme pomocí deklarace filtru pro intent tohoto typu s požadovaným MIME-typem nebo URI. Následující příklad filtruje pro NFC tagy s formátem dat NDEF, které obsahují prostý text:

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

Pokud chceme zpracovat intent typu `ACTION_TECH_DISCOVERED`, je potřeba vytvořit nový XML soubor, který specifikuje NFC technologie, které naše aktivita podporuje. Filtr je uplatněn, pokud jsou uvedené technologie podmnožinou všech podporovaných technologií daným tagem. Pokud tedy například NFC tag podporuje technologie *Mifare Ultralight* a *NdefFormatable*, pak, aby pravidlo filtru souhlasilo, musí obsahovat právě tyto dvě technologie nebo alespoň jednu z nich. V souboru `AndroidManifest.xml` pak bude odkaz na tento soubor a odpovídající intent filtr.

4.4 *Android Application Record*

Od *Androidu* verze 4.0 je možné použít také velice zajímavý typ záznamu s názvem *Android Application Record* (AAR), pomocí kterého je možné explicitně uvést, která aplikace daný tag zpracuje. Odkaz na aplikaci je zprostředkován pomocí absolutního jména balíčku dané aplikace. Záznam obsahující AAR nemusí být první záznam v NDEF zprávě, proto jej můžeme použít společně v kombinaci s MIME typem nebo URI. Pokud aplikace není na zařízení nainstalována, je otevřen obchod *Google Play*, který nabídne stažení aplikace. AAR má přednost před intent filtry. [14]

4.5 *Foreground Dispatch System*

Na *Androidu* je možné krátkodobě změnit prioritu záznamů AAR a systému intentů pomocí *Foreground dispatch systému*. Ten umožňuje aktivitě na popředí, aby měla prioritu ve zpracování přečtených NFC tagů. Bez tohoto systému by například nebylo možné pohodlně zapisovat data z vlastní aplikace na NFC tag, neboť pokaždé po přiblížení tagu k zařízení by se spustila aplikace, která obsahuje *intent-filter* pro tento typ tagů, případně dialog pro výběr akce. [6]

4.6 Vytváření běžných typů NDEF záznamů

Od API verze 14 je k dispozici metoda pro automatické vytváření URI záznamů `createUri(...)`. Dále je od API verze 16 možné použít i metody `createMime(...)` a `createExternal(...)` pro vytváření MIME a externích typů záznamů. Další typy záznamů je nutné vytvářet ručně pomocí konstruktoru třídy `NdefRecord`. [\[14\]](#)

Kapitola 5

Návrh aplikace

Při návrhu aplikace bylo nejdůležitější si nejprve ujasnit, co vlastně od aplikace chceme a pro jakou cílovou skupinu je směřována. Tyto aspekty mají nejdůležitější vliv na celý následující vývoj. Dále bylo nutné také zapřemýšlet nad tím, čím chceme, aby se aplikace odlišovala od podobných na trhu.

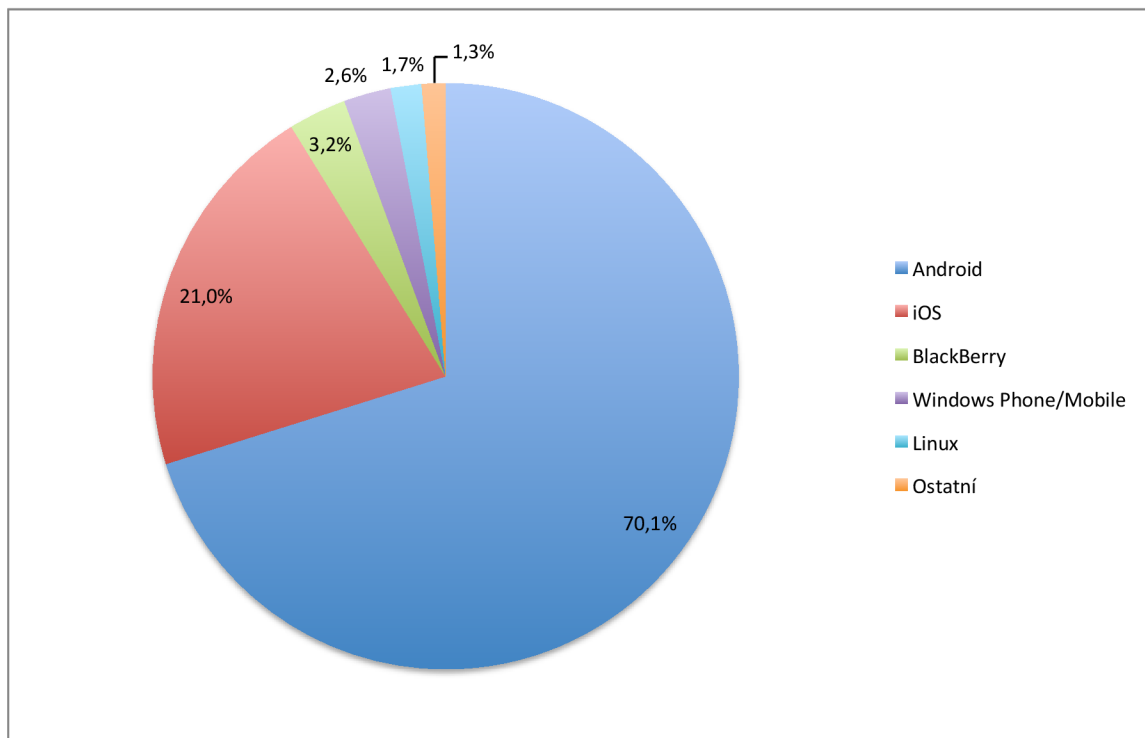
5.1 Cílová skupina uživatelů

Jedním z hlavních předpokladů pro úspěch nové aplikace je, aby byla směřována pro co možná nejširší základnu uživatelů. Z tohoto důvodu je má aplikace navrhována pro obecné použití, aby si v ní každý našel, co potřebuje. Nejedná se tedy o aplikaci zaměřenou jen a pouze na zkušené programátory či o aplikaci zabývající se pouze úzkým oborem zájmu uživatelů.

Jedním z největších faktorů určujících okruh budoucích uživatelů je volba operačního systému mobilního zařízení, pro který bude aplikace určena. V současné době jsou na trhu nejvýraznější operační systémy *Android* a *iOS*, a potom, v daleko menší míře *BlackBerry* a *Windows Phone* nebo *Windows Mobile* (viz Obrázek 5.1). Volba systému *iOS* by byla zbytečná z důvodu, že produkty firmy *Apple*, které tento systém mají, nemají NFC adaptér. Nejvýraznějším zástupcem mobilních OS je tedy *Android*, který nabízí poměrně kvalitní podporu NFC. V tomto případě je nutno také zvolit minimální SDK, které má aplikace podporovat. Kvůli některým funkcím pracujícím s NFC jsem se rozhodl pro použití minimální SDK verze 14, což odpovídá *Androidu Ice Cream Sandwich*. Tímto rozhodnutím sice znemožním použití aplikace uživatelům, kteří stále mají nižší verzi systému jako např. *Gingerbread* (cca 40% uživatelů *Androidu* — viz Obrázek 5.2), na druhou stranu je však nutné vzít v úvahu, že těchto uživatelů bude ubývat, a naopak bude přibývat uživatelů s novější verzí systému. Navíc většina telefonů s NFC již má SDK verzi 14 a vyšší.

V budoucnosti by jistě bylo možné uvažovat o konverzi aplikace pro systémy *Windows Phone/Mobile*, *BlackBerry*, případně *iOS*, pokud by nové produkty nabízely NFC.

Dalším limitujícím faktorem je, že tuto aplikaci budou moci využít pouze uživatelé, kteří mají ve svém mobilním zařízení NFC adaptér. Toto omezení vyplývá už ze samotného principu aplikace a nelze se mu vyhnout. V současné době je takovýchto zařízení poměrně málo, ale hlavně z důvodu bezdrátových plateb lze očekávat jejich vzrůstající procento (viz Muna [29]).



Obrázek 5.1: Zastoupení mobilních operačních systémů na trhu v posledním kvartálu roku 2012 [4]

5.2 Funkce aplikace

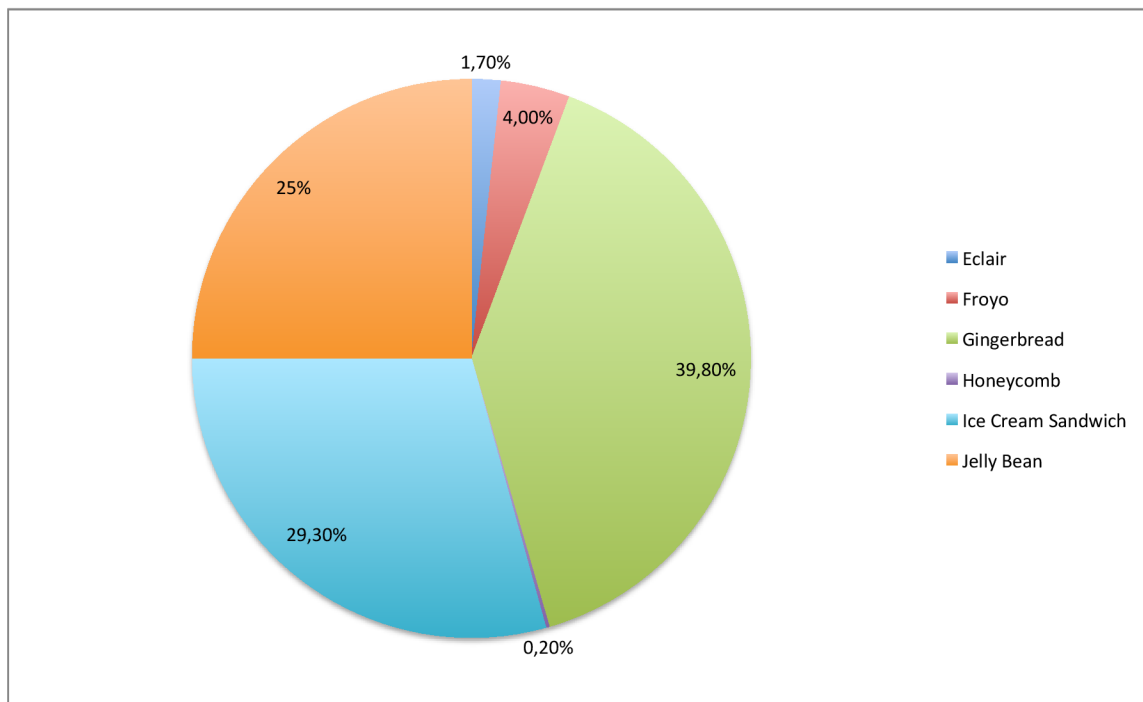
Při návrhu bylo také nutné si ujasnit, co chceme, aby aplikace uměla a jakým způsobem komunikovala s uživatelem.

Aplikace by měla uživateli nabídnout množinu operací a úkolů, které jsou na mobilním zařízení s *Androidem* běžně dostupné. Mělo by se jednat o spíše jednodušší akce, které uživatel často vykonává — složitější úkoly je pak možné složit z několika takových akcí. Uživatel by měl být schopen také zvolit konkrétní detaily dané operace. Například pokud by chtěl aktivovat budík, pak by aplikace měla být schopna nabídnout možnost zvolení na kolik hodin apod. Je možné, že se uživatel překlikne nebo se rozhodne jinak, proto by mělo být možné tyto akce i upravovat, případně mazat.

Poté, co je člověk spokojen s výběrem a nastavením jednotlivých akcí, musí mít možnost tyto akce zapsat na zvolený NFC tag. Při zapsání může dojít k situaci, kdy na tagu již budou nahrána nějaká data, případně bude zformátován jiným způsobem než jako NDEF. Kvůli těmto případům musí aplikace umět také mazat a formátovat jednotlivé tagy.

Po nějaké době se může uživatel rozhodnout, že by chtěl nějakou akci na již naprogramovaném tagu přidat, odstranit, případně upravit. Samozřejmě je možné tohoto docílit novým nastavením všech akcí, ale pro zjednodušení jsem se rozhodl aplikaci také přidat možnost načíst již naprogramovaný tag, a poté upravit konkrétní operaci, kterou chceme změnit. Není tedy třeba vkládat všechny akce znovu ručně.

Může také nastat situace, kdy určitý tag používáme na konkrétním místě, například ve formě nálepky v autě a stejné akce bychom chtěli vykonávat i jinde. Proto musí aplikace umět také klonování již existujících tagů, což znamená zapsání stejné série operací včetně



Obrázek 5.2: Zastoupení verzí *Androidu* na trhu v dubnu 2013 [9]

detailů z již naprogramovaného tagu na nový.

Z hlediska zvýšení uživatelské přívětivosti je také vhodné zahrnout do aplikace nápovědu, která nové uživatele poučí a upozorní na některé důležité detaily.

5.2.1 Návrh operací pro naprogramování

Při výběru akcí, které by měly jít na tag zapsat, bylo nutné vycházet z možností, které nabízí čistý *Android*, protože jednotliví výrobci vytvářejí vlastní nadstavby a některé funkce či nastavení, které jsou dostupné na zařízení např. od firmy *HTC*, nemusí být dostupné na smartphonu např. od *Samsungu*. V Tabulce 5.1 je rozřazen do kategorií seznam operací, které byly navrženy pro implementaci.

5.3 Uživatelské rozhraní

Již při návrhu jsem se rozhodl klást veliký důraz na uživatelské rozhraní aplikace. V současné době je v obchodě *Google Play* dostupné nepřehledné množství aplikací, od těch velice propracovaných a profesionálních, až po ty nejjednodušší. Spousta z nich však trpí ne příliš propracovaným uživatelským rozhraním. Uživatel často dlouho tápe, než nalezne, co hledá. Některé věci bývají udělány složitěji, než je potřeba, některé akce, které by uživatel potřeboval, nelze provést vůbec a podobně. V tomto ohledu se dosti přikláním k filozofii Steva Jobse a společnosti *Apple*, pro které je design a co možná nejjednodušší uživatelské rozhraní důležité stejně jako samotná funkčnost produktu. Někteří programátoři se snaží ohromit velkým množstvím různých nastavení a voleb, já ale zastávám názor: čím jednodušší, tím lepší.

Připojitelnost a umístění	Wifi on/off Wifi připojit WiFi hotspot Bluetooth on/off Mód letadlo GPS on/off Mobilní data on/off Určování polohy Google Poloha a vyhledávání	Displej	Jas Notifikační dioda Auto-rotace Režim spánku Velikost písma
		Zprávy	Poslat email Poslat SMS
		Media	Přehrát/zastavit Další skladba Předchozí skladba
Zvuk a hlasitost	Zvukový profil Vyzváněcí tón Hlasitost vyzvánění Hlasitost budíku Hlasitost médií Vibrace Zvuk upozornění Systémové zvuky	Aplikace a služby	Spustit aplikace Otevřít URL Navigovat Svítilna Zastavit procesy Zrcadlo Vytvořit poznámku Úsporný režim Uzamknout obrazovku Dock v autě
Kalendář	Událost v kalendáři Časové razítko		
Hovory	Vytvořit hovor		

Tabulka 5.1: Návrh operací pro implementaci

5.3.1 Základní menu

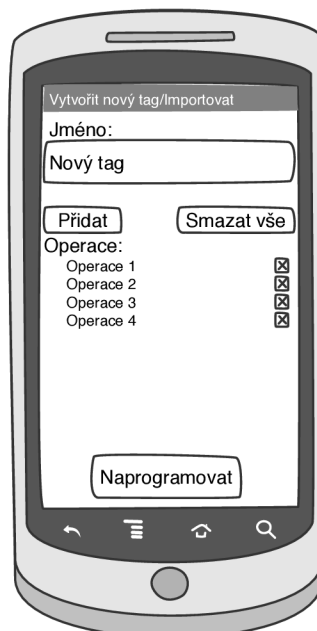
Při tvorbě základního menu jsem se rozhodl vzít si inspiraci z menu na zařízeních *Android* a *iOS*, kde jsou na obvykle tmavém pozadí ikony jednotlivých aplikací s piktogramem, který uživateli jednoznačně napoví, co stisknutí daného tlačítka spustí. Pro úplnost je ještě pod ikonou název aplikace. Podobného principu využiji i já, ale místo aplikací budou ikony spouštět jednotlivé akce aplikace. Uživatel tedy nebude hned po spuštění zmaten, ale přivítá ho známé prostředí, kde bude jednoznačně vědět, co dělat. Stisk kterékoliv ikony jej dovede na obrazovku, kde bude moci konkrétní akci provést. Ikony by měly být v jednotném stylu, aby byla dodržena konzistence aplikace.

5.3.2 Hlavní obrazovka pro vytváření a upravování operací

Při programování tagů by uživatel měl mít přehled o všech nastavených operacích, měl by mít možnost přidávat další, mazat a upravovat je, a následně je zapsat na zvolený NFC tag. Při nahrání již existujícího tagu je možné využít stejné obrazovky s vyplněnými údaji.



Obrázek 5.3: Základní menu aplikace



Obrázek 5.4: Obrazovka pro vytváření a upravování operací

Kapitola 6

Realizace aplikace

V této kapitále se dále budu zabývat samotnou realizací aplikace od vymýšlení jména po technické detaily. Budou probírána úskalí, na která jsem během řešení narazil, a jak jsem s nimi naložil.

6.1 Dostupné prostředky pro vývoj

Vývoj aplikace probíhal na počítači s operačním systémem *OS X Mountain Lion* prostřednictvím *Android SDK*, který obsahuje integrované vývojové prostředí *Eclipse* s pluginem *ADT (Android Developer Tools)*. Tento balík obsahuje také emulátor, na kterém je možné aplikaci testovat, nicméně není možné emulovat NFC adaptér a tagy, proto nebyl použit.

Jako referenční zařízení pro testování posloužil mobilní telefon *HTC One X* se systémem *Android Jelly Bean* ve verzi 4.1.1, což odpovídá SDK verzi 16. Pro testování funkčnosti byly použity NFC tagy s čipy *Mifare Classic 1K*, *Mifare Classic 4K* a *Mifare Ultralight*.

6.2 Pojmenování aplikace

První věcí, kterou bylo na začátku realizace nutné vymyslet, bylo pojmenování aplikace. Rozhodl jsem se v názvu použít slovo NFC, aby bylo na první pohled patrné, že s touto technologií pracuje. Název by měl být poměrně jednoduchý (v *Google Play* obchodě je možné nalézt aplikace, které jsou v podstatě pojmenovány svým popisem) a lehce zapamatovatelný a měl by, pokud možno, co nejlépe popisovat i funkci aplikace. Podobné aplikace jsou pojmenovány například *NFC Task Launcher*, *NFC TagWriter*, *NFC Tag Writer & Reader*.

Po zvážení těchto faktů jsem se rozhodl aplikaci pojmenovat *NFC TaskTags*. Z tohoto názvu je patrné, že aplikace pracuje s NFC a má co dočinění s tagy a úkoly. Přestože některé aplikace podporují i jiné lokalizace včetně přeložení názvu, rozhodl jsem se toto pojmenování zachovat pro všechny jazyky. V některých jazycích totiž název může působit nereprezentativně, a navíc jsem názoru, že jméno je jen jedno, bez rozdílu jakým jazykem člověk mluví.

6.3 Jazyk aplikace

Aplikace byla od počátku psána v anglickém jazyce, protože tento jazyk používá zhruba 375 miliónů lidí jako svůj rodný jazyk, dalších 375 miliónu jako druhý jazyk a dalších přibližně

750 miliónů se jej učí jako cizí jazyk (viz Graddol [22]). Systém *Android* je z velké části používán právě v zemích, kde tomuto jazyku rozumí.

Všechny texty jsou uloženy mimo zdrojový kód v souboru `strings.xml` ve složce `/res/values` (viz podkapitola 3.3). Z tohoto důvodu je možné aplikaci kdykoliv poměrně jednoduše rozšířit o další lokalizace přidáním souboru `strings.xml` s vlastními texty do složky `/res/values-kód`, kde kód je označení jazyka nebo kombinace jazyk-region (např. `values-fr` bude pro Francii a `values-ja` pro Japonsko). Systém *Android* pak sám vybere lokalizaci podle nastaveného jazyka systému zařízení. [8]

Po dokončení aplikace byla pro demonstraci výše popsaného způsobu přidána lokalizace do češtiny, která bude implicitně použita na zařízeních, která mají tento jazyk systému nastaven.

6.4 Logo aplikace

Logo aplikace bylo vytvořeno v programu *Gimp*¹. Při jeho vytváření bylo použito tzv. *N-Mark*, což je univerzální symbol pro NFC vytvořený *NFC Forem* (viz webové stránky *NFC Fora* [12]). Tento symbol je v logu použit ve slově NFC jako náhrada za klasické písmeno *N*. Je to nejdominantnější část loga, která by měla v uživateli evokovat, že se jedná o program využívající NFC.

Jako pozadí loga bylo zvoleno pozadí již dříve zmíněného *N-Mark*. V dolní části loga je pak využito toho, že obě slova *Task* i *Tags* začínají stejně, a proto je jejich společný základ výraznější. Celé logo je na Obrázku 6.1



Obrázek 6.1: Logo aplikace

¹viz <http://www.gimp.org>

6.5 Změny oproti návrhu

Při realizaci aplikace byly provedeny určité změny oproti původnímu návrhu. Jednou z nich je vypuštění položky *Nastavení* z menu aplikace, ukázala se totiž jako nadbytečná. Všechna nastavení byla přesunuta přímo k úkolům, se kterými souvisí. Do budoucna je možné tuto položku přidat a umožnit v aplikaci měnit některé věci jako např. vzhled atpod.

Dále bylo nutné upustit od implementace některých operací, které by mohla aplikace provádět. Ukázalo se totiž, že některé navrhované úkoly lze provádět pouze na zařízeních *HTC*, a že některá nastavení lze měnit pouze systémovými aplikacemi, případně je nelze měnit vůbec. V Tabulce 6.1 jsou popsány některé změny.

Nastavení	Důvod
Wi-Fi hotspot	nelze měnit konfiguraci
Mód letadlo	od Androidu 4.2 může měnit pouze systém
GPS	může měnit pouze systém
Určování polohy Google	není API
Poloha a vyhledávání	není API
Systémové zvuky	nelze měnit vůbec
Svítilna	dostupné pouze na <i>HTC</i>
Vytvořit poznámku	ukázalo se jako zbytečné
Úsporný režim	dostupné pouze na <i>HTC</i>
Zamknout obrazovku	nelze přímo

Tabulka 6.1: Vypuštěné či pozměněné operace

6.6 Vzhled aplikace

Systém *Android* nabízí od verze *Ice Cream Sandwich* tři základní systémová témata, na kterých lze aplikaci stavět. Jedná se o témata *Holo Dark*, které je tmavé se světlým textem, *Holo Light*, které je světlé s tmavým textem a *Holo Light* s tmavým *action barem*². Témata kromě barev všech položek a pozadí také definují vzhled ovládacích prvků. Je samozřejmě možné vytvořit si celý vlastní styl, kvůli zachování konzistence se systémovými aplikacemi jsem ale rozhodl právě jedno z těchto předpřipravených témat použít, konkrétně téma *Holo Dark*. [16]

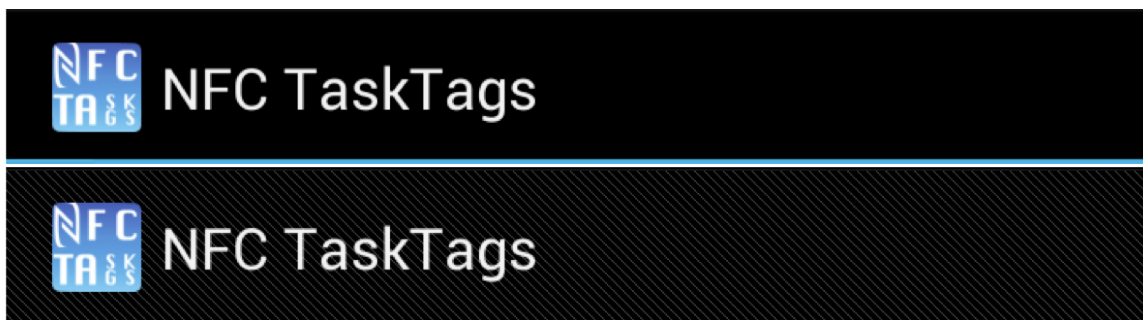
Oproti původnímu vzhledu byl pozměněn vzhled *action baru* ve stylu *action baru* v aplikaci *Google Play*. Srovnání výchozího a nového vzhledu je na Obrázku 6.2.

6.6.1 Základní menu aplikace

Menu aplikace bylo realizováno podle návrhu na Obrázku 5.3. Byly použity ikony s jednotným vzhledem se stejným pozadím, jako je pozadí ikony aplikace. Jednotlivé ikony obsahují bílé piktogramy značící význam položek. Jako zdroj těchto piktogramů byly použity systémové ikony pro *action bar*. Při potenciálním budoucím rozšíření aplikace je možné přidávat do tohoto menu další položky.

Oproti původnímu návrhu ubyla položka *Nastavení*, protože všechna jednotlivá nastavení byla přesunuta přímo k odpovídajícím akcím. Finální podoba je zobrazena na Obrázku 6.3.

²*action bar* je neměnná lišta s často používanými tlačítky při horním okraji obrazovky



Obrázek 6.2: Porovnání původního a přepracovaného *action baru*

6.6.2 Základní obrazovka pro programování tagů

Při realizaci této obrazovky jsem vycházel z návrhu na Obrázku 5.4. Jako základ celého ovládání slouží *action bar* v horní části obrazovky. V jeho levé části je ikona pro navrácení do hlavního menu a název obrazovky. V pravé části jsou pak hlavní ovládací prvky, konkrétně je zde tlačítko pro přidávání operací, tlačítko pro smazání všech nastavených operací a tlačítko pro zápis na tag. Všechna tato tlačítka jsou opět tvořena piktogramy.

V hlavní části obrazovky je pak textové pole pro jméno tagu, které je ve výchozím stavu zakázáno. Je možné jej povolit a použít zaškrtnutím položky *Pojmenovaný tag* v sekci *Volby*. Dále je zde možnost *Pouze ke čtení*, po jejímž vybrání se aplikace při zápisu na tag pokusí tento tag uzamknout, aby jej nebylo možné přemazat. Následuje sekce *Akce*, v níž jsou zobrazeny všechny již nastavené operace. Na každou z operací je možné kliknout a upravit její nastavení, dále je možné každou položku smazat kliknutím na křížek vedle názvu akce. Vzhled této obrazovky je zobrazen na Obrázku 6.4.

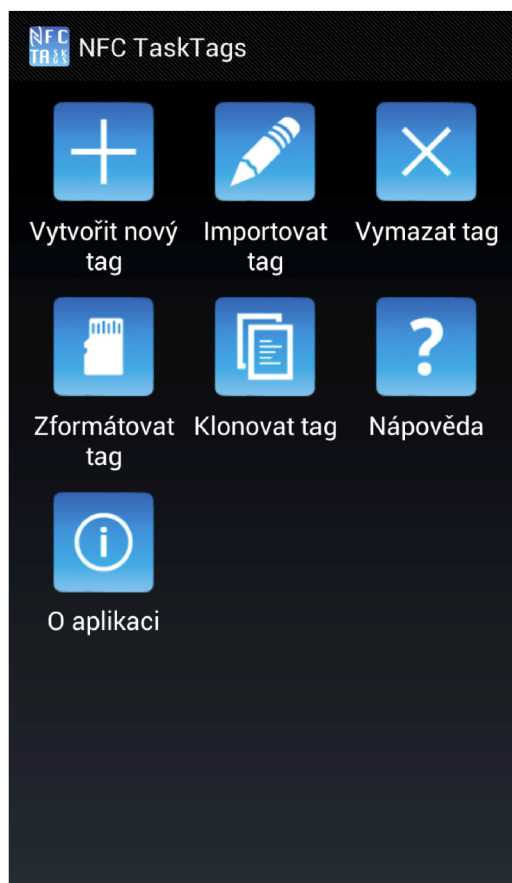
6.7 Oprávnění aplikace

Vzhledem k tomu, že aplikace dokáže ovládat širokou škálu nastavení a služeb zařízení, je nutné specifikovat nutná oprávnění do souboru `AndroidManifest.xml`. Uživatel je poté při instalaci obeznámen se vším, k čemu má aplikace přístup a musí tyto podmínky odsouhlasit, aby mohl aplikaci začít používat. Použitá oprávnění jsou zobrazena v Tabulce 6.2.

6.8 Architektura aplikace

V následující sekci budou popsány některé zajímavosti implementace. Aplikace původně vznikala jako jediný projekt se všemi třídami v jednom balíku. Součástí zadání ovšem bylo implementovat knihovnu funkcí, která bude realizovat tvorbu a využití NFC štítků. Z tohoto důvodu byl původní jeden projekt rozdělen na dva, kde jeden zahrnuje v podstatě jen třídy starající se o uživatelské rozhraní aplikace (dále projekt aplikace) a druhý, realizující všechny důležité funkce (dále projekt knihovny).

Projekt, který obsahuje hlavní funkce aplikace tedy slouží jako knihovna. Díky tomuto je možné tuto knihovnu použít z jakéhokoliv jiného projektu pouze přidáním reference na knihovnu a uvedením názvu balíku, ve kterém se konkrétní funkce nacházejí, bez nutnosti kopírovat zdrojový kód.



Obrázek 6.3: Základní menu aplikace

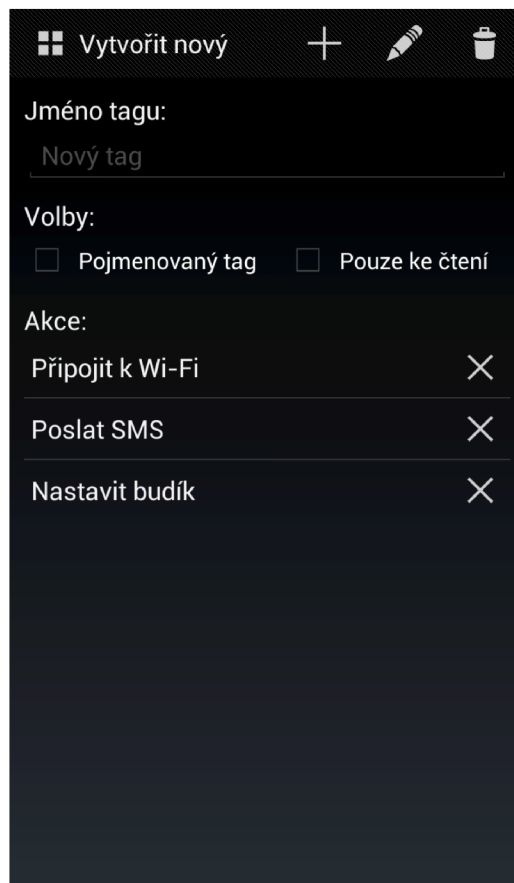
Při psaní zdrojových souborů byly dodržovány určité konvence pro větší přehlednost kódu. Všechny aktivity jsou pojmenovány jako `ActivityNázev.java`, ostatní třídy nesou název vyjadřující jejich hlavní funkci. Soubory s rozvržením obrazovky jsou pojmenovány jako `activity_název.xml` a soubory s rozvržením dialogů jako `dialog_název.xml`. Pro názvy prvků (třídy, funkce, proměnné) je používán způsob *CamelCase*³. V ostatních případech jsou dodržovány běžné konvence psaní aplikací pro *Android* (názvy ikon, soubory s texty atd.).

6.8.1 Projekt aplikace

Tento projekt shromažďuje všechny třídy, které mají souvislost s uživatelským rozhraním aplikace. Jedná se o jednotlivé obrazovky, dialogy, grafiku aplikace apod. Balík, ve kterém se všechny třídy nalézají nese název `com.nomak.nfctasktags`.

Hlavní třídou je třída `ActivityMain`. Jedná se o aktivitu, která je spuštěna ihned po otevření aplikace. Toto chování je nutné specifikovat v souboru `AndroidManifest.xml`, kde je ke specifikaci aktivity přidán speciální *intent-filter*. V rámci této třídy je na obrazovce vykresleno hlavní menu aplikace a specifikovány akce pro stisk jednotlivých ikon. Po jejich stisku je zpravidla vytvořen explicitní intent pro ostatní aktivity.

³viz <http://en.wikipedia.org/wiki/CamelCase>



Obrázek 6.4: Obrazovka pro programování tagů

Třída, která nese popisek a odkaz na obrázek ikony, se jmenuje `MenuItem`. Je využívána, kromě hlavní třídy, i třídou `ImageAdapter`. Jedná se o adaptér pro položky hlavního menu — tedy vlastně jakýsi most mezi jednotlivými položkami a mřížkou (`GridView`) hlavní obrazovky. Obsahuje svoji nejdůležitější funkci `getView(...)`, která definuje co, a jakým způsobem se bude vykreslovat [33].

Další velice důležitou třídou je `ActivityNewTag`, což je aktivita pro programování tagů. Jedná se tedy v podstatě o nejpoužívanější část aplikace (z hlediska uživatele). Je spuštěna poté, co uživatel z hlavního menu vybral položku *Vytvořit nový tag*, případně poté, co je přečten tag v módu *Import tagu* a její vzhled je na Obrázku 6.4. Její hlavní funkcí je zpracovávat vstupy z dialogů pro výběr úkolů tak, aby měl uživatel přehled o tom, co již nastavil, přičemž je zároveň na pozadí upravován objekt představující tag, který bude následně serializován a naprogramován.

Pro zobrazení již zvolených akcí v aktivitě `ActivityNewTag` se používá objekt třídy `ListView` s adaptérem definovaným ve třídě `ListAdapter`. Jedná se opět o adaptér, který zapouzdřuje názvy jednotlivých úkolů a tlačítka pro jejich smazání.

Po stisknutí tlačítka se znakem plus v *action baru* aktivity `ActivityNewTag` se zobrazí dialog pro výběr konkrétního úkolu, který chce uživatel na tag zapsat. Vzhled a chování tohoto dialogu je implementováno třídou `DialogActionChooser`, která rozšiřuje třídu `DialogFragment`. Pro vytvoření dialogu je použita třída `AlertDialog.Builder`. V tomto

Hodnota	Význam
ACCESS_NETWORK_STATE	Umožní přístup k informacím o sítích
ACCESS_WIFI_STATE	Umožní přístup k informacím o Wi-Fi sítích
BLUETOOTH	Umožní připojit se ke spárovaným zařízením
BLUETOOTH_ADMIN	Umožní objevit a spárovat zařízení
CALL_PHONE	Umožní vytvořit přímý telefonní hovor
CHANGE_NETWORK_STATE	Umožní měnit stav připojení k sítím
CHANGE_WIFI_STATE	Umožní měnit stav připojení k Wi-Fi sítím
KILL_BACKGROUND_PROCESSES	Umožní ukončit procesy na pozadí
NFC	Umožní provádět V/V operace skrz NFC
READ_CONTACTS	Umožní číst uživatelské kontakty
READ_SYNC_SETTINGS	Umožní číst nastavení synchronizace
SEND_SMS	Umožní zasílat SMS
SET_ALARM	Umožní vyslat intent pro nastavení budíku
WRITE_CALENDAR	Umožní zapisovat do kalendáře
WRITE_SETTINGS	Umožní číst a zapisovat systémová nastavení
WRITE_SYNC_SETTINGS	Umožní zapisovat nastavení synchronizace

Tabulka 6.2: Použitá oprávnění aplikace [11]

dialogu je seznam kategorií akcí, který se po kliknutí na jednu z nich rozbíjí v seznam konkrétních úkolů. Této struktury je možné dosáhnout použitím objektu třídy `ExpendableListView` s jednoduchým adaptérem `SimpleExpendableListAdapter`, který je součástí *Android API*. Tato třída také definuje rozhraní `ChooserListener` s metodou `chooserPositiveClick(...)`, které je implementováno třídou `ActivityNewTag`, díky čemuž je zajištěna komunikace mezi těmito třídami.

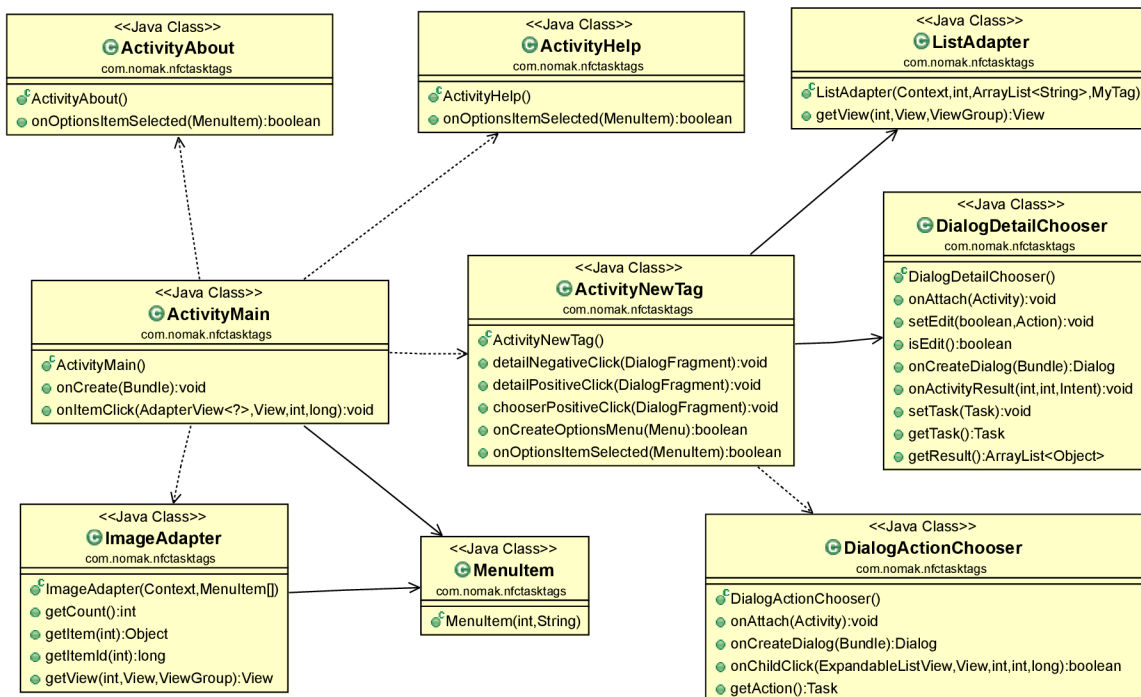
Podobným způsobem komunikuje `ActivityNewTag` se třídou `DialogDetailChooser`, která zapouzdřuje dialog, jenž se zobrazí po vybrání konkrétního úkolu z předchozího dialogu. Tato třída definuje rozhraní obsahující metody `detailPositiveClick(...)` pro odsouhlasení a `detailNegativeClick(...)` pro zrušení dialogu. Uživatel si zde může nastavit jednotlivé detaily akce, která se má provést, a po jejich potvrzení je dialog uzavřen a úkol je přidán do seznamu. Tento dialog nabývá různých vzhledů a rozvržení v závislosti na tom, jaký úkol je nastavován. Pro předání detailů zpátky aktivitě `ActivityNewTag` se používá pole objektů, ve kterém jednotlivé položky nabývají jednoduchých typů jako `string`, `bool`, `int` apod.

Výše popsané třídy včetně svých veřejných metod a závislostí jsou zobrazeny v diagramu tříd na Obrázku 6.5.

6.8.2 Projekt knihovny

Projekt *NFCTaskTagsLibs* shromažďuje všechny třídy, které se starají o vnitřní běh aplikace. Je teoreticky možné tyto třídy použít i v jiné aplikaci, proto je tento projekt nastaven jako knihovna. Třídy jsou přístupné v balíku `com.nomak.nfctasktags.lib`. V následujících odstavech jsou jednotlivé třídy popsány.

Třída `MyTag` zapouzdřuje všechna data, která jsou následně serializována a zapsána na NFC tag — jedná se tedy o jeho datovou reprezentaci. Obsahuje pouze dvě proměnné, a to jméno tagu a list akcí uložených v objektu třídy `Action`. Kromě těchto proměnných



Obrázek 6.5: Diagram tříd aplikace

obsahuje také spoustu metod pro práci s nimi. Instance této třídy je často předávána mezi objekty ostatních tříd, přičemž existuje vždy pouze jedna instance této třídy. Z tohoto důvodu je u této třídy použit návrhový vzor *singleton* (viz Hrebenar [23]), což usnadňuje přístup k jejímu objektu.

Každá úloha v tagu je uložena v objektu třídy *Action*. Tento objekt obsahuje informaci o konkrétním úkolu, která je vyjádřena výčtovým typem *Task*, a dále list textových řetězců popisujících detaily a nastavení úkolu. Kromě těchto proměnných také obsahuje metody pro práci s nimi.

ActivityNfcWrite je aktivita, která vybízí uživatele, aby přiložil telefon k tagu pro zápis informací na něj. Tato aktivita je použita pro vytváření nového tagu, klonování tagu a mazání a formátování tagu. Při spuštění *ActivityNfcWrite* je aktivován *Foreground Dispatch System*, který čeká na přiblížení tagu. Po jeho načtení je zavolána metoda objektu třídy *NfcTagWriter*, která vykoná jednu z akcí: zápis, formát, vymazání.

Aktivita *ActivityCloneRead* slouží pro načtení již naprogramovaného tagu. Analogicky jako předchozí aktivita spouští metodu třídy *NfcTagReader* a po načtení informací z tagu jsou tato data poskytnuta třídě *ActivityNfcWrite*, která se postará o zapsání kopie dat na nový tag.

NfcTagWriter je třída ovládající *Foreground Dispatch System* a realizující všechny operace, které se týkají zápisu dat na tag. Při zápisu je nejprve otestováno, zda je tag naformátován pro NDEF. Pokud ano, musí být tag schopen zápisu a musí mít dostatečnou kapacitu pro data. Pokud všechny podmínky splňuje, jsou data zapsána, případně je i tag ochráněn proti dalšímu přepisování. Při mazání tagu je zapsána prázdná zpráva a pro formátování je využito standardní API pro práci s NFC tagy.

Třída *NfcTagReader* funguje obdobně jako předchozí třída s tím rozdílem, že data

nezapisuje, ale čte. Kromě klonování tagu je používána také pro import, který je zprostředkován aktivitou `ActivityImport`. Tato třída po načtení dat pouze ověří, zda se jedná o data vytvořená touto aplikací, a poté vytvoří intent se zabalenou zprávou, který je odchyten aktivitou obsahující *intent-filter* pro akci `NFC_TASK_TAGS_IMPORT`. V této aplikaci je to aktivita `ActivityNewTag` z projektu aplikace, která načtená data zpracuje a vhodně zobrazí.

Jednou z nejdůležitějších tříd celé aplikace je aktivita `ActivityNfcRead`, která se stará o interpretaci naprogramovaného tagu. Všechny úkoly, které jsou tedy na tagu zapsány jsou na zařízení provedeny právě prostřednictvím této třídy. Postupně načítá z deserializovaného objektu třídy `MyTag` všechny `Action` a jednotlivě je vykonává.

Ovládání některých nastavení je velice jednoduché, většinou totiž existuje tzv. *manager*, pomocí kterého je nastavení možno měnit. Např. pro práci s Wi-Fi adaptérem existuje `WifiManager`, pro Bluetooth `BluetoothAdapter`, pro změnu vyzváněcího módu `AudioManager` a pro správu vyzváněcích tónů `RingtoneManager` a další. Pro některá nastavení ovšem takový správce neexistuje — v tom případě je nutno využít funkci `System.Settings.putInt(ContentResolver cr, String name, int value)`. Pro parametr `name` se používají konstanty ze stejné třídy vyjadřující konkrétní nastavení, které chceme měnit, parametr `value` je pak hodnota, která se do tohoto nastavení zapíše. Některá nastavení ovšem vůbec nenabízí správce ani metody, pomocí kterých je možné je měnit. Jediný způsob na jejich ovládání může být použití *Java reflexe* (viz *The Java Tutorials* [20]). Pomocí tohoto je možné přistoupit k API, které je skryté. Nevýhodou tohoto přístupu je, že není zaručeno, že v příští aktualizaci systému bude tento přístup zachován. Z tohoto důvodu je změna těchto nastavení v mé aplikaci spíše experimentální. Všechny zrealizované operace včetně způsobu implementace jsou zobrazeny v Tabulce 6.3.

Kromě změny nastavení může uživatel vykonávat různé další úkoly, například spustit přehrávač skladeb, otevřít URL nebo aplikaci apod. V těchto případech je vytvořen intent, který je pak dále zpracován aktivitou zvolené aplikace. Pokud chce tedy uživatel například otevřít webovou stránku, je použit jeho oblíbený internetový prohlížeč.

Trochu odlišná situace nastává při změně nastavení jasu displeje. Samotný zápis nové hodnoty totiž skutečný jas nezmění. Je potřeba spustit novou aktivitu bez uživatelského rozhraní, která bude spuštěna určitý minimální čas, a poté se sama ukončí. Tato aktivita je implementována třídou `ActivityBrightness`.

Poslední třídou knihovny je třída `Serializer` s metodou `serialize(...)` pro serializaci objektu na pole bytů a metodou `deserialize(...)` pro operaci opačnou.

Diagram nejdůležitějších tříd knihovny je zobrazen na Obrázku 6.6.

6.9 Zveřejnění aplikace

Jelikož považuji aplikaci za povedenou a myslím, že ji spousta lidí včetně mě může shledat užitečnou, rozhodl jsem se ji veřejně publikovat v obchodě *Google Play*. Pro zveřejnění bylo nutné si založit vývojářský účet u *Googlu*, což zahrnovalo vyplnění základních informací o sobě, odsouhlasení licenčních podmínek a zaplacení registračního poplatku \$25. Po splnění těchto kroků jsem získal přístup do internetové aplikace *Google Play Developer Console*, pomocí které je možné jednotlivé aplikace publikovat a spravovat.

Aplikaci bylo nutné před zveřejněním digitálně podepsat a exportovat jako balíček `.apk`, který byl poté nahrán do *Developer Console*, kde bylo dále nezbytné vyplnit název aplikace a její popis, logo a screenshoty. Tyto údaje je možné vložit ve více lokalizacích, v mém případě se jedná o češtinu a angličtinu. Dokonce je možné zvolit, ve kterých zemích a pro

Operace	Řešení	Operace	Řešení
Wi-Fi on/off	M	Připojit k Wi-Fi	M
Wi-Fi hotspot	R	Bluetooth on/off	M
Viditelnost Bluetooth	I	Mód letadlo	SN+I
Mobilní data on/off	R	Auto-synchro. on/off	M
Jas	SN+I	Auto-rotace	SN
Prodleva displeje	SN	Neuspávat při nabíjení	SN
Vyzváněcí profil	M	Tón vyzvánění	M
Hlasitost vyzvánění	M	Hlasitost budíku	M
Hlasitost médií	M	Vibrovat při vyzvánění	SN
Tón upozornění	M	Tón budíku	M
Poslat e-mail	I	Poslat SMS	M/I
Zavolat	I	Ovládání médií	I
Nastavit budík	I	Budík po časovači	I
Časové razítko	M/I	Otevřít aplikaci	I
Otevřít URL	I	Navigovat na	I
Zastavit procesy	M	Zrcadlo	I
Režim auto	M		

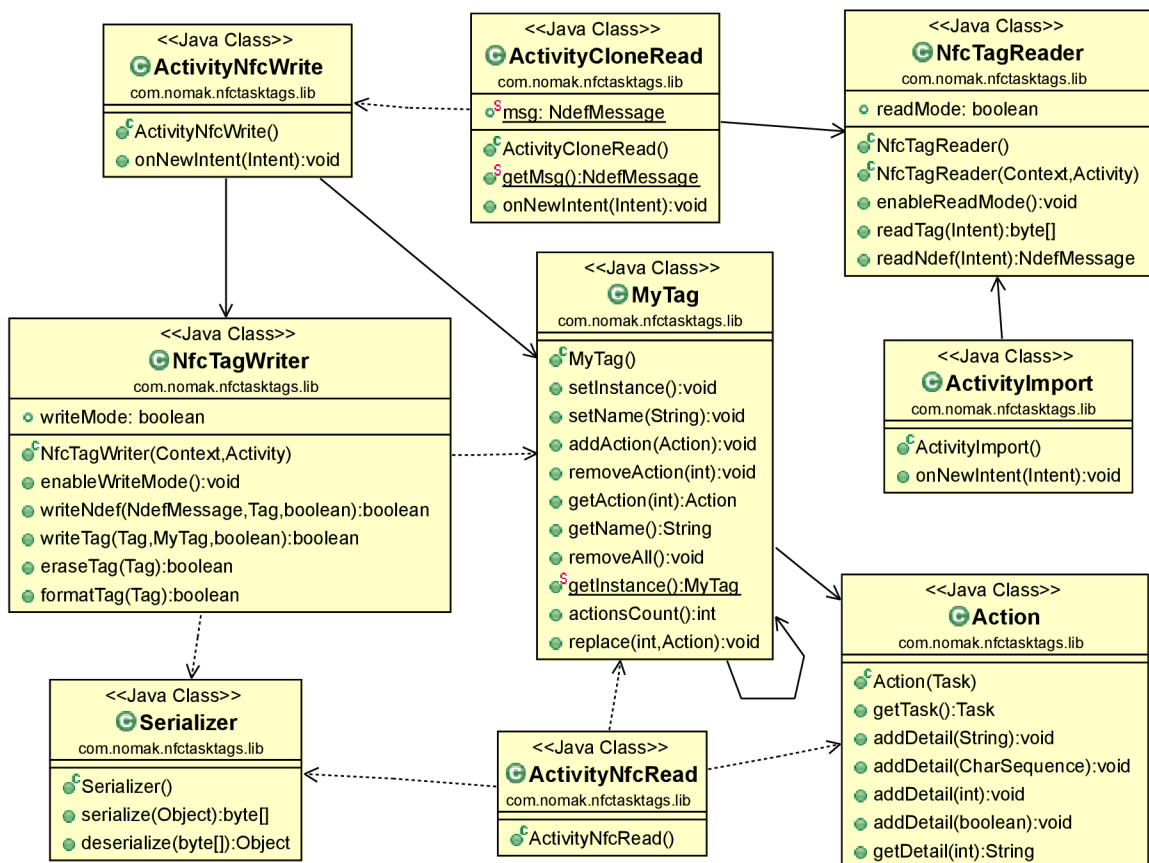
SN = Systémové nastavení M = Manager I = Intent R = Reflexe

Tabulka 6.3: Způsob realizace jednotlivých operací

jaké operátory bude aplikace dostupná a na jakých zařízeních si ji bude možné stáhnout. Má aplikace *NFC TaskTags* je v současné době⁴ dostupná ve všech zemích pro celkem 239 různých typů zařízení.

Pomocí *Google Play Developer Console* je možné sledovat počet stažení, recenze a hodnocení od uživatelů, případné chyby, které se vyskytnou a spoustu dalších informací. v případě aktualizace se pouze nahradí .apk soubor.

⁴stav aktuální k 18.4.2013



Obrázek 6.6: Diagram tříd knihovny

Kapitola 7

Testování a vyhodnocení aplikace

V této kapitole bude popsáno, jakým způsobem byla aplikace testována, následně bude rozebrán dotazník, který byl položen testovacím uživatelům a diskutovány dosažené výsledky.

7.1 Testování během vývoje

Během samotné realizace aplikace byly testovány všechny právě přidávané funkce. Na základě těchto testů byla každá objevená chyba okamžitě odstraněna. Toto testování probíhalo na reálném zařízení (viz podkapitola 6.1) a s pomocí debuggeru. Jako testovací NFC tagy byly použity plastové karty a přívěšek s čipy *Mifare Classic*.

Pro toto testování se ukázala jako velice přínosná aplikace *LogCat*, která sbírá a zobrazuje systémové logy a logy aplikací. Tato aplikace je součástí DDMS (*Dalvik Debug Monitor Server*), což je speciální debugovací prostředí pro *Android*, které je zahrnuto v ADT. Při implementaci často docházelo k chybám typu neošetřená výjimka. Díky této aplikaci bylo možné zobrazit *stack trace* a tak objevit, kde tyto výjimky vznikají.

7.2 Testování na uživateli

Po dokončení aplikace jsem se rozhodl ji vyzkoušet na patnácti potenciálních uživateli. Testování probíhalo tak, že jsem nejprve uživatelům vysvětlil smysl a princip aplikace, a poté je požádal o provedení jistých činností s aplikací. Po jejich provedení jsem je požádal o vyplnění krátkého dotazníku.

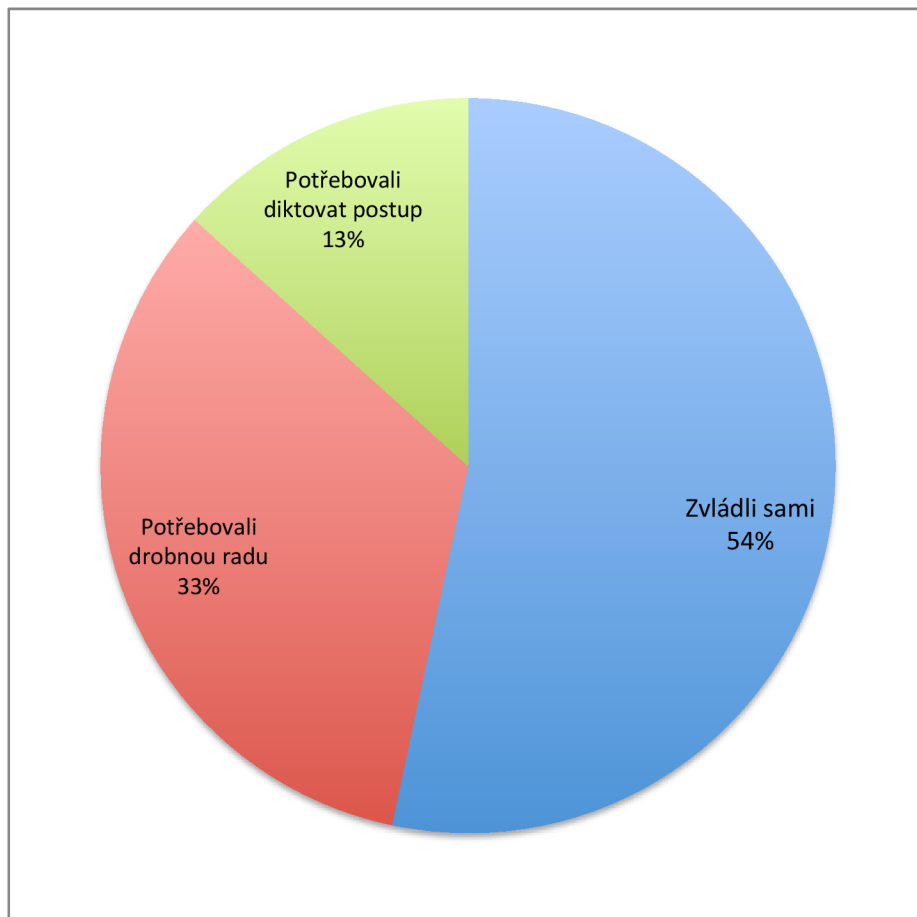
7.2.1 Sledování uživatelů při používání

Uživatelům byl zadán jednoduchý úkol, který by měl běžný uživatel s aplikací zvládnout. Během provádění úkolu jsem uživatele pozoroval, a pokud bylo potřeba, tak jsem jim poradil. Výsledky tohoto testování byly následně vyneseny do grafů.

První úkol byl vytvoření nového tagu se zadanými akcemi, kterými byly:

- zapnout Bluetooth
- změnit hlasitost vyzvánění
- poslat SMS se zadaným textem na zadané číslo

Po zvolení těchto akcí měly být všechny zapsány na poskytnutý NFC tag. Během tohoto testu se uživatelé většinou na začátku potřebovali zorientovat, a proto jim vložení první akce trvalo déle, ostatní však byly o poznání rychleji. Výsledky tohoto testu jsou zobrazeny v Obrázku 7.1.



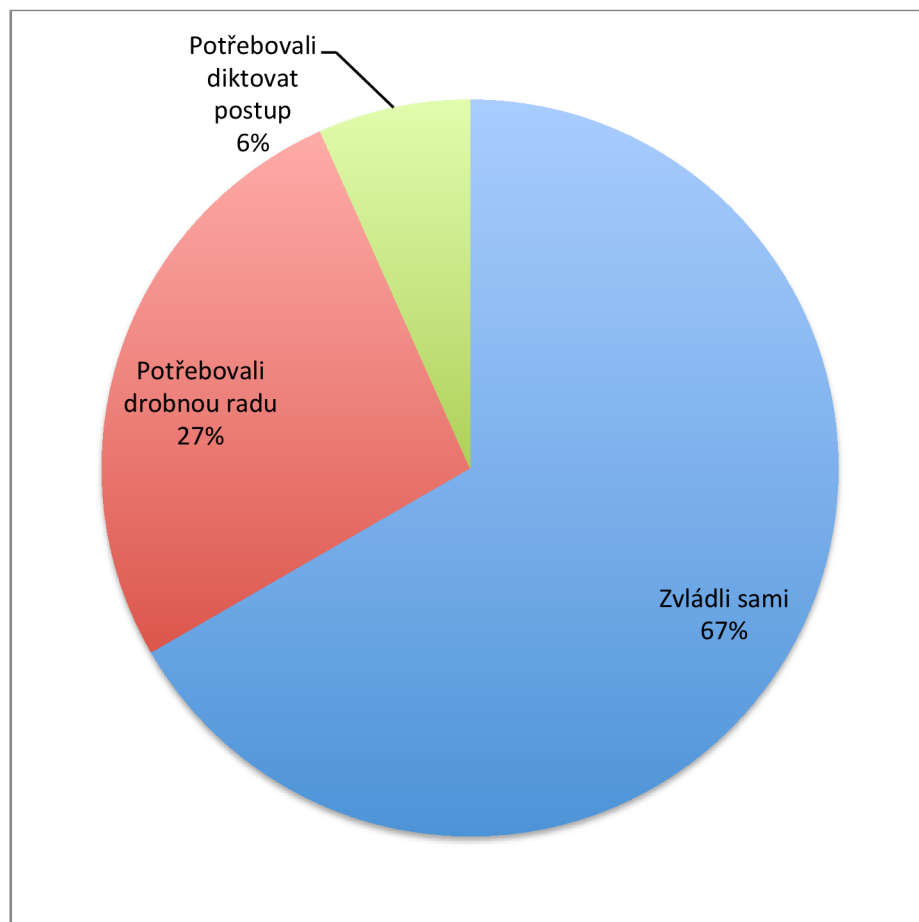
Obrázek 7.1: Graf výsledku testu 1

Následně byli uživatelé požádáni o naimportování tagu z minulého úkolu, kde měli upravit nebo nahradit položky tak, aby nový tag obsahoval tyto operace:

- vypnout Bluetooth
- poslat SMS na stejné číslo, ale s jiným textem než v prvním testu
- otevřít aplikaci *Facebook*

Takto upravené akce měly být opět zapsány na stejný NFC tag. Tento test zvládla většina uživatelů mnohem rychleji než test první. Výsledky jsou zobrazeny na Obrázku 7.2.

Z těchto výsledků se dá usoudit, že ovládání aplikace je skutečně poměrně intuitivní a dá se předpokládat, že se uživatel po krátké době s aplikací naučí plně pracovat.



Obrázek 7.2: Graf výsledku testu 2

7.2.2 Dotazník

Po prvních dvou testech byl této skupině uživatelů předložen krátký dotazník, kde měli jednotlivé položky ohodnotit známkou jako ve škole. Uživatelé byli dotazováni na intuitivnost ovládání aplikace, vzhled aplikace a její užitečnost. Výsledky tohoto dotazníku jsou zobrazeny na Obrázku 7.3.

Z těchto grafů je patrné, že by se mělo ještě zapracovat na způsobu ovládání aplikace, neboť se některým uživatelům nezdá dosti intuitivní, na druhou stranou jsou poměrně spokojeni se vzhledem. Co se týče užitečnosti, je jasné, že aplikace není pro každého a jen někteří ji reálně využijí.

7.3 Kapacita tagů

Pro zajímavost bylo experimentálně zjištěno, kolik úkolů se na jeden tag s čipem *Mifare Classic* s velikostí paměti 1 kB vejde. Byly vybrány namátkou pouze některé operace, které se na tagu opakovaly tolikrát, kolikrát to bylo možné. Počet jednotlivých akcí, které bylo možné na zvolený tag zapsat je zobrazen v Tabulce 7.1.

Z těchto hodnot je patrné, že na jeden běžný tag nelze takovýchto operací zapsat spoustu. Nejvíce místa zabírá *Změnit vyzváněcí tón*, protože se na tag zapisuje název tónu,

Úkol	Počet
Wi-Fi on/off	8
Připojit k zabezpečené Wi-Fi	5
Změnit vyzváněcí tón	4
Poslat SMS	234 znaků
Spustit aplikaci	4
Spustit zrcadlo	10

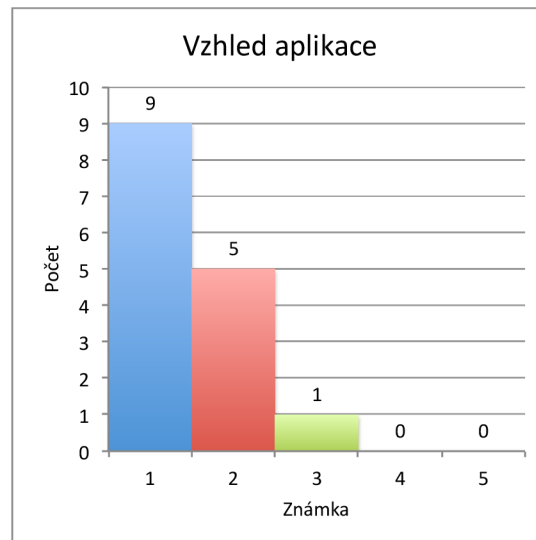
Tabulka 7.1: Kapacita tagu pro některé operace

který může být poměrně dlouhý, a pak *Spustit aplikaci*, kde se zapisuje název balíčku spouštěné aplikace. Nejméně místa pak zabírá například *Zrcadlo*, kdy se na tag zapisuje pouze typ akce bez dalších detailů, nebo *Wi-Fi on/off* (nebo jiná akce, která má hodnoty *Zapnout*, *Vypnout*, *Změnit stav*), kde se zapisuje pouze jedna hodnota.

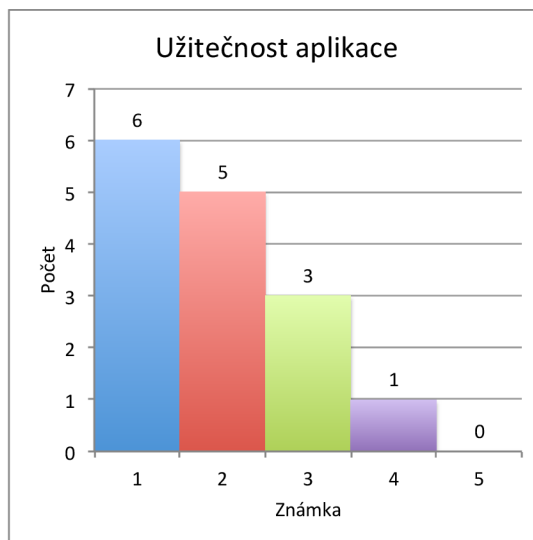
Při běžném používání aplikace však, dle mého názoru, stačí maximálně 4–5 operací na jeden tag, proto by tyto hodnoty měly být dostatečné. Pro použití většího množství úkolů je třeba zvolit tag s kapacitou alespoň 4 kB.



(a) Intuitivnost



(b) Vzhled



(c) Užitečnost

Obrázek 7.3: Grafy výsledků dotazníku

Kapitola 8

Závěr

Cílem této bakalářské práce bylo seznámit se s technologií NFC, jejím využitím v praxi a její podporou na platformě *Android*. Dále bylo úkolem navrhnout a implementovat systém pro tvorbu a využití vlastních NFC štítků.

Toto téma jsem si zvolil sám, protože se velice zajímám o mobilní telefony a hlavně o nové technologie. NFC je poměrně nový prostředek pro bezdrátovou komunikaci a v současné době je velice aktuální právě proto, že jej výrobci mobilních zařízení přidávají do svých produktů. Mým záměrem v této práci však bylo poukázat na to, že NFC je možné využít i jinak než je v širším povědomí — většina lidí totiž tuto technologii zná jen v souvislosti s bezkontaktními platbami telefonem, které se stávají stále modernější.

Nedá se říci, že by NFC bylo rychlejší nebo pracovalo na větší vzdálenosti než Wi-Fi, právě naopak — díky krátké vzdálenosti, na kterou pracuje se otevírá spousta nových možností využití. Jednou z nich je využití pasivních NFC zařízení — tagů pro zapsání dat, která po načtení aktivním zařízením — mobilním telefonem s NFC adaptérem, vyvolají v tomto zařízení určitou odezvu. Takové odezvy mohou být velice různorodé, od nastavení budíku, přes posláni SMS zprávy až po spuštění navigace na určité místo.

Právě takovou aplikaci, která dokáže data na NFC tagy zapisovat, a poté je i interpretovat, jsem vytvořil. Při jejím vytváření jsem zjistil, že podobné aplikace jsou již v obchodě *Google Play* dostupné (viz podkapitola 1.1), nicméně jsem si všiml, že spousta z nich trpí poměrně nepřehledným a nepřívětivým uživatelským rozhraním. Na základě tohoto zjištění jsem se rozhodl zaměřit právě na to, aby se ovládání aplikace stalo intuitivnější a přehlednější. Tento záměr se dle mého subjektivního názoru povedl a vypovídají o tom i výsledky testování (viz podkapitola 7.2). Dále jsem se pokusil o to, aby tagy mohly ovlivňovat co možná nejširší škálu nastavení a operací, které lze na telefonu provádět. Tento cíl jsem do jisté míry splnil také, i když některé původně zamýšlené operace nebylo možné implementovat (viz podkapitola 6.5). Aplikace byla rozdělena do dvou projektů, kde jeden zapouzdřuje všechny části týkající se uživatelského rozhraní a druhý obsahuje třídy pro práci s NFC. Tento druhý projekt je koncipován jako knihovna, je tedy možné jej použít i v jiných projektech.

Dle mého názoru se mi povedlo vytvořit aplikaci, která není jen součástí bakalářské práce, nýbrž je i reálně využitelná běžnými uživateli *Androidu*. z tohoto důvodu jsem aplikaci i veřejně publikoval v obchodě *Google Play* (viz podkapitola 6.9) a je tedy zdarma dostupná všem, kteří mají telefon s operačním systémem *Android* a adaptérem pro NFC.

Do budoucna mám v plánu do aplikace přidat nastavení, kde si uživatel bude moci zvolit vzhled aplikace, případně budou přidány zvuky a podobně. Také by bylo možné přidat lokální uložení pro vytvořené tagy, což by umožňovalo tvořit kopie, případně upravovat již

naprogramované tagy bez nutnosti je mít reálně u sebe. S dalšími verzemi *Androidu* plánuji přidat i potenciální nové operace a nastavení, které se mohou objevit. Dále popřemýšlím o úspornějším způsobu zápisu dat na tagy, aby mohly pojmout více akcí.

Kromě technologie NFC jsem se blíže seznámil s platformou *Android* a se způsobem vytváření aplikací pro tento systém. Mimo to jsem rozšířil své znalosti programovacího jazyka Java a značkovacího jazyku XML, které jsou v těchto aplikacích hojně využívány. Dále jsem zlepšil své schopnosti s grafickým editorem *Gimp*, který jsem využíval pro tvorbu plakátu, loga, ikon a další grafiky aplikace.

Celkově jsem tedy názoru, že jsem splnil celé zadání této bakalářské práce a udělal jsem i spoustu práce navíc, která vyústila ve vytvoření funkční a zajímavé aplikace, která může najít uplatnění i v širším měřítku.

Literatura

- [1] NFC Data Exchange Format (NDEF). [online]. 2006-07-24 [cit. 2013-04-15].
Dostupné z: http://www.nfc-forum.org/specs/spec_list/#ndefts.
- [2] Near Field Communication (NFC) Introduction and Software Development. *CNXSoft: Embedded Software Development* [online]. 2010-12-28 [cit. 2013-04-15].
Dostupné z:
<http://www.cnx-software.com/2010/12/28/near-field-communication-nfc/>.
- [3] NXP launches new generation of NFC tags with built-in security features. *NFC World* [online]. 2012 [cit. 2013-04-14]. Dostupné z:
<http://www.nfcworld.com/2012/10/31/320871/nxp-launches-new-generation-of-nfc-tags-with-built-in-security-features/>.
- [4] Android and iOS Combine for 91.1% of the Worldwide Smartphone OS Market in 4Q12 and 87.6% for the Year, According to IDC. *IDC: Analyze the Future* [online]. 2013-02-14 [cit. 2013-04-14]. Dostupné z:
<http://www.idc.com/getdoc.jsp?containerId=prUS23946013#.UWsz0b8soyl>.
- [5] Android SDK. *Android Developers* [online]. 2013 [cit. 2013-04-15]. Dostupné z:
<http://developer.android.com/sdk/index.html>.
- [6] Advanced NFC. *Android Developers* [online]. 2013 [cit. 2013-04-14]. Dostupné z:
<http://developer.android.com/guide/topics/connectivity/nfc/advanced-nfc.html>.
- [7] App Framework. *Android Developers* [online]. 2013 [cit. 2013-04-14]. Dostupné z:
<http://developer.android.com/about/versions/index.html>.
- [8] Application Fundamentals. *Android Developers* [online]. 2013 [cit. 2013-04-14].
Dostupné z:
<http://developer.android.com/guide/components/fundamentals.html>.
- [9] Dashboards. *Android Developers* [online]. 2013 [cit. 2013-04-14]. Dostupné z:
<http://developer.android.com/about/dashboards/index.html>.
- [10] Managing Projects. *Android Developers* [online]. 2013 [cit. 2013-04-14]. Dostupné z:
<http://developer.android.com/tools/projects/index.html#>.
- [11] Manifest.permission. *Android Developers* [online]. 2013 [cit. 2013-04-14]. Dostupné z:
<http://developer.android.com/reference/android/Manifest.permission.html>.
- [12] N-Mark. *NFC Forum* [online]. 2013 [cit. 2013-04-14]. Dostupné z:
<http://www.nfc-forum.org/resources/N-Mark/>.

- [13] Near Field Communication. *Android Developers* [online]. 2013 [cit. 2013-04-14]. Dostupné z:
<http://developer.android.com/guide/topics/connectivity/nfc/index.html>.
- [14] NFC Basics. *Android Developers* [online]. 2013 [cit. 2013-04-14]. Dostupné z:
<http://developer.android.com/guide/topics/connectivity/nfc/nfc.html>.
- [15] *NFC Forum* [online]. 2013 [cit. 2013-04-14]. Dostupné z:
<http://www.nfc-forum.org/home/>.
- [16] Themes. *Android Developers* [online]. 2013 [cit. 2013-04-14]. Dostupné z:
<http://developer.android.com/design/style/themes.html>.
- [17] About NFC. *NFC Forum* [online]. 2013 [cit. 2013-04-23]. Dostupné z:
<http://www.nfc-forum.org/aboutnfc/>.
- [18] Touch Devices. *Android Open Source Project* [online]. 2013 [cit. 2013-04-23]. Dostupné z:
<http://source.android.com/tech/input/touch-devices.html>.
- [19] UI Overview. *Android Developers* [online]. 2013 [cit. 2013-04-23]. Dostupné z:
<http://developer.android.com/design/get-started/ui-overview.html>.
- [20] Trail: The Reflection API. *The Java Tutorials* [online]. [cit. 2013-04-14]. Dostupné z:
<http://docs.oracle.com/javase/tutorial/reflect/>.
- [21] Doupal, F.: NFC - bezdrátová komunikace blízké budoucnosti? *NOTEBOOK.cz* [online]. 2011-8-17 [cit. 2013-04-15]. Dostupné z:
<http://notebook.cz/clanky/technologie/2011/nfc-bezdratova-komunikace-blizke-budoucnosti>.
- [22] Graddol, D.: *The Future of English? : Guide to Forecasting the Popularity of the English Language in the 21st Century*. London: The British Council, 1997, 64 s. ISBN 9780863553561.
- [23] Hrebenar, J.: Java: návrhový vzor Singleton. *Živě.cz* [online]. 2009-08-28 [cit. 2013-04-14]. Dostupné z:
<http://programovani.blog.zive.cz/2009/08/java-navrhovy-vzor-singleton/>.
- [24] Korb, K.: NFC tagy: co jsou vlastně zač a jak fungují. *near field* [online]. 2012-03-15 [cit. 2013-04-15]. Dostupné z:
<http://nearfield.cz/clanky/nfc-tagy-co-jsou-vlastne-zac-a-jak-funguji-5>.
- [25] Korb, K.: Základní kámen byl položen: operátoři se dohodli na Czech Wallet. *near field* [online]. 2012-09-19 [cit. 2013-04-15]. Dostupné z:
<http://nearfield.cz/clanky/zakladni-kamen-byl-polozen-operatori-se-dohodli-na-czech-wallet-69>.
- [26] Korb, K.; Pultzner, M.: Secure element: klíč k mobilním platbám. *near field* [online]. 2012-04-06 [cit. 2013-04-15]. Dostupné z:
<http://nearfield.cz/clanky/secure-element-klic-k-mobilnim-platbam-20>.
- [27] Kypta, T.: Vyvíjíme pro Android:Úvod. *Svět Androida* [online]. 2011 [cit. 2013-04-14]. Dostupné z:
<http://www.svetandroida.cz/vyvijime-pro-android-1-uvod-201103>.

- [28] Lee, W.-M.: *Beginning Android 4 Application Development*. Indianapolis: John Wiley & Sons, Inc, 2012, 560 s. ISBN 978-1-1181-9954-1.
- [29] Muna, R.: NFC Could Revolutionize Daily Consumer Activities. *IHS iSuppli* [online]. 2011-09-20 [cit. 2013-04-14]. Dostupné z:
<http://www.isuppli.com/Mobile-and-Wireless-Communications/MarketWatch/Pages/NFC-Could-Revolutionize-Daily-Consumer-Activities.aspx>.
- [30] Náprstek, M.: Co je technologie NFC (Near Field Communication). *nfctech.cz* [online]. 2011-11-09 [cit. 2013-04-15]. Dostupné z:
<http://www.nfctech.cz/co-je-near-field-communication-nfc/>.
- [31] Page, L.: Update from CEO. *Google: Official Blog* [online]. 2013-03-13 [cit. 2013-04-23]. Dostupné z:
<http://googleblog.blogspot.cz/2013/03/update-from-ceo.html>.
- [32] Petřek, P.: Vývoj pro Android: II. *Zdroják.cz* [online]. 2010 [cit. 2013-04-14]. Dostupné z: <http://www.zdrojak.cz/clanky/vyvoj-pro-android-ii/>.
- [33] Ujbányai, M.: *Programujeme pro Android*. První vydání. Praha 7: Grada Publishing a.s., 2012. ISBN 978-80-247-3995-3.
- [34] Womack, B.: Google Says 700,000 Applications Available for Android. *BloombergBusinessweek* [online]. 2012-10-29 [cit. 2013-04-23]. Dostupné z:
<http://www.businessweek.com/news/2012-10-29/google-says-700-000-applications-available-for-android-devices>.

Příloha A

Obsah CD

Aplikace/ složka obsahující soubory aplikace

NFCTaskTags.apk soubor aplikace, který lze přímo nainstalovat na zařízení

NFCTaskTags/ soubory projektu aplikace pro *Eclipse*

NFCTaskTagsLib/ soubory projektu knihovny pro *Eclipse*

Demonstrační video/ složka obsahující soubory videa

Demonstrační video.mp4 soubor s demonstračním videem

Plakát/ složka obsahující soubory plakátu

Plakát.xcf zdrojový soubor plakátu pro *Gimp*

Plakát.png rasterizovaný plakát

Technická zpráva/ složka obsahující soubory technické zprávy

Technická zpráva.pdf soubor s touto technickou zprávou

Technická zpráva/ zdrojové soubory technické zprávy pro L^AT_EX

Příloha B

Plakát

SAMSUNG

Převrat v ovládání telefonů právě přichází ...

Otočný číselník → Tlačítka → Stylus → Dotyková gesta

NFC TaskTags

Nfc TaskTags je přelomová nová aplikace pro Android využívající NFC. Pomocí ní si můžete definovat operace, od nastavení budíku, přes zaslání SMS až po spuštění navigace na určité místo, které následně zapíšete na NFC tag. Pro provedení těchto akcí pak stačí telefon k tomuto tagu přiblížit bez nutnosti Vašeho dalšího zásahu.

Jedná se tedy o **BEZDOTYKOVÉ OVLÁDÁNÍ**.

Vlastnosti:

- Podpora Android 4.0 a novější
- Česká a anglická lokalizace
- 33 různých akcí telefonu
- Správa NFC tagů
- Čistý a moderní vzhled
- Intuitivní ovládání
- A další ...

Ke stažení na Google Play