

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Komunikace a identifikace informačních majáků

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 14. května 2013

Bc. Jiří Hromádka

Děkuji vedoucímu práce Ing. Ladislavu Pešíčkovi za odborné vedení a cenné rady při vypracování diplomové práce.

Abstract

This diploma thesis deals with communication and identification of information beacons. Information beacon is either visible or invisible element (or system of elements), which can be detected by smartphone or tablet, and information content, can be read from it. In this thesis there is also designed the usage of various technologies and methods that can be used to identify information beacons around user, both inside buildings and in outdoor areas.

The result of this thesis is a mobile application for the Android platform that communicates with prepared system of information beacons and provides the user with information content based on his location or effort to obtain information

Obsah

1	Úvod	1
1.1	Informační majáky	1
1.2	Cíl práce	1
2	Existující systémy informačních majáků	2
2.1	Význam informačních majáků	2
2.2	Google Maps	4
2.2.1	Výhody a nevýhody	5
2.3	Google Indoor Maps	5
2.3.1	Výhody a nevýhody	6
2.4	RedPin	7
2.4.1	Algoritmus určení polohy	8
2.4.2	Výhody a nevýhody	9
2.5	TripAdvisor	9
2.5.1	Výhody a nevýhody	10
2.6	ByteLight	11
2.6.1	Výhody a nevýhody	11
2.7	Systémy pro konkrétní instituce	12
2.7.1	Výhody a nevýhody	12
2.7.2	iTAG.cz	12
3	Možnosti identifikace informačních majáků	14
3.1	Pojem Indoor positioning	14
3.2	GPS	14
3.3	QR kód	15
3.3.1	Jak přečíst QR kód	15
3.3.2	Dostupné knihovny	17
3.4	WiFi	18
3.4.1	WiFi otisky	19
3.4.2	WiFi triangulace	20
3.4.3	Omezení mobilních platforem	20

3.4.4	WiMax	21
3.5	Bluetooth	21
3.5.1	Využití	22
3.5.2	Omezení mobilních platforem	22
3.6	Near Field Communication	23
3.6.1	Typy tagů, komunikace	24
3.6.2	Využití	25
3.6.3	Zabezpečení vs. interaktivita	26
3.7	Locata	26
3.7.1	Využití	26
3.8	Rozpoznání obrazu scény	27
3.9	LED diody	28
3.9.1	Využití	28
3.10	Zvuk	29
3.10.1	Využití	29
3.11	Další metody	30
3.11.1	Metody závislé na znalostech	30
3.11.2	Metoda založená na všímavosti	31
3.12	Odhalení podvodů	31
3.12.1	Spojení s GPS polohou	32
3.13	Přínos pro instituci	32
4	Platforma Android	33
4.1	Novinky	33
4.1.1	Project Butter	33
4.1.2	Vložené fragmenty	34
4.1.3	Action Bar	34
4.1.4	API podpory různých obrazovek	35
4.2	Přehled verzí	35
4.2.1	Support Library	36
4.3	Fragmenty	36
4.3.1	Komunikace s aktivitou	39
4.4	Vhodnost pro realizaci klienta	40
5	Mobilní klient pro platformu Android	41
5.1	Popis vytvořeného systému Tagiee	41
5.2	Návrh aplikace	42
5.2.1	Funkce aplikace	42
5.2.2	Návrh UI	45
5.2.3	Komunikace se systémem Tagiee	45
5.2.4	Jedinečná identifikace zařízení	47

5.2.5	Algoritmus hledání polohy	48
5.2.6	Odůvodnění oprávnění aplikace	51
5.3	Google apis	53
5.3.1	Google Maps Android API v2	54
5.3.2	YouTube Data API v3	55
5.4	Použití NFC	56
5.4.1	The Tag Dispatch System	57
5.4.2	Foreground Dispatch System	58
5.4.3	NDEF formát	59
5.5	Implementace QR čtečky	59
5.6	Databáze	60
5.7	Verze pro tablety	61
5.7.1	Podpora různých obrazovek	62
5.8	Tagiee Admin	64
5.8.1	Funkce aplikace	64
5.8.2	Příprava NFC tagu	66
5.9	Ukázky aplikací	67
5.9.1	Tagiee	68
5.9.2	Tagiee Admin	69
6	Ověření funkčnosti	70
6.1	Modelová situace - DOD areál ZČU Bory	70
6.2	Ověření funkčnosti - Areál ZČU Bory	70
6.2.1	Výsledky ověření	72
7	Závěr	75
A	Přílohy	i
A.1	Struktura CD	i
A.2	Uživatelská příručka	ii
A.2.1	Instalace aplikace	ii
A.2.2	Používání aplikace	ii
A.3	Seznam JSON požadavků a odpovědí	iv
A.3.1	add_review	iv
A.3.2	beacon_found	iv
A.3.3	device_visit_institution	v
A.3.4	get_institution_by_uid	v
A.3.5	get_institution_list	vii
A.3.6	get_institution_revision	vii
A.3.7	get_reviews	viii

1 Úvod

1.1 Informační majáky

Informační maják není ve větší míře zavedený pojem a proto následuje jeho definice pro tuto práci. Informační maják představuje jakýkoliv at' už viditelný, nebo neviditelný fyzický prvek (či soustavu fyzických prvků), který lze prostřednictvím mobilního „chytrého“ zařízení detekovat a přečíst v něm uložené informace nebo na základě identifikace informačního majáku si o příslušné informace automaticky zažádat např. na serveru, který může být umístěn v síti Internet.

V současné době jsou mobilní „chytrá“ zařízení (tzv. smartphone) celosvětově rozšířená a vytváří jakýsi pomyslný most mezi dříve oddělenými obory. Umožňují zvýšit míru interakce, aktuálnost informací a dávají možnost nabídnout uživateli vše, co by mohl v daném okamžiku potřebovat a chtít. Z těchto důvodů je ideální využít tato „chytrá“ zařízení pro systém informačních majáků. Tento fakt navíc podporuje skutečnost, že čím dál tím větší procento mobilních zařízení má dostupné aktivní připojení k síti Internet.

Škála využití systémů informačních majáků je velice pestrá. Některé již zavedené systémy jsou popsány v kapitole 2, dále je pak v kapitole 6 názorný modelový příklad pro vytvořený systém informačních majáků. Možnosti, jak lze informační majáky prostřednictvím mobilních zařízení detekovat a případně z nich číst informace, jsou popsány v kapitole 3.

1.2 Cíl práce

Cílem této diplomové práce je prozkoumat trh a zjistit, jaké systémy podobného typu existují a jaké možnosti identifikace informačních majáků využívají. Dále budou prozkoumány další možnosti identifikace informačních majáků. Na základě zjištěných skutečností bude vybrána množina technologií vhodných pro realizaci systému informačních majáků a bude vytvořena mobilní aplikace pro platformu Android, která by měla uživateli pohodlně nabídnout aktuální obsah v závislosti na jeho poloze, či snaze obsah získat.

2 Existující systémy informačních majáků

Obecně se všechny systémy informačních majáků dají rozdělit pro použití ve vnitřních prostorech (tzv. Indoor systémy), na použití v prostorách vnějších (Outdoor systémy) či jejich kombinaci. Typ systému omezuje použití některých technologií pro identifikaci majáků, jelikož například uvnitř budovy nebývá možné zachytit GPS signál a porovnávat tak polohu uživatele a majáku.

2.1 Význam informačních majáků

Význam informačních majáků se může lišit v závislosti na instituci, ve které jsou použity, a na účelu, ke kterému jsou využívány.

Použití ve vzdělávacích institucích

Majáky se dají použít ve vzdělávacích institucích, jako je například areál univerzity, ve kterém je těžké se při první návštěvě orientovat, a mohou novému členovi pomoci najít místnosti, kde probíhá jeho výuka či kanceláře, do kterých se má dostavit. Konkrétní příklad použití ve vzdělávací instituci je uveden v kapitole 6.

Použití v podniku

Dalším příkladem může být větší podnik, kde noví zaměstnanci neznají firemní procesy a politiku a využitím informačních majáků se mohou rychleji zařadit do pracovního procesu. Informační majáky mu mohou například sdělit, které všechny dokumenty musí vyplnit, které lidi musí navštívit před tím, než půjde za vedoucím s určitým požadavkem apod. Informační majáky nemusí sloužit pouze ke sdělování informací. Lze je použít například i pro komunikaci s firemním systémem. Předchozí příklad, lze tedy rozšířit. Nový zaměstnanec přímo z informačního majáku získá dokumenty, které musí vyplnit, a skrze

příslušný maják je také vyplněné odevzdá. Tím se výrazně zkrátí čas celého procesu, nemusí se také zbytečně plýtvat papírem apod. Z pohledu zaměstnavatele jsou informační majáky prostředkem, jak umožnit zaměstnancům pracovat efektivněji a rychleji.

Použití ve veřejných institucích

Použití informačních majáků se také nabízí ve veřejných institucích jako jsou muzea, výstaviště, zoologické zahrady, botanické zahrady apod. Skrze informační majáky je možné návštěvníky informovat o aktuálně probíhajících soutěžích a případně skrze majáky umožnit návštěvníkům soutěžit. Soutěž může probíhat jako prosté odpovídání na otázky, či nutnost navštívení všech majáků v dané instituci. Dále je možné návštěvníkům nabízet multimediální obsah související s exponátem / stánkem, před kterým se aktuálně nacházejí.

Obecně lze říci, že zavedením informačních majáků v jakékoliv instituci má provozovatel možnost zvýšit svou atraktivitu, zlepšit informovanost návštěvníků, zvýšit interakci s návštěvníky a případně podle statistik upravit vystavovaný obsah. V neposlední řadě mohou informační majáky snížit náklady na popis exponátů (popis a doprovodné multimedia si návštěvník zobrazí na svém zařízení a provozovatel tak nemusí investovat např. do zobrazovacích zařízení, případně při jejich absenci neochudí návštěvníka o multimediální zážitek).

Komerční použití

V čistě komerčním prostředí, jako jsou nákupní centra, lze informačních majáků využít k nabízení produktů na základě toho, před kterým obchodem se zákazník právě nachází. Běžnému uživateli se tento způsob nabízení zboží může zdát jako nucená a nechtěná reklama a systém nemusí chtít využívat. Ovšem pokud si zákazník zboží zakoupí na základě nabídky informačního majáku, může mu být nabídnuta sleva a tím provozovatel „donutí“ zákazníka v jeho nákupním centru systém informačních majáků využívat.

Tím, že bude zákazník hledat majáky (potažmo slevy) po nákupním centru, se zvětšuje šance, že si s sebou odnese i zboží, pro které vůbec nešel. Lze tedy bez obav říci, že informační majáky jsou vhodné i pro čistě komerční účely.

2.2 Google Maps

Google Maps samy o sobě nejsou přímo systémem informačních majáků. Jedná se o pouhé zjišťování polohy na základě signálu GPS, popř. je poloha určena podle základových stanic operátora (BTS), které mobilní zařízení detekuje, dle BTS, k níž je zařízení připojeno, či na základě okolních WiFi zařízení, jejichž více či méně přesnou polohu má Google zaznamenanou. Jedná se o Outdoorový polohovací systém¹.

Přidanou hodnotou mohou být tzv. markery, což jsou virtuální body, které se do mapy dají přidat. Jednotlivé markery lze pojmenovat a jako jejich obsah do nich lze uložit libovolná textová informace. Vložený text může být navíc uložen jako HTML kód, tudíž jde vytvořit téměř libovolný obsah majáku (text, obrázky, hypertextových odkazy apod.). Lze vložit i prvek *iframe*², čímž se otevírá cesta pro vložení externího obsahu, jako jsou například vložená tzv. *embedded* videa. Google nicméně myslí na bezpečnost, tudíž nedovolí vložit např. JavaScript kód, nebo *iframe*, který obsahuje pro Google nepověřenou stránku (při testech se povedlo vložit pouze *embedded* YouTube video). Takto vybavené body už lze považovat za informační majáky.

Již zmíněné markery nejsou výsadou pouze map od společnosti Google. Podobnou funkcionalitu nabízí většina poskytovatelů map. Z českých poskytovatelů je to například Seznam.cz a jeho mapy, u kterých ovšem není možné uložit více bodů s jakoukoliv přidanou informací (ať už se jedná o pojmenování nebo obsah). Pojmenovaný a obsahově naplněný bod lze uložit a sdílet maximálně jeden, což je pro systém informačních majáků nedostačující.

Google Maps je vcelku dobře použitelný systém informačních majáků pro uzavřenou skupinu, jako jsou například zaměstnanci jedné organizace. Nehodí se ovšem pro veřejné instituce. Viz níže - nevýhody.

¹Systém zabývající se určením polohy ve venkovních prostorech, mimo budovy.

²HTML prvek, který v sobě umí zobrazit jiný HTML dokument

2.2.1 Výhody a nevýhody

Výhody

Nespornou výhodou Google Maps je jejich rozšířenost. Mapy od společnosti Google jsou hojně využívány a každé zařízení postavené na platformě Android³ má klientskou aplikaci pro Google Maps, kde si může majáky zobrazovat.

Další výhodou je fakt, že služba je bezplatná a majáky se dají pohodlně nastavit skrze webové rozhraní.

Nevýhody

Jistou nevýhodou tohoto řešení je fakt, že vytvořené markery jsou privátní, tzn. vidí je pouze ten uživatel, který je vytvořil. Je zde určitá možnost takto vytvořenou mapu včetně majáků sdílet a to buď pomocí emailových pozvánek nebo URI.

Markery je samozřejmě možné vytvořit jako veřejné, ty se ovšem automaticky nezobrazí na mapě. Zobrazí se pouze jako výsledek hledání konkrétního dotazu. Z tohoto důvodu jsou takto vytvořené veřejné mapy nepoužitelné.

2.3 Google Indoor Maps

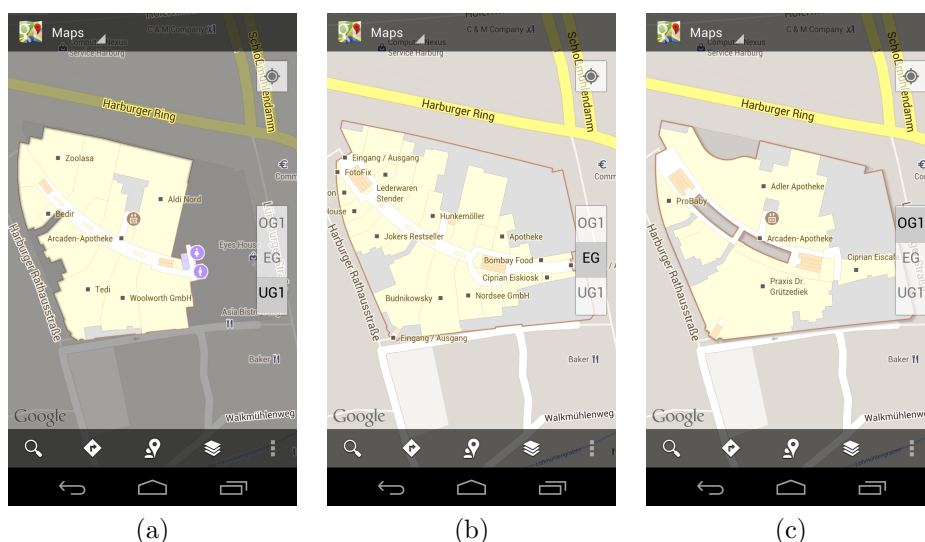
Google Indoor Maps je, jak už název napovídá, kombinace Google Maps a navigace uvnitř budov. Jedná se o rozšíření Google Maps, které umí zobrazit vnitřní plán budovy a uživatelskou pozici a to včetně jednotlivých podlaží. Indoor Maps se používají nejčastěji v obchodních centrech, letištích, muzeích, stadionech a vlakových nádražích.

Projekt sám o sobě oproti Google mapám nepřináší žádné vylepšení, co se určování polohy týče. Stejně jako Google mapy využívá GPS (uvnitř budovy

³nemusí platit pro tzv. *rootnutá* zařízení a upravené systémy - takovýto zařízení není ovšem významné procento a mívají je zkušební uživatelé, kteří si klientskou aplikaci umí v případě potřeby doinstalovat

velmi zřídka), zachytává WiFi a Bluetooth signály a určuje polohu pomocí triangulace. Polohy vysílajících WiFi a Bluetooth zařízení jsou zachyceny a následně upravovány podle informací z mobilních zařízení, která se dostala do dosahu vysílání těchto zařízení a mají povoleno odesílat svoji polohu a další informace zpět společnosti Google. Z tohoto důvodu není zjišťování polohy uživatele tak přesné, jako tomu je u GPS, jelikož ani poloha vysílajících zařízení není úplně přesně určena.

K tomu, aby byla navigace uvnitř budovy dostupná, je ovšem nutné společnosti Google poskytnout plány jednotlivých pater budovy. Na začátku roku 2013 je takto zmapováno 10 000+ budov z 13 zemí. V České republice není takto zmapována ani jedna budova. [ind()] Ukázka z nativní aplikace je vidět na obrázku 2.1. Na pravé straně obrazovky je vidět aktivní přepínač pro jednotlivá patra budovy.



Obrázek 2.1: Zobrazení plánů jednotlivých pater nákupního střediska Arcaden, Hamburg, Německo

2.3.1 Výhody a nevýhody

Výhody

Výhodou je možnost využití další ze služeb společnosti Google - Latitude, která slouží ke sdílení polohy uživatele. Dále je to také kvalita plánů, jelikož

pro vložení plánu pater budovy je nutné dodržet minimální požadavky.⁴

Nevýhody

Nevýhodou je fakt, že nativní klient pro Indoor Maps je pouze na platformě Android. Ostatní platformy (iOS, Windows Phone) musí použít webové rozhraní.

Dalším negativem je nemožnost využívání *markerů*, jako tomu bylo i Google Maps, pokud se jedná o vícepatrové budovy. Markerům se totiž nedá nastavit, pro které podlaží se mají zobrazovat. V případě Indoor Maps jsou markery jednotlivé místnosti v budově (např. obchody v nákupním centru). Markery se tedy nedají dynamicky přidávat a odebírat například na základě aktuálně probíhajících akcí apod.

2.4 RedPin

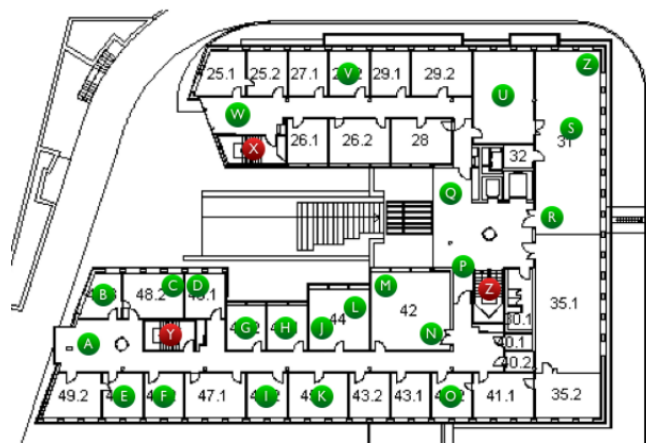
RedPin⁵ je čistě Indoor polohovací systém, který se snaží určit uživateli polohu s přesností na místnost. Pro určení polohy využívá zachycené WiFi signály (BSSID, včetně RSSI), Bluetooth a GSM signály od okolních zařízení.

Zvláštností systému je workflow určení mapy vnitřního prostoru. Systém má v sobě na začátku pouze plány budovy rozčleněné na jednotlivá patra a uživatelská aplikace dostane pouze tento plán. Celý systém tak neprochází na začátku existence fází „trénování“, ale spoléhá na komunitu využívající systém, že trénování provedou v průběhu fungování. Tento princip vychází z myšlení autorů, kteří počítají s tím, že se okolí neustále mění - jednotlivé WiFi přístupové body se nahrazují za nové, někdy nevysílají atd., a tak se celý systém může neustále adaptovat.

Workflow trénování a fungování systému je následující. Po spuštění odesílá aplikace na server nalezené vysílající WiFi zařízení (BSSID, RSSI) a server se na základě již uložených „otisků“ poloh snaží určit uživateli polohu. Pokud server není schopný polohu určit, je uživatel vyzván, aby zvolil patro a iden-

⁴dostupné na <http://support.google.com/gmm/bin/answer.py?hl=en&answer=1711534&topic=1685871&ctx=topic>

⁵domovská stránka <http://www.redpin.org/>



Obrázek 2.2: Ukázka natrénovaného systému. Zelené body určují polohu uložených „otisků“.[Bolliger()]

tifikátor místnosti, ve které zrovna je, případně ji určil přímo na mapě. Tím dochází k trénování systému. Díky tomuto kontinuálnímu trénování, je systém RedPin schopný se adaptovat na změny v okolním elektromagnetickém prostředí. Natrénovaný systém je vidět na obrázku 2.2.

Systém RedPin je open-source a na svých stránkách nabízí ke stažení jak serverovou část, která je napsaná v programovacím jazyce Java, tak i mobilní klienty. Jedná se o multiplatformní systém, protože mobilní klienti jsou dostupní pro platformu Android, iOS, dokonce i pro Symbian. Klient pro Windows Phone v současné době není dostupný.

2.4.1 Algoritmus určení polohy

Jak bylo zmíněno výše, poloha je určována podle unikátních identifikátorů vysílajícího zařízení a síly signálu. Uvedeno na příkladu, algoritmus funguje následovně. Mějme dvě sousedící místnosti A a B. V systému je uložen „otisk“ obou místností. V místnosti A je umístěn WiFi přístupový bod, který vysílá. V případě, že se uživatel nachází v místnosti B, zachytává jeho zařízení signál stejně tak, jako by byl v místnosti A. V případě této shody, server provede určení polohy znovu s přihlédnutím na rozdíl síly změřeného signálu se silou signálu uloženou v otisku místností. Čím menší bude rozdíl, tím větší „bonus“ bude přičten k jednotlivé poloze. Nejlepší poloha bude vrácena a zobrazena uživateli.

2.4.2 Výhody a nevýhody

Výhody

Výhodou celého projektu je jeho otevřenost. Každý zájemce si může systém stáhnout, upravit pro své potřeby a nasadit do provozu.

Dalším pozitivem je deklarovaná přesnost, kde systém by měl být schopný poskytnout určení polohy s přesností na místnost.

Nevýhody

Z pohledu nasazení ve veřejné instituci je určitou nevýhodou workflow trénování systému. Z počátku prázdný systém není návštěvníkovi veřejné instituce k užítku a žádost o identifikaci místnosti v případě nenalezení polohy návštěvníka je nežádoucí. Ačkoliv je systém open-source (což dává možnost chování upravit dle požadavků), odstranit výše zmíněné nežádoucí vlastnosti, by znamenalo odstranit podstatu projektu.

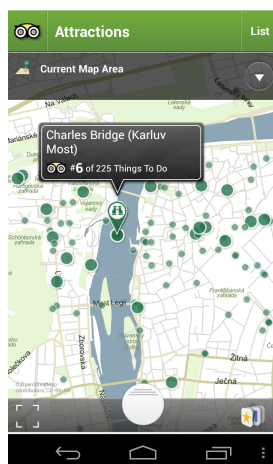
Dalším negativem je fakt, že klient pro platformu iOS nemůže být distribuován prostřednictvím portálu AppStore. Klient totiž využívá privátní API pro zjištění BSSID WiFi zařízení, což se neshoduje s podmínkami pro uveřejnění aplikace na portálu AppStore.

2.5 TripAdvisor

TripAdvisor⁶ je multiplatformní Outdoor systém, který lze využívat jak z mobilních aplikací, tak z webového prohlížeče. Tento systém slouží, jak už název napovídá, jako rádce při plánování výletů. Mobilní aplikace slouží jako průvodce po významných světových městech, jako je například Amsterdam, Praha, Řím, New York atd.

Pro každé město je samostatná aplikace a samostatná databáze nákupních center, restaurací, bankomatů, zastávek městské dopravy, turistických atrakcí

⁶domovská stránka <http://www.tripadvisor.com/>



Obrázek 2.3: Ukázka aplikace TripAdvisor pro hlavní město Praha - seznam turistických atrakcí.

apod. Aplikace dále umožňuje stáhnout databázi obsahu do zařízení, tudíž je uživatel i v zahraničí bez připojení k Internetu schopný aplikaci využívat.

Aplikace využívá pouze GPS a GSM signál k určení polohy uživatele a na základě jeho polohy doporučuje nejbližší turistické vyžití v daném městě.

2.5.1 Výhody a nevýhody

Výhody

Velkou výhodou jsou velmi podrobná data a jejich možnost stažení do vnitřního úložiště zařízení, což dává možnost systém využívat i v zahraničí, kde není připojení k Internetu pro cizince levnou záležitostí.

Nevýhody

Tento systém nelze nijak využít pro vlastní potřeby z pohledu provozovatele systému informačních majáků - nelze přidávat své majáky s vlastním obsahem.

2.6 ByteLight

Systém ByteLight⁷ je pouze *Indoor* lokalizační systém, který je jedinečný v určování polohy uvnitř budov - využívá světlo. K tomu je ještě zapotřebí speciálních LED žárovek, které jsou schopné blikat určitou frekvencí. Frekvence se pohybuje řádově ve stovkách Hz, takže je pro lidské oko nezachytitelná. Kamera mobilního zařízení je ovšem schopna toto blikání zachytit a zjistit jeho frekvenci.

Frekvence blikání je pro ByteLight to samé, jako je MAC adresa pro síťové rozhraní - jedinečně určuje žárovku. Na základě znalosti umístění žárovky, lze poměrně přesně zjistit i polohu návštěvníka.

V současné době má společnost ByteLight patentované své LED žárovky s klasickým závitem, tudíž pro používání není třeba budovat novou infrastrukturu, ale stačí pouze vyměnit žárovky.

2.6.1 Výhody a nevýhody

Výhody

Výhodou tohoto systému je jeho relativně vysoká přesnost při určení polohy uvnitř budovy. Další výhodou je API a vzorové aplikace, které ByteLight nabízí, takže lze tuto technologii využít téměř jakýmkoliv zamýšleným způsobem.

Nevýhody

Značnou nevýhodou je cena LED žárovek, kde jedna žárovka se pohybuje okolo 2500,- Kč za kus. Stávající žárovky se nedají vylepšit o ByteLight technologii, ale musí se pořídit nové. Pro společnost, která již investovala do úsporného LED osvětlení, může být zavedení ByteLight systému velmi nákladná záležitost. Žárovky jsou navíc v současné době vyráběny pouze pro napětí 120 V, tudíž pro státy, kde je napětí v síti větší je systém nepoužitelný. [byt()]

⁷domovská stránka <http://www.bytelight.com/>

2.7 Systémy pro konkrétní instituce

Předchozí probírané systémy byly do jisté míry univerzální. Odlišné byly nejčastěji způsobem, jakým zjišťovaly polohu uživatele a co vše na základě jeho polohy bylo možné uživateli poskytnout.

Mezi další příklady patří systémy, které jsou vytvořeny přímo pro konkrétní instituci. Nejčastěji se jedná o muzea, zoologické a botanické zahrady, výstaviště nebo městskou hromadnou dopravu konkrétního města.

Jejich společným jmenovatelem je zjišťování polohy uživatele, které se děje vesměs vždy využitím signálu GPS nebo je uživatelova poloha určena přibližně v rádech desítek metrů pomocí GSM signálu. Velmi zřídka je využita i jiná technologie, jako je například NFC (více o NFC v části 3.6) či QR kódy.

Aplikace uživatelům nabízí více či méně interaktivní obsah. Zpravidla obsahují mapu významných bodů dané instituce nebo pouze statickou mapu s vyznačenou cestou, kudy by se měl návštěvník vydat. Zřídka tyto aplikace umožňují zaslat instituci zpětnou vazbu v podobě emailové zprávy nebo přidat hodnocení celé instituce, popř. jednotlivých exponátů. Další přidanou hodnotou pro návštěvníka jsou doprovodná multimédia (fotografie, videa atd.), které má aplikace uložena v sobě.

2.7.1 Výhody a nevýhody

Výhodou takovýchto systémů je skutečnost, že jsou dělané na míru konkrétní instituci či situaci a je tedy předpoklad, že pro ni budou fungovat spolehlivě. Avšak z toho plyne v podstatě jejich nevýhoda a to, že nejsou obecné a nijak dále použitelné.

2.7.2 iTAG.cz

Takovým systémem, který je tvořen na míru je například **iTAG.cz**, což je systém informačních majáků pro Plzeňské městské dopravní podniky.

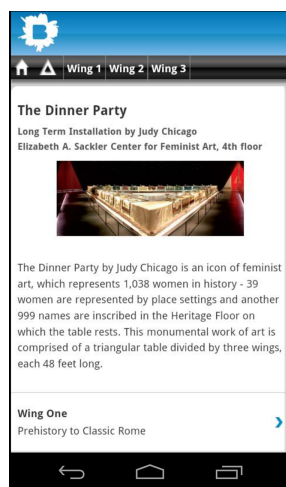
Na každé zastávce je pod klasickým papírovým jízdním řádem umístěn v malé průhledné krabičce QR kód společně s NFC tagem. Po přečtení at'

už QR kódu nebo NFC tagu, je mobilní zařízení přeměřováno na webovou stránku s aktuálním jízdním řádem pro konkrétní zastávku a směr.

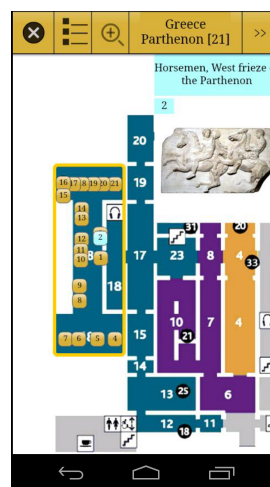
Pro tento systém není třeba mít v zařízení nainstalovaného klienta, stačí mít nainstalovanou libovolnou QR čtečku. To se může jevit jako výhoda, avšak v případě, že zařízení nemá připojení k Internetu, neexistuje možnost jak „obsah majáku“ získat.



(a) Zoologická zahrada Brno



(b) Muzeum v Brooklynu



(c) British Museum

Obrázek 2.4: Ukázka klientů systému informačních majáků pro konkrétní instituce realizované na platformě Android

3 Možnosti identifikace informačních majáků

V této kapitole budou rozebrány vybrané technologie, které by se daly využít pro identifikaci informačních majáků jak uvnitř tak mimo budovy a které by bylo vhodné použít při realizaci mobilního klienta informačních majáků na platformě Android.

3.1 Pojem Indoor positioning

V předchozích kapitolách byl problém určování polohy uvnitř budovy - tzv. Indoor positioning - několikrát zmíněn. V současné době je určení polohy mobilních zařízení až na jednotky metrů mimo budovy (pomocí GPS) běžnou a samozřejmou záležitostí, nicméně určení polohy uvnitř budov je nový a rozvíjející se obor, který představuje velkou výzvu.

K určení polohy uvnitř budov se dá využít již hotová infrastruktura (fungující WiFi přístupové body apod.) nebo je nutné pro konkrétní technologii vytvořit infrastrukturu novou (rozmístit NFC tagy - viz 3.6, QR kódy - viz 3.3, atd.)

3.2 GPS

GPS - Global Positioning System je původně vojenský družicový polohový systém, který provozuje Ministerstvo obrany Spojených států amerických. Pro určení polohy a času se využívá tzv. kódový přístup. Družice vysílají na Zem signály, které obsahují polohu družice v prostoru a časovou značku, kdy byl signál odeslán. Zařízení, které signály přijme, si vypočte, jaká je jeho poloha.

K určení polohy lze teoreticky využít signál pouze ze 2 družic, nicméně vypočtená poloha je velmi nepřesná. Ze signálů ze 3 družic lze určit polohu mnohem přesněji, nicméně se jedná o 2D určení polohy nerespektující nadmořskou výšku. Ze 4 a více družic lze pomocí metody triangulace získat

stabilní a přesnou polohu na jakémkoliv místě na planetě.[Cooksey()]

Nevýhodou GPS je skutečnost, že jeho signál není dostatečný, aby pronikl skrze stěny budov. Průnik signálu skrze stěny řeší projekt *Locata*, v části 3.7.

GPS není jediným globálním družicovým systémem pro určování polohy. Rusko má svůj *GLONASS*, Čína *Compass* a Evropa připravuje svůj v projektu *Galileo*. Všechny zmíněné systémy používají kódový přístup pro určení polohy. [Cooksey()]

Jelikož je GPS přijímač téměř v každém zařízení postaveném na platformě Android, je vhodné tuto technologii zahrnout do výčtu technologií pro budoucího klienta systému informačních majáků.

3.3 QR kód

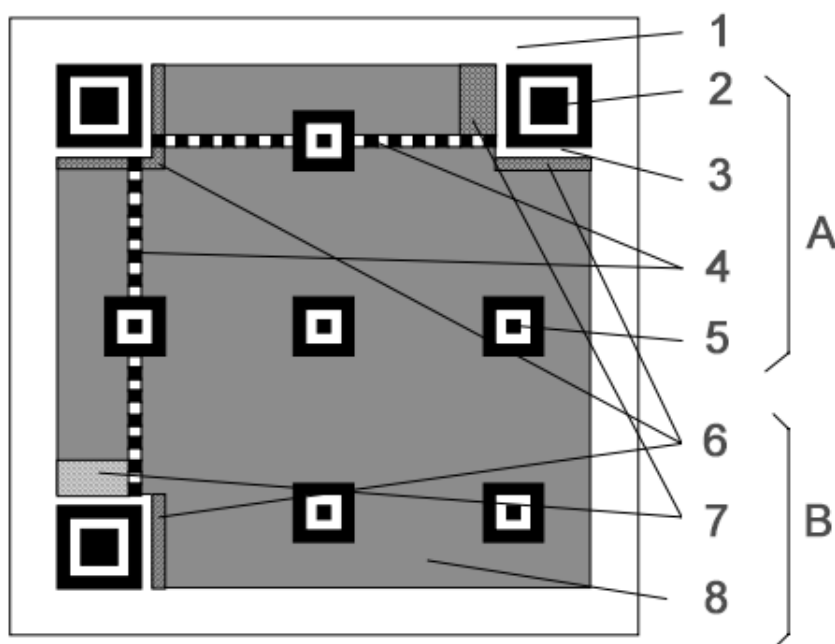
QR kód byl navržen v roce 1994 pro automobilku Toyota jako jakýsi rodokmen automobilu, jenž lze získat velmi rychle. Z toho plyne i význam písmen QR - Quick Response, nebo-li rychlá odpověď. Jedná se dvourozměrný systém čárových kódů, který narozdíl od klasických jednorozměrných čárových kódů, dokáže udržovat přibližně stokrát více dat.

Specifikace QR kódů obsahuje dnes již 40 verzí, přičemž s každou verzí roste množství dat, které lze do QR kód uložit. Do největšího z nich - verze 40 - je možné uložit až 4296 alfanumerických znaků nebo 2953 znaků z osmibitové znakové sady *ISO 8859-1* (někdy nazývané Latin 1).[qr-(2000)]

3.3.1 Jak přečíst QR kód

Na obrázku 3.1 jsou vysvětleny jednotlivé část QR kódu a jejich význam při jeho čtení. [qr-(2000)]

1. je tzv. *zóna tichá*. Je to místo, kde by se neměly vyskytovat žádné znaky. Zpravidla bývá buď 4x, 2x nebo 1x široká jako je nejmenší prvek na QR kódu - tzv. *modul* (viz níže)
2. jsou *polohovací vzory*, podle kterých čtečka dokáže určit hranice kódu.



Obrázek 3.1: Způsob rozložení a význam informací uvnitř QR kódu. [qr-(2000)]

Detekovat se dají jak horizontálně tak vertikálně podle následujícího vzoru: Č-B-Č-Č-Č-B-Č [Č = černá, B = bílá]

3. jsou bílé oddělovače pro polohovací vzory.
4. je velmi důležitá část. Skládá se z nejmenších částí QR kódu - **modulů**, které se pravidelně střídají. Jejich hlavním úkolem je určit mřížku ostatních modulů v celém QR kódu.
5. jsou *vyrovnávací vzory*. Pomáhají „narovnat“ QR kódy, které jsou nakreslené / nalepené na zakřivené ploše. Čím větší QR kód bude, tím více těchto vyrovnávacích vzorů bude mít.
6. značí *informace o formátu dat*. Skládá se z 15 bitů (modulů), přičemž těchto 15 bitů je v kódu umístěno dvakrát (jednou v celku vlevo nahoře, podruhé rozdělené napůl u zbývajících polohovacích vzorů). Tato informace je záměrně redundantní, protože dekodování celého kódu závisí na korektním přečtení této informace. Z 15 bitů je pouze 5 datových, zbylých 10 je pro korekci chyb.

7. obsahuje *informace o verzi*, jež určuje verzi QR obrazce, tzn. množství informací, které jsou v obrazci uloženy. Tento blok je 18 bitů (3x6 bitů) velký a stejně jako předchozí *informace o formátu dat* je v obrazci uložen redundantně. I tato informace je kritická pro správné dekódování QR kódu. Informace o verzi se vyskytují v QR kódech verze 7 a vyšší.
8. jsou *datová a opravná kódová slova*, která jsou 8 bitů dlouhá. Opravná kódová slova používají opravný algoritmus se 4 úrovněmi oprav. Nejnižší úroveň zaručuje 7% obnovení kódových slov a ta nejvyšší zaručuje 30% obnovení. Čím vyšší úroveň algoritmu je použita, tím je samozřejmě menší prostor pro data. Díky těmto opravným blokům lze vytvářet i „umělecké“ QR obrazce, kdy je uvnitř umístěn například malý obrázek, či logo.

Na základě dříve zjištěného kódování QR obrazce z bodu 6. jsou jednotlivé rozpoznané bity dekódovány a je z nich složena výsledná uložená informace.

Pokud bychom shrnuli výše vysvětlené body, dají se jednoduše rozdělit. **A** (body 2 - 5) jsou funkční vzory a **B** (body 6 - 7) jsou datové části.

3.3.2 Dostupné knihovny

Na Internetu je k dispozici hned několik volně dostupných knihoven pro dekódování QR obrazců.

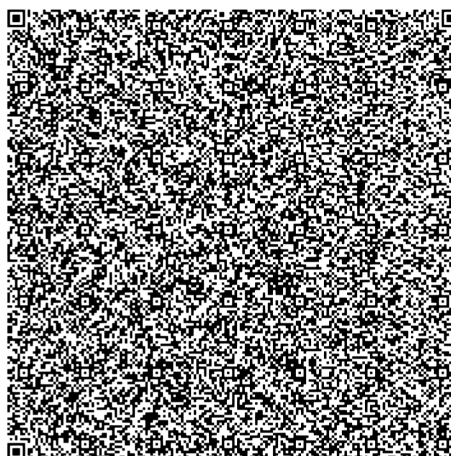
ZXing¹

ZXing je open-source knihovna pro čtení jedno i dvourozměrných čárových kódů (tedy i QR kódů) napsaná v programovacím jazyce Java s porty do ostatních jazyků. Dá se využít jak pro čtení QR kódů na zařízeních Android či iOS, tak i pro generování QR kódů.

¹dostupná na <https://code.google.com/p/zxing/>



(a) Jednoduchý QR kód se zakódovanou textovou zprávou.



(b) QR kód verze 40 - v současnosti největší QR kód. Je složen z 177 x 177 modulů.

Obrázek 3.2: Ukázka QR kódů

Zbar²

Stejně jako v předchozím příkladu, se i zde jedná o open-source knihovnu, která umí také číst vícero druhů čárových kódů. Navíc dokáže provádět skenování kódů v reálném čase například z video streamů. Knihovna je multiplatformní.

QR kód je určitě další z technologií, která by se měla uplatnit při realizaci systému a klienta pro informační majáky. Ke správnému přečtení QR kódu stačí pouze kamera, kterou má dnes naprostá většina mobilních zařízení.

3.4 WiFi

WiFi (nebo Wifi, Wi-Fi) je v současné době velmi rozšířená technologie umožňující zařízením vyměňovat si data bezdrátově. WiFi využívá rádiové vlny, které vysílá na frekvenci 2,4 GHz nebo 5 GHz. WiFi je standardizovaná technologie Institutem pro elektrotechnické a elektronické inženýrství (IEEE) pod označením *802.11x*, kde *x* zastupuje písmeno značící konkrétní standard.

²dostupná na <http://zbar.sourceforge.net/>

Vzhledem k dnešní rozšířenosti WiFi, kdy vysílajících zařízení je v obydlích oblastech a nákupních centrech apod. velká hustota, je WiFi dalším ideálním adeptem pro navrhovaný systém informačních majáků - zaměřeno na navigaci uvnitř budovy, kam nedosáhne GPS signál. Díky již zmíněné rozšířenosti WiFi není zpravidla ani nutné budovat dodatečnou infrastrukturu a v případě nutnosti vybudování WiFi infrastruktury nejsou dedikované vysílající zařízení nijak nákladnou záležitostí.

WiFi signál má uvnitř budovy přibližný dosah od 20 do 50 metrů (tento údaj je samozřejmě závislý na konkrétně použité specifikaci 802.11, síle použitého vysílače a charakteru překážek v okolí). Obecně platí, že síla signálu se zvyšující se vzdáleností od majáku klesá a naopak.

Veškeré WiFi přístupové body podle specifikace 802.11 vysílají v určitém intervalu tzv. *Beacon frame*, což je jeden z řídicích rámců, ve kterém jsou obsaženy všechny informace o dané síti, jako je např. BSSID, SSID³, možnosti sítě atd. To v praxi znamená, že jakékoliv (nejen) mobilní zařízení je schopné prozkoumat okolí a zjistit dostupné sítě v okolí identifikovatelné podle jedinečného identifikátoru v podobě BSSID a to i včetně síly signálu. Síla signálu se udává v jednotce *dBm*⁴. Podle výsledků průzkumu okolí lze zjistit polohu uživatele následujícími způsoby.

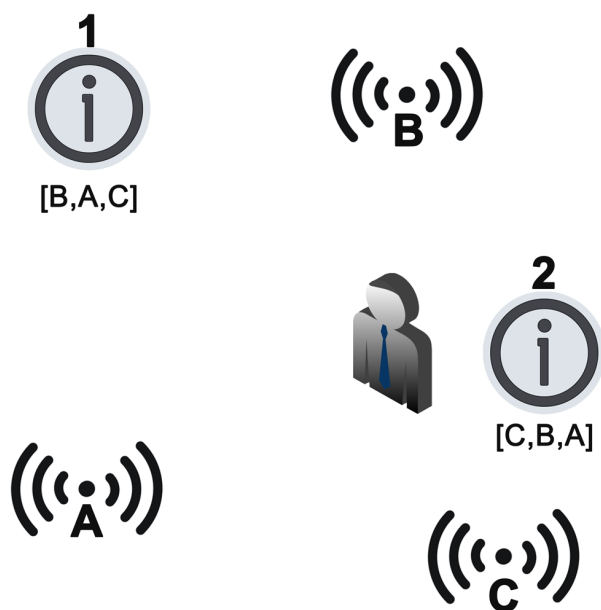
3.4.1 WiFi otisky

Tato metoda spočívá v průzkumu okolí v pravidelných intervalech. V samotném zařízení nebo systému, ke kterému se zařízení bude připojovat, musí existovat databáze WiFi přístupových bodů, jejichž kombinace (nejlépe seřazená podle síly signálu) je přiřazena ke konkrétnímu místu uvnitř budovy. Po prvním průzkumu okolí a porovnání dat s databází je možné určit přibližnou uživatelskou polohu uvnitř budovy a podle toho nechat systém dál reagovat.

Na obrázku 3.3 je rámcově znázorněno použití WiFi otisků. Na obrázku jsou 3 vysílače **A**, **B**, **C**, dva informační majáky **1** a **2**, kde WiFi otisk na jejich pozici je uveden v hranatých závorkách podle síly naměřeného signálu sestupně. Poslední je na schématu návštěvník. Pokud by na své současné pozici provedl průzkum okolí, získá WiFi otisk **[C,B,A]**, což podle uložených otisků systém vyhodnotí jako skutečnost, že se návštěvník s největší pravděpodob-

³neplatí v případě, že je vysílání názvu sítě zakázáno

⁴u 802.11 sítě se pohybuje od -90dBm do -10dBm



Obrázek 3.3: Příklad použití WiFi otisků.

ností vyskytuje v okolí informačního majáku 2.

3.4.2 WiFi triangulace

Metoda WiFi triangulace vychází z předchozí metody. I zde se provádí periodický průzkum okolí, ale měří se zde pokles či zvýšení síly signálu jednotlivých WiFi přístupových bodů v okolí vzhledem k předchozím měření. Zahrnutím dat z pohybových senzorů, které má většina mobilních zařízení (ať už postavené na platformě Android, iOS či Windows Phone), lze přesněji zjistit polohu návštěvníka i případně monitorovat jeho pohyb.

3.4.3 Omezení mobilních platform

Využití WiFi pro zjištění lokace je omezené pouze na platformu Android. Jak již bylo zmíněno výše, WiFi přístupové body lze jedinečně identifikovat pouze pomocí BSSID. K tomuto údaji se použitím API lze dostat pouze na platformě Android. WiFi, ať už triangulace či otisky, je další technologie, která má smysl při realizaci klienta informačních majáků.

Na platformě iOS není možné zjistit BSSID okolních zařízení aniž by aplikace nevyužívala privátní API, které pokud programátor v aplikaci použije, tak vytvořenou aplikaci nebude moci vystavit na portálu AppStore. BSSID lze zjistit pouze v případě, že je zařízení k WiFi síti připojeno, což není pro automatické zjišťování polohy podle okolních WiFi přístupových bodů použitelné.

SDK platformy Windows Phone je v tomto ohledu ještě omezenější, jelikož nenabízí žádnou možnost, jak zjistit BSSID⁵.

3.4.4 WiMax

WiMax je také bezdrátovou technologií (IEEE 802.16), která je ovšem zaměřená spíše na venkovní síť. I přesto ji někteří výrobci mobilních zařízení zařadili do své výbavy. WiMax sítě začaly být budovány v Americe v roce 2008, nicméně nezaznamenaly větší úspěch a brzy je nahradily LTE sítě. Významnějšího úspěchu dosáhl WiMax pouze v Rusku, Mongolsku a Pákistánu.[wim()] Díky tomu lze konstatovat, že WiMax není pro potřeby informačních majáků ideální technologií jako je např. WiFi, ale při jejím případném větším rozšíření by se dala využít alespoň jako doplňková.

3.5 Bluetooth

Bluetooth je bezdrátová technologie, která se využívá nejčastěji k výměně dat na kratší vzdálenosti. Bluetooth vysílače jsou podle specifikace SIG⁶ rozděleny do 3 tříd:

třída	dosah	maximální příkon
1	až 100 metrů	100mW
2	až 10 metrů	2,5mW
3	okolo 1 metru	1mW

Tabulka 3.1: Maximální dosah a příkon Bluetooth zařízení dle třídy.[bt()]

Každé zařízení - stejně jako tomu bylo v případě WiFi - má jedinečnou adresu, podle které je možné ho identifikovat.

⁵platí pro WP7 SDK i pro WP8 SDK

⁶SIG je organizace, která dohlíží nad specifikací Bluetooth

Maximální teoretická rychlost, kterou může Bluetooth dosáhnout, je pro systém informačních majáků nerelevantním údajem, jelikož stejně jako u WiFi má smysl používat Bluetooth pouze jako vysílače - není třeba zařízení párovat a vyměňovat data.

Výhodou Bluetooth je v případě třídy **2** a **3** jeho nízká energetická náročnost, přičemž malý dosah signálu není překážkou (může být naopak výhodou). Samotný vysílač bez jakékoliv další funkčnosti lze napájet pouze baterií a jeho pořízení není finančně nijak náročné.

3.5.1 Využití

Oproti WiFi, která je natolik rozšířená, že ve valné většině případů není potřeba dotvářet infrastrukturu vysílačů, je Bluetooth v nevýhodě. V případě, že by systém využíval pouze Bluetooth vysílače, musela by být s největší pravděpodobností vytvořena nová síť vysílačů. Se sítí vysílačů, které mají pevné a známé umístění, může být určení polohy návštěvníka o to přesnější.

Mírně upravené nízkoenergetické Bluetooth 4.0 majáky použila společnost Nokia a vyvinula systém HAIP⁷, který se má vyznačovat velmi velkou přesností (určení polohy až na 20cm). [bt-()]

Pro Bluetooth lze využít obdobné metody, tj. metoda *otisků* a metoda *triangulace*, jako u WiFi.

3.5.2 Omezení mobilních platforem

I zde je omezení obdobné jako u případu WiFi. Pouze platforma Android je schopná programově zjistit jméno a unikátní adresu vysílajících Bluetooth zařízení v okolí. Nedovede ovšem už zjistit sílu přijatého signálu, čímž se vylučuje použití zmíněné metody *triangulace* a zbývá pouze metoda *otisků*. Metoda Bluetooth otisků v případě realizace Android klienta informačních majáků má určitě smysl.

Na platformě iOS je omezení také obdobné jaké bylo u WiFi. Jméno a unikátní adresu sice lze zjistit, ale pouze s použitím privátního API, což opět

⁷High Accuracy Indoor Positioning

znemožňuje vystavit aplikaci na portálu AppStore.

Pro Windows Phone 7 SDK není žádná možnost, jak pracovat s Bluetooth rozhraním. WP8 SDK již nabízí základní možnost, jak s Bluetooth rozhraním pracovat, bohužel ale nenabízí možnost zobrazit okolní viditelná zařízení.

3.6 Near Field Communication

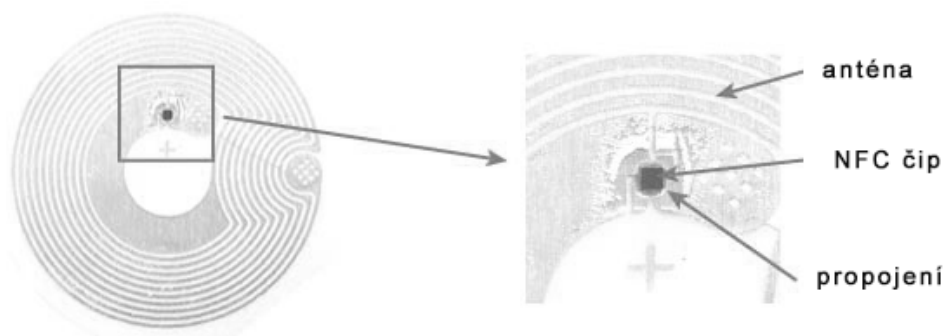
Near Field Communication, dnes známá spíše pod zkratkou NFC je masově rozvíjející se „bezkontaktní mobilní technologií“. V rámci NFC je sada standardů, jakým způsobem mají zařízení navázat komunikaci mezi sebou. Jedná se o bezdrátovou technologii, která umožňuje bezkontaktní platby, čtení / zápis dat pouze na velmi malé vzdálenosti (zpravidla⁸ nesmí vzdálenost být větší než dvacet centimetrů) apod. NFC je navrženo tak, aby bylo kompatibilní s existujícími standardy bezkontaktních karet a již vytvořenou infrastrukturou. [nfc(a)]

Maximální rychlost přenosu dat mezi dvěma aktivními zařízeními může dosáhnout až 1Mbit/s. V případě komunikace aktivního zařízení s pasivním je rychlost až 424 kbit/s, což není nijak velká rychlost, ale vzhledem k vlastnostem pasivních zařízení (více v 3.6.1) a filosofie NFC je dostačující.[nfc(a)]

komunikace aktivní-aktivní zařízení

Při tomto druhu komunikace jsou obě zařízení nezávisle na sobě napájena a jsou tedy aktivní. Takovým příkladem komunikace jsou dvě zařízení smartphone vyměňující si data. NFC se v tomto případě vyplatí pouze na přenos malého objemu dat, jako je URL webové stránky, vizitky atd. Při větším objemu se dat, jako například multimediální soubory, se NFC použije pouze jako prostředek k vytvoření například Bluetooth nebo *WiFi Direct* spojení, přes které jsou pak data poslána významně větší rychlostí.

⁸již existují *NFC proxy* projekty, které přes Internet „zvýšit“ dosah na libovolnou vzdálenost



Obrázek 3.4: Jak vypadá NFC tag uvnitř. [nfc(b)]

komunikace aktivní-pasivní zařízení

V pasivním módu probíhá komunikace mezi aktivním zařízením, které komunikaci inicializuje, a nenapájeném zařízením - tzv. *tag*. Jeho největší výhodou je, že nepotřebuje žádné baterie, aby udržel uložená data. Anténa pasivního zařízení (složení je ilustrováno na obrázku 3.4) je navržena tak, aby měnila okolní magnetické pole v energii využitelnou pro své napájení. Anténa reaguje pouze na magnetické pole o frekvenci 13,56MHz, které vytváří aktivní zařízení, chce-li komunikovat.

Velikost antény tagu se odráží na kvalitě komunikace. Čím je anténa větší, tím lépe dokáže napájet tag a přenosová rychlost je rychlejší.

3.6.1 Typy tagů, komunikace

Podle současné specifikace organizace NFC Forum jsou 4 typy NFC tagů, kde každý typ má jiné parametry. Základní vlastnosti:

- **Type 1:** přepisovatelný, lze uzamknout paměť proti přepsání, velikost mezi 96B - 2kB.
- **Type 2:** přepisovatelný, lze uzamknout paměť proti přepsání, velikost mezi 48B - 2kB.
- **Type 3:** přepisovatelný, lze trvale uzamknout paměť proti přepsání, velikost až 1MB.

- **Type 4:** prepisovatelný, lze trvale uzamknout paměť proti přepsání, velikost až 32kB.
[nfc(d)]

Velmi rozšířeným zástupcem je čip typu 2 *Mifare Classic*⁹, který vychází ze specifikace organizace NXP a není tedy vyhovujícím NFC tagem. Čtení / zápis na tento čip je nicméně podporováno většinou aktivních NFC zařízení, jelikož jsou vybaveni NFC čipem právě od NXP. Ceny NFC tagů se samozřejmě liší podle druhu. Již zmíněný tag Mifare Classic 1k se ve formě nálepek či plastových přívěsků dá pořídit od 30,-Kč za jeden kus.

NFC Forum specifikovalo také strukturu dat pro komunikaci s NFC tagy. Formát dat se nazývá *NDEF*. Na tagy lze zapisovat i po jednotlivých bitech na základě svého vlastního protokolu. Takový tag je ovšem čitelný pouze čtečkou, která vlastnímu protokolu rozumí.

3.6.2 Využití

NFC je technologie, která bude určitě zahrnuta při realizaci Android klienta systému informačních majáků. Díky komunikaci pouze na pár centimetrů lze návštěvníkovu polohu určit velmi přesně (samozřejmě za předpokladu, že NFC tag bude nepohyblivý a jeho poloha bude přesně známa). V tagu mohou navíc být uložena libovolná data, které mohou vyvolat libovolnou akci.

Platforma Android je plně vybavena pro komunikaci s NFC čipy. Platforma iOS nemá pro komunikaci jakoukoliv podporu, což logicky vychází ze skutečnosti, že společnost Apple nemá ve svém portfoliu žádné zařízení podporující tuto technologii. Windows Phone podporuje NFC až od SDK verze 8.

Tagy nemusí být omezeny pouze pro čtení. Návštěvník na něj může i zapisovat data, čímž se otevírá velké množství scénářů použití. Při každém čtení se může například automaticky inkrementovat počet přečtení. Dalším příkladem může být spojení se sociální sítí - při čtení se přečte předchozí uložená adresa profilu na sociální síti a do tagu se uloží nová adresa profilu uložená v zařízení návštěvníka. Profil „posledního“ návštěvníka bude nabídnut současnému návštěvníkovi k další akci (přidání mezi přátelé, napsat zprávu apod.).

⁹např. Plzeňská karta nebo JIS karta jsou typu Mifare Classic 1k

3.6.3 Zabezpečení vs. interaktivita

Při interaktivním využívání tagů je třeba myslet i na zabezpečení. Například pokud NFC tag pouze obsahuje URL profilu na sociální síti, který telefon poté pouze otevře, může útočník podvrhnout URL za svoji stránku se škodlivým kódem. Řešením je trvalé zamknutí čipu proti přepsání, čímž ovšem ztrácíme interaktivitu. Dalším řešením je NFC tag *zašifrovat*. Tuto možnost ovšem nemá každý NFC čip. Příkladem je například bloková šifra **3DES** nebo **Crypto-1**.

Crypto-1 byla vyvinuta přímo pro NFC tady, využívající 48 bitové klíče. Díky této slabíně v podobě 48 bitových klíčů již byla překonána a metodou hrubé síly lze šifru prolomit do 50 minut.[nfc(c)]

3.7 Locata

Locata je projekt řešící některé nedostatky GPS, jako je například slabý signál uvnitř budov. Projekt by neměl sloužit jako náhrada GPS, nýbrž jako rozšíření a doplnění GPS.

Locata pokrývá vždy určitou oblast - není celosvětová. Vybraná oblast se musí pokrýt pozemními vysílači. Všechny vysílače v dané oblasti jsou propojeny a neustále je jim synchronizován vnitřní čas. Stejně jako GPS i Locata vysílá signály se svoji polohou a časovou značkou, které koncové zařízení zachytává a určuje tak svoji polohu.

Hlavním rozdílem je frekvence, na které vysílače vysílají (2,4 GHz), a výkon vysílaného signálu, který je oproti GPS podstatně větší. To této technologii umožňuje, aby byl její signál zachytitelný i uvnitř budov.[loc()]

3.7.1 Využití

Pro systém informačních majáků, kde klienti jsou pouze mobilní zařízení, je bohužel tato technologie nepoužitelná. V současné době neexistuje žádné mobilní zařízení, které by dokázalo tento signál zachytit, jelikož přijímače nedosáhly takového stupně miniaturizace, aby se daly umístit přímo do mobilního zařízení.

3.8 Rozpoznání obrazu scény

Další možností jak určit, že návštěvník je na konkrétním místě v instituci, je porovnání okolní scenérie s nasnímanou scenérií. Metoda se zdá být využitelná, jelikož návštěvník potřebuje pouze zařízení s fotoaparátem, který je dnes standardní výbavou, a výpočetní výkon, který dnešní zařízení dovedou bez problémů poskytnout.

Snímání konkrétního místa

V tomto případě by návštěvník musel dopředu vědět, jakou scénu má na mobilním zařízení zachytit. Ta by se porovnávala se uloženými scénami v systému a podle podobnosti by se dalo rozhodnout, kde návštěvník stojí. Jednotlivé scény by samozřejmě v systému musely být uloženy společně s polohou odkud byly pořízeny.

Průběžné snímání

Oproti předchozímu příkladu by zde probíhalo průběžné snímání scény a neustálé porovnávání dat se systémem. Tento přístup se jeví jako nevýhodný, protože neustálé porovnávání by vyžadovalo vyšší výpočetní výkon, což by se nepříznivě projevilo na výdrži baterie mobilního zařízení.

Porovnání by bylo možné přesunout do systému (na serverovou část). To by ovšem znamenalo každou zachycenou scénu odeslat na server, což by mohlo generovat velký datový tok.

At' už se jedná o snímání konkrétního místa nebo o průběžné snímání, může být tato metoda nepřesná. Hlavně u snímání venkovních scén, kdy nemusí být světelné podmínky konstantní, může systém vyhodnotit podobnosti chybně. Určitou pomocí v tomto problému by mohlo být vybudování sítě statických kamer, které by v určitém intervalu snímaly konkrétní scénu a aktuální snímek poslaly do systému jako předlohu pro porovnávání.

Dalším negativním vlivem na správnost porovnání scén může být kvalita fotoaparátů v mobilních zařízeních, kde se rozlišení snímače fotoaparátu může pohybovat od 0,3 Mpx do 15 Mpx.

Obrazy by šly například porovnávat podle jejich histogramů, což při výše uvedených škodlivých vlivech nemusí být dostatečně přesné. Další metody a algoritmy porovnávání obrazů zde nebudou uvedeny, jelikož to není náplní této práce.

Závěrem pro tuto metodu lze konstatovat, že je pro informační systémy spíše nepoužitelná.

3.9 LED diody

LED diody jsou dalším nástrojem, jak lze zjistit polohu návštěvníka. Lze totiž využít toho, že její svítící P-N přechod diody nemá žádnou setrvačnost, jako je tomu u klasické žárovky, tudíž rozsvícení a zhasínání je prakticky okamžité. LED dioda potom může blikat určitou unikátní frekvencí nebo podle předepsaného unikátního vzoru tak, že pro lidské oko bude toto blikání nepostřehnutelné a bude se jevit jako konstantní svit.

Tuto frekvenci či vzor je schopný zachytit fotoaparát, který blikání dokáže detekovat a zpracovat. Na základě zpracované informace je možné zjistit, kterou diodu zařízení zachytilo a se znalostí rozmístění každé diody je možné relativně přesně určit aktuální polohu uživatele.

3.9.1 Využití

Využití LED diod je omezeno výhradně na vnitřní použití, jelikož použití ve venkovních prostorech by bylo značně ovlivněno okolním světlem. Navíc samotné použití LED diod / žárovek nemá přes den venku opodstatnění (pro venkovní prostory je ideální technologií GPS).

Při použití ve vnitřních prostorech by se musela vybudovat síť blikajících diod. Navíc jejich rozmístění by muselo být vhodně uspořádáno, aby nedocházelo k jejich vzájemnému rušení. I ve vnitřních prostorech by mohlo dojít přes den k rušení venkovním světlem, pokud by byly diody nevhodně umístěny - například blízko u oken.

Pro realizovaný systém informačních majáků bylo rozhodnuto tuto technologii nevyužít, kvůli náročnosti vybudování infrastruktury.

Další věcí, na kterou je třeba dát pozor při určování frekvence blikání jednotlivých diod je, aby blikání nebylo postřehnutelné lidským okem. Nevhodné blikání by mohlo být zdrojem problémů pro návštěvníky, kteří trpí epileptickými záchvaty.

Tuto technologii vzal za svou systém *ByteLight*, který vytvořil vlastní LED žárovky použitelné jako pro osvětlení tak zároveň i pro identifikaci. Více je o systému uvedeno v sekci 2.6 na straně 11.

3.10 Zvuk

Dalším využitelným zdrojem, který je všude kolem nás, je zvuk. Stejně jako v předchozím příkladu s LED diodami, kde se využívalo jedinečné frekvence blikání, které bylo navíc lidským okem nepostřehnutelné, by se dal obdobně využít zvuk.

Zvuk by buď musel být za hranicí slyšitelnosti lidským uchem, nebo by musel sloužit jako kulisa k určité situaci v instituci (například v zoologické zahradě by se mohlo jednat o zvuky zvířat nebo prostředí, ve kterém žijí). Zvuky, ať už lidským uchem slyšitelné či neslyšitelné, by zachytávalo mobilní zařízení, které by okamžitě provádělo jeho analýzu.

Analýza zvuku by se na mobilních zařízeních dala provádět technikou zvanou Rychlá Fourierova transformace (FFT). Výhodou je, že FFT se dá upravit, aby nebyla výpočetně tak náročná a dala se provádět v reálném čase i na mobilních zařízeních.

3.10.1 Využití

Jak již bylo zmíněno, výpočetní náročnost analýzy signálu by nebyla na mobilních zařízeních překážkou. Všechny mobilní platformy mají dostupné API, pro práci s mikrofonom. Problémem by mohlo být okolní rušení. Například v uzavřené místnosti plné lidí, by mohl být zachycený zvukový signál natolik zkreslen, že nebude rozpoznán.

Z výše uvedených důvodů nebude tato technologie zahrnuta do připravovaného klienta informačních majáků.

3.11 Další metody

Následující metody jsou založeny čistě na znalostech či aktuálních vjemech uživatele. Nevyužívá se zde jakéhokoliv elektromagnetického záření, či programového zpracování obrazu či zvuku, jako tomu bylo ve výše uvedených možnostech identifikace.

3.11.1 Metody závislé na znalostech

U této metody by se dalo využít znalostí návštěvníka. Může se jednat například o metodu hádanek. Pokud bude znění hádanky umístěno pouze na jednom konkrétním místě, po jejím správném zodpovězení je jasné, kde se návštěvník pohybuje. Metoda bude blíže vysvětlena na konkrétní modelové situaci.

Představme si návštěvníka zoologické zahrady, který přijde k výběhu slonů. Na malé informační tabulce, kde bývají základní informace o daném zvířeti, bude text hádanky (např. „Kolik je slonů v naší zoologické zahradě?“). Pokud ji návštěvník uhádne a do klienta napíše správnou odpověď, bude mu zobrazen další, například i multimediální obsah o daném zvířeti. Odpověď může být validována přímo na mobilním klientovi, či přes Internet v serverové části systému.

Problém u této metody může být v jazykové bariéře. Nemusí se tedy hodit pro instituce, kde je předpoklad, že ji budou navštěvovat i návštěvníci ze zahraničí. Další překážkou by mohla být nevhodně zvolená hádanka, na kterou lze odpovědět i více způsoby, případně je hádanka natolik složitá, že ji uhodne pouze malé procento návštěvníků a tím budou ostatní zbytečně ochuzeni o bonusový materiál.

Možností rozšíření této metody je velké množství. Metoda by se například dala rozšířit o bonusové body za každou správně uhodnutou hádanku. Předem definovaný počet získaných bodů by mohl návštěvník vyměnit za správnou odpověď na hádanku, kterou nemůže vyřešit. Návštěva instituce se poté stává jakousi hrou.

3.11.2 Metoda založená na všímavosti

Tato metoda je velmi podobná předchozí metodě. Zde se ovšem nevyužívá návštěvníkův intelekt, nýbrž jeho všímavost. Metoda bude opět vysvětlena na příkladu.

Při vstupu do instituce bude návštěvníkovi dáno na vědomí, že jsou u jednotlivých exponátů skryty například číselné kódy, které má hledat. Po nalezení konkrétního číselného kódu, ho návštěvník zadá do klientské aplikace a ta mu v případě správnosti může zobrazit další obsah spojený s tímto číselným kódem podobně, jako tomu bylo u předchozí *metody hádanek*.

Tato metoda řeší problém předchozí metody s jazykovou bariérou, přičemž je do jisté míry zachován model hry. Číselný kód zapsaný arabskými číslicemi je čitelný pro většinu populace, jelikož se jedná o dnes nejrozšířenější systém zápisu čísel.

Díky své obecnosti a jednoduchosti bude tato metoda zahrnuta do realizovaného klienta systému informačních majáků.

3.12 Odhalení podvodů

Zavedení systému informačních majáků může být pro libovolnou instituci časově případně i finančně nákladné. Pokud by identifikace informačních majáků návštěvníkem byla odměněna bonusovým multimediálním obsahem, může se stát, že provozovatel instituce nebude chtít, aby se k bonusovému obsahu dostali i lidé, kteří jeho instituci nenavštívili. V případě, že při identifikaci majáků bude uživateli zobrazeno např. pouze reklamní sdělení, není třeba tento problém řešit.

Představme si ovšem situaci, kdy bonusový materiál v podobě videí, jejichž vytvoření bylo nákladné, je dostupný po zadání číselné kombinace. Tyto kombinace se dají po celé instituci vyhledat a libovolně nasdílet na Internetu. Pak tyto číselné kódy může zadat do klientské aplikace uživatel, který instituci nikdy nenavštívil, a přesto dostane bonusový materiál.

Tento problém se netýká technologií WiFi, Bluetooth, LED diody a zvuk. Uživatel, který by se chtěl neoprávněně dostat k bonusovému obsahu, by musel nasimulovat stejné podmínky, jako jsou u konkrétního majáku. U pří-

padu WiFi by musel vytvořit identickou infrastrukturu vysílačů (tzn. stejné BSSID s příslušným výkonem), pro metodu LED diod by musel zjistit frekvenci a vzor blikání diody apod. Úsilí spojené se zjištěním nutných informací bude zpravidla větší, než klasická návštěva instituce.

3.12.1 Spojení s GPS polohou

Mnohem menší úsilí vyžaduje zjištění bonusových kódů (viz 3.11.2), okopírování QR kódů (viz 3.3) či přečtení obsahu NFC tagu a jeho znovu vytvoření (pokud není tag zašifrován, viz 3.6)

Jedním z možných řešení je kontrola aktuální polohy uživatele. Před zobrazením bonusového obsahu je možné kontrolovat polohu uživatele a v případě, že se od instituce nachází dál, než je předem definovaný počet metrů (kilometrů), nebude mu obsah zobrazen. I aktuální polohu uživatele, lze na mobilních zařízeních oklamat pomocí tzv. *mock locations*, nicméně existují metody, jak tomu zabránit.

3.13 Přínos pro instituci

Implementace systému informačních majáků může být sice nákladná, nicméně zvedá úroveň interaktivity a atraktivity dané instituce. Informační majáky nemusí plnit pouze funkci nositelů obsahu, ale mohou být součástí soutěže. Lze si představit soutěž, kde návštěvníci soutěží v počtu nalezených majáků. Za určitý počet nalezených majáků může být návštěvník například odměněn reklamními předměty, slevou na další vstupné atd.

Vyhlášení nových soutěží může přilákat potencionální návštěvníky k opětovné návštěvě, pro kterou by jinak neměli důvod apod.

V navrhovaném systému budou k identifikaci majáků použity technologie GPS, QR kód, WiFi, Bluetooth, NFC a metoda založená na všímavosti (v podobě číselných kódů). Realizovaný mobilní klient na platformě Android bude schopen využívat všechny vyjmenované technologie a metody. Ostatní platformy (iOS, Windows Phone) budou využívat pouze podmnožinu technologií, kterou daná platforma umožňuje.

4 Platforma Android

Platformu Android není nutné v dnešní době nijak zvlášť představovat. Jedná se o mobilní platformu, na které jsou dnes postaveny stovky miliónů nejen mobilních zařízení po celém světě, a jejich počet každým dnem roste. Každý den je po světě aktivováno více než 1 milion zařízení.¹ Detailní popis celé platformy je uveden v mé bakalářské práci s názvem *Knihovna pro tvorbu síťových aplikací v OS Android* v kapitole 2 a 3.

Výše odkazovaná bakalářská práce popisuje platformu do verze 3 vyjma. V dnešní době je tato platforma ve verzi 4.0 (kódové označení Ice Cream Sandwich), 4.1, 4.2 (Jelly Bean). Verze 4 měla hlavně za úkol sjednotit verze a použití API mezi mobilními zařízeními a tablety a celkově více sjednotit doted' roztržitěnou platformu (co se verzí týče). 5. verze s kódovým označením Key Lime Pie se připravuje k vydání. Představena by měla být na festivalu *Google I/O* v půlce května letošního roku.

4.1 Novinky

V této sekci budou uvedeny pouze některé novinky² od verze 3, které jsou buď velmi významné nebo byly použity při tvorbě klienta informačních majáků.

4.1.1 Project Butter

Od verze 4.1 přidal Google do výbavy platformy tzv. *Project Butter*, který odstraňuje jeden z velkých problémů platformy, a tím je pomalá reakce systému a nepřiliš plynulé grafické efekty.

V překladu se „Projekt Máslo“ jmenuje velmi příznačně, jelikož s jeho implementací běží grafické efekty a odezva „jak po másle“. Využitím metody *tripple buffering* se zvedl počet snímků za vteřinu (*fps*) až na hodnotu 60.

¹údaj z dubna 2013 [and(e)]

²výpis všech novinek je dostupný na adrese <http://developer.android.com/about/versions/index.html>



Obrázek 4.1: Ukázka UI prvku Action Bar[and(a)]



Obrázek 4.2: Ukázka UI prvku Title Bar

4.1.2 Vložené fragmenty

Od verze 4.2 (API 17) je možné nejen vkládat fragmenty do aktivit, ale fragmenty do již vložených fragmentů. Tím se dají vytvářet znovupoužitelné komponenty (fragmenty), které lze využít pro tvorbu dynamického UI. Více o fragmentech je k nalezení v následující sekci 4.3.

4.1.3 Action Bar

Action Bar je prvek UI, který od verze 3 kompletně nahrazuje dosavadní *title bar* umístěný vždy na vrchu aplikace. Jedná o jeden z nejdůležitějších UI prvků, který lze v aplikaci implementovat. Obsahuje logo aplikace³ a navigační prvky, ať už se jedná o tlačítko *up*, které prochází zpětně aplikaci dle stavu zásobníku, či záložkové menu nebo roletovou navigaci. Na obrázku 4.1 jsou vidět výše zmíněné komponenty. Pod číslem 1 je logo aplikace a navigační tlačítko *up*, pod číslem 2 je navigační komponenta. Pod číslem 3 a 4 jsou jednotlivé prvky *Action Menu*. Ty jsou buď vidět jako samostatné ikony, pokud to tak programátor zamýšlel a je na Action Baru dostatek místa a nebo jsou skryté a zobrazí se až po kliknutí na symbol zde označený číslem 4.

Celý Action Bar je upravitelný, tudíž není problém do něj vložit například vyhledávací pole, použít na něj styly, aby vyhovoval grafickému designu aplikace nebo ho lze úplně skrýt. Oproti *Title Baru* nabízí mnohem větší prostor a funkčnost, což je zřejmé při porovnání obrázků 4.1 a 4.2

³základní nastavení, lze samozřejmě skrýt či nahradit za jakýkoliv libovolný obrázek

4.1.4 API podpory různých obrazovek

Od verze 3.2 nabízí platforma nové přepracované API pro podporu různých velikostí a rozlišení obrazovek. OS Android neběží totiž výhradně na mobilních zařízeních, ale i například na televizích. Na vývojáře je ze strany společnosti Google vyvíjen tlak, aby svou aplikaci vytvořili pouze v jedné verzi, která si sama poradí s různým rozlišením a velikostí obrazovek. Toto nové API tento úkol vývojářům usnadňuje.

Vývojáři tedy stačí naprogramovat logickou část aplikace pouze jednou a poté „jen“ vytvořit různé rozložení prvků na základě typu a rozlišení obrazovky, na které daná aplikace zrovna běží. Nové API přidává nové konfigurační kvalifikátory, u kterých lze přesně specifikovat šířku / výšku v jednotkách *dp*:

swXXXdp	Určuje minimální šířku v <i>dp</i> , pro kterou se mají zdroje použít. Tato hodnota je nezávislá na konkrétní orientaci zařízení.
wXXXdp	Minimální šířka v <i>dp</i> , pro kterou se mají příslušné zdroje použít. Tato hodnota je závislá na konkrétní orientaci zařízení.
hXXXdp	Analogicky v předchozím řádku pro výšku.

Tabulka 4.1: Nové kvalifikátory.[and(b)]

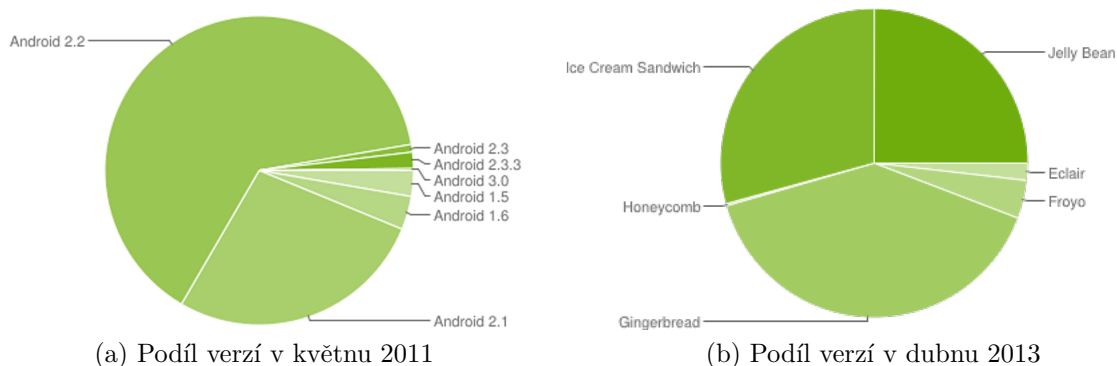
Kvalifikátory se dají samozřejmě i kombinovat, čímž lze dosáhnout kvalitního UI pro téměř jakékoliv zařízení. Příklady jsou uvedeny v tabulce 4.2.

kvalifikátor	příklad použití
sw600dp	nejčastěji 7" tablety
sw720dp	nejčastěji 10" tablety
w600dp	minimální šířka pro použití více panelového UI
sw600dp-w720dp	zařízení s velkou šířkou

Tabulka 4.2: Příklad použití kvalifikátorů.[and(b)]

4.2 Přehled verzí

Jak je vidět na obrázku 4.3, celá platforma prochází neustálým a docela živým vývojem. Za necelé dva roky se dříve majoritní zastoupení verze 2.2 (*Froyo*) změnilo na minoritní zastoupení s 4% podílem. Tuto verzi v průběhu



Obrázek 4.3: Porovnání podílu jednotlivých verzí OS Android

významně nahradila verze 2.3.x, která je ovšem za svým vrcholem a každý měsíc ztrácí přibližně 5% podílu na úkor nové verze 4+. I verze 3.x je postupně utlumována, jelikož se ukázalo, že šlo pouze o přechodnou verzi pro tablety a od verze 4 obsahuje Android SDK nástroje, aby mohly aplikace fungovat korektně jak na tabletech, tak telefonech.

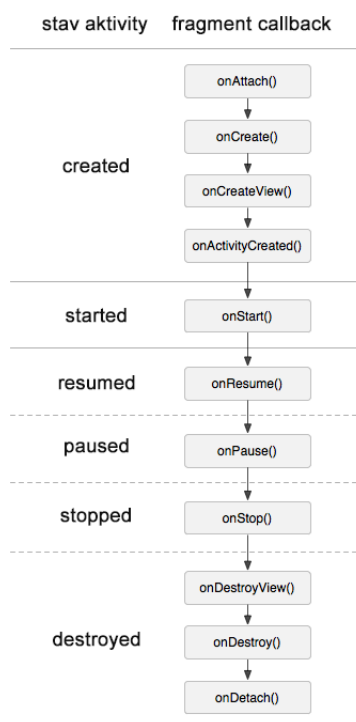
4.2.1 Support Library

Ačkoliv zastoupení verze 2.3.x na trhu stále klesá, je počet zařízení stále významný. Jelikož se platforma Android snaží zmírnit svou roztříštěnost, připravila tzv. *support library*, která umožňuje používat většinu nových funkcí API i na starších zařízeních s nižší verzí OS Android. Jejím cílem je zjednodušit vývoj a umožnit vývojáři méně se starat o verze, na kterých jeho aplikace poběží.

Tato knihovna umožňuje používat nové API na zařízeních **už** od verze **1.6**, kterých je méně než 0.1%. Je tedy např. možné využívat fragmenty, ve verzi, která na ně nikdy nebyla připravena. V případě využití knihovny, musí být samozřejmě tato knihovna součástí aplikace.

4.3 Fragmenty

Prvek *fragment* je sám o sobě novinkou, která víceméně změnila celý návrh aplikací, a proto je mu věnována celá sekce. Fragmenty se v platformě objevily

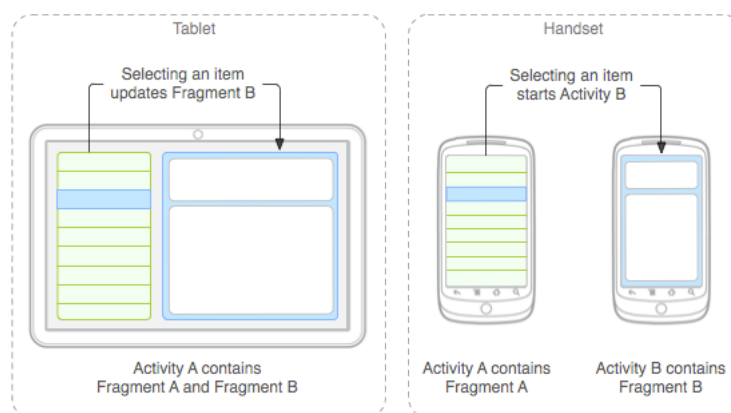


Obrázek 4.4: Životní cyklus fragmentu.[and(c)]

od verze 3.0.

Fragment se nejčastěji používá jako část uživatelského rozhraní. Lze o něm smýšlet jako o „podaktivitě“, která má svůj vlastní životní cyklus. Životní cyklus fragmentu se liší pouze v pár detailech od životního cyklu *aktivity*, je ovšem ovlivněn životním cyklem aktivity, ve které je umístěn. Fragment může existovat i bez uživatelského rozhraní, je tedy pro uživatele neviditelný a provádí pouze práci na pozadí.

Při tvorbě fragmentu by měl programátor implementovat minimálně metody `onCreate()`, `onPause()` a v případě, že je fragment součástí uživatelského rozhraní, tak i metodu `onCreateView()`, která vrací vykreslené uživatelské rozhraní.



Obrázek 4.5: Ukázka vícenásobného použití fragmentu.[and(c)]

Vložení fragmentu

Fragment sám o sobě nemůže existovat, vždy musí být vložen v nějaké aktivitě. Pokud se fragment používá jako část uživatelského rozhraní, měl by být navržen jako znovupoužitelná komponenta aktivity. Filosofie fragmentu totiž říká, že by fragment měl být použitelný ve více aktivitách. Ukázka použití je vidět na obrázku 4.5. V levé části je vidět tablet, který si může díky velké obrazovce dovolit umístit jak fragment se seznamem, tak fragment s detailem do jedné aktivity. Naproti tomu v pravé části je vidět klasický telefon s nedostatečnou šířkou displeje. Detail je proto zobrazen samostatně v další aktivitě.

Fragment lze do aktivity vložit buď staticky v rámci předem definovaného uživatelského rozhraní aktivity v XML souboru nebo dynamicky za běhu aplikace. První případ je vidět na následujícím výpisu kódu. V druhém případě se využívá tzv. *transakcí fragmentů*. Každá aktivita má svůj *FragmentManager*, který umožňuje vkládat fragmenty do aktivity za běhu. Příklad použití je uveden za následujícím výpisem kódu. Po zavolání metody `commit()` je transakce dokončena a fragment je uživateli zobrazen. Voláním metody `addToBackStack()`, se fragment přidá do zásobníku aplikace, tudíž při stisku tlačítka „zpět“, se provede odstranění fragmentu, nikoliv celé aktivity.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/
   res/android"
3     android:layout_width="match_parent"

```

```
4     android:layout_height="match_parent "
5     android:orientation="vertical" >
6     <fragment
7         android:id="@+id/supportMapFragment "
8         android:layout_width="match_parent "
9         android:layout_height="match_parent "
10        class="com.google.android.gms.maps.SupportMapFragment
11        " />
12 </LinearLayout>
```

```
1 //ukazka pouziti transakci
2 Fragment fragment = new ReviewsFragment();
3 FragmentTransaction ft = getSupportFragmentManager().
4   beginTransaction();
5 ft.replace(android.R.id.content, fragment);
6 ft.addToBackStack(null);
7 ft.commit();
```

4.3.1 Komunikace s aktivitou

Ačkoliv je fragment komponenta, která má být použita nezávisle na aktivitě, je občas nutné, aby s aktivitou komunikoval. K tomu je třeba vytvořit si *callback* rozhraní, které bude fragment po aktivitě požadovat, aby ho implementovala. Při vytváření fragmentu, si v metodě `onAttach()` fragment otestuje, zda je jeho požadavek splněn.

```
1 private GetContentDataIface mActivityRef;
2 //callback rozhrani
3 public interface GetContentDataIface {
4     public Beacon getBeaconData();
5     public List<Resources> getAllResources();
6 }
7
8 @Override
9 public void onAttach(Activity activity) {
10     super.onAttach(activity);
11     try {
12         //test zda je splnen požadavek fragmentu
13         mActivityRef = (GetContentDataIface) activity;
14     }
15     catch (ClassCastException e) {
16         throw new ClassCastException(activity.toString() + "┐must
17         ┐implement┐GetContentData┐interface");
18     }
19 }
```

4.4 Vhodnost pro realizaci klienta

Platforma Android je pro realizaci klienta systému informačních majáků více než vhodná. Je to především zásluhou její rozšířenosti po celém světě, ale také díky otevřenosti této platformy.

V kapitole 3 bylo uvedeno množství způsobů, jak lze informační majáky identifikovat s využitím mobilního zařízení. V případech 3.4 a 3.5, kdy se jedná o poměrně klíčové technologie⁴ pro identifikaci majáků, se vyskytl u platforem iOS a Windows Phone problém. Není totiž možné programově získat potřebné informace. U platformy Android k žádnému takovému omezení nedochází.

Lze tedy konstatovat, že má smysl realizovat klienta pro systém informačních majáků na platformě Android.

⁴WiFi a Bluetooth

5 Mobilní klient pro platformu Android

V této kapitole bude podrobně popsán proces tvorby mobilního klienta pro platformu Android. Nejprve však bude popsán systém, ke kterému klient přistupuje.

5.1 Popis vytvořeného systému Tagiee

Systém, jako serverová část pro mobilní klienty, byl vytvořen v rámci diplomové práce *Systém pro správu a realizaci informačních majáků*, kterou vypracoval Bc. Luděk Vlk v roce 2013.

Vytvořený systém dostal název *Tagiee* a slouží jako katalog institucí a informačních majáků pro mobilní klienty. Systém disponuje webovým rozhraním pro administraci, skrze které může správce své instituce spravovat - může přidávat / mazat / upravovat informační majáky, může upravovat informace o instituci samotné, ať už se jedná o popis, cenu vstupného či otevírací hodiny. Pro každý informační maják lze definovat vlastní obsah, včetně multimédií.

Každý informační maják má přiřazenou svou „polohu“. Polohou se nemyslí pouze geografické souřadnice - může být navíc doplněna o WiFi či Bluetooth otisk (viz 3.4.1, 3.5) nebo nemusí vůbec geografický údaj obsahovat a může být určena pouze těmito otisky a to nejčastěji v případě identifikace uvnitř budov. Polohy jsou do systému zadávány využitím administrační aplikace *Tagiee Admin*, viz 5.8.

Systém je navržen dostatečně obecně, aby bylo možné co nejširší využití a zároveň, aby bylo možné vytvořit použitelné mobilní klienty jak na platformě Android, tak i na platformách iOS a Windows Phone, jejichž některá omezení při identifikaci informačních majáků jsou uvedeny v kapitole 3.

Podrobný popis systému je k dispozici v diplomové práci Bc. Lud'ka Vlka - *Systém pro správu a realizaci informačních majáků*, v kapitolách 3 a 4.

5.2 Návrh aplikace

V této části bude popsán návrh realizované klientské aplikace - její funkce a uživatelské rozhraní.

5.2.1 Funkce aplikace

Klientská aplikace Tagiee byla navržena tak, aby měla následující funkčnost:

Zobrazení institucí

Po startu bude aplikace zjišťovat aktuální polohu zařízení prostřednictvím lokalizačních služeb. Na základě zjištěné lokace požádá Tagiee systém o vrácení seznamu institucí, které systém Tagiee využívají a které jsou vzdáleny maximálně N kilometrů. Tato konstanta je definována v systému Tagiee. Po vybrání instituce bude zobrazen její detail a uživatel bude moci pokračovat s využíváním aplikace.

V případě, že by uživatel nechtěl nebo nemohl mít zapnuté lokalizační služby, bude mu nabídnuta možnost přihlásit¹ se do instituce zadáním jejího unikátního identifikátoru nazvaného *uuid*. Tento identifikátor si musí uživatel zjistit sám.

Aplikace dále nabídne seznam již navštívených institucí, ke kterým se bude moci uživatel znovu přihlásit.

Přihlášení k institucím bude možné i bez dostupného internetového připojení - viz dále.

Komunikace s Tagiee API

Aplikace bude schopná komunikovat přes Internet s Tagiee systémem skrze API. Komunikace musí probíhat i na pomalém mobilním připojení k Internetu. Příkazy a odpovědi jsou podrobněji popsány v části 5.2.3.

¹nejedná se o přihlášení ve smyslu zadání jména a hesla - přihlášení probíhá automaticky na pozadí, viz 5.2.4

Zobrazení základních informací o instituci

Po zvolení a přihlášení se ke konkrétní instituci aplikace uživateli zobrazí základní informace o instituci, jako je její stručný popis, kontaktní údaje, otevírací hodiny, cena vstupného, poloha na mapě a hodnocení. Dále bude uživateli zobrazen výčet technologií (WiFi, Bluetooth, lokalizační služby), které by měl mít povolené, aby byl schopný aplikaci pro konkrétní instituci plně využít.

V případě dostupného připojení k Internetu se provede kontrola, zda jsou data uložená v zařízení o dané instituci aktuální a případně se provede jejich aktualizace.

Identifikace majáků

Aplikace bude schopná provádět identifikaci majáků využitím následujících metod. Po úspěšné identifikaci zobrazí obsah identifikovaného informačního majáku.

- **poloha:** Aplikace bude zjišťovat co nejpřesnější aktuální polohu uživatele využitím lokalizačních služeb, dále bude zjišťovat okolní WiFi a Bluetooth vysílání. Na základě zjištěných informací se bude aplikace snažit určit a zobrazit nejbližší informační majáky v okolí uživatele. Algoritmus identifikace využitím těchto technologií je popsán v sekci 5.2.5.
- **kód:** Aplikace bude umožňovat uživateli zadat třímístný číselný kód coby podružný jedinečný identifikátor informačního majáku.
- **QR kód:** Aplikace bude umožňovat uživateli naskenovat a vyhodnotit QR kód. K tomuto účelu nebude použita externí čtečka (ve smyslu jiné aplikace), ale skenování QR kódu bude probíhat přímo v klientské aplikaci.
- **NFC:** Aplikace bude umožňovat číst NFC tagy. NFC tag bude typu 2 (viz 3.6.1) a zpráva v něm bude naformátována ve formátu NDEF.
- **Historie:** Aplikace bude umožňovat uživateli nechat si zobrazit seznam již nalezených informačních majáků.

Zobrazení obsahu majáku

Aplikace bude umět zobrazit obsah informačního majáku. Bude se jednat o textové informace a odkazy doplněné o galerii obrázků a streamované video.

Hodnocení instituce

Klient bude umožňovat zobrazit dostupné hodnocení instituce, ke které je uživatel aktuálně přihlášen, a dále mu dovolí danou instituci ohodnotit 1 - 5 hvězdami včetně krátkého slovního hodnocení.

Offline prohlížení

Aplikace dovolí uživateli po přihlášení k instituci stáhnout veškerý její multimediální obsah do vnitřního úložiště zařízení. To umožní uživateli aplikaci aktivně využívat i v případě nedostupnosti internetového připojení a zároveň tato funkčnost šetří mobilní datový tarif, pokud bude v instituci volně přístupná WiFi pro účely stažení offline obsahu. Dále aplikace uživateli dovolí již stažený obsah dané instituce z vnitřní paměti zařízení vymazat.

Odesílání statistik

Klient bude odesílat statistiky využívání celého systému. Bude odesílat statistiky typu

- uživatel se přihlásil do instituce
- uživatel našel informační maják a zobrazil si jeho obsah. V tomto případě bude odesílat i způsob, jak maják našel - tzn. podle polohy, přečtením QR kódu, vložení číselného kódu, přečtením NFC tagu.

Odesílání statistik se musí v případě nedostupnosti internetového připojení pozdržet a provést až při dostupnosti připojení k Internetu.

Best Practices

Aplikace bude dodržovat *best practices* jak pro výdrž baterie, tak pro paměťovou náročnost. Aplikace bude využívat lokalizační služby pouze v případě, že je opravdu potřebuje. Tím se výrazně šetří baterie zařízení.

Dále bude aplikace využívat *holder pattern* pro správnou recyklaci *View* a tím se sníží paměťová náročnost aplikace.

Jedinečná identifikace

Uživatel a jeho zařízení se bude vůči systému Tagiee ověřovat automaticky na základě jedinečného identifikátoru zařízení, viz část 5.2.4.

5.2.2 Návrh UI

Návrh jednotlivých komponent UI a jejich propojení je znázorněno na obrázku 5.1. Silné čáry znázorňují přechody v aplikaci. Slabé čáry jsou znakem vložení prvku UI do stávajícího stavu aplikace.

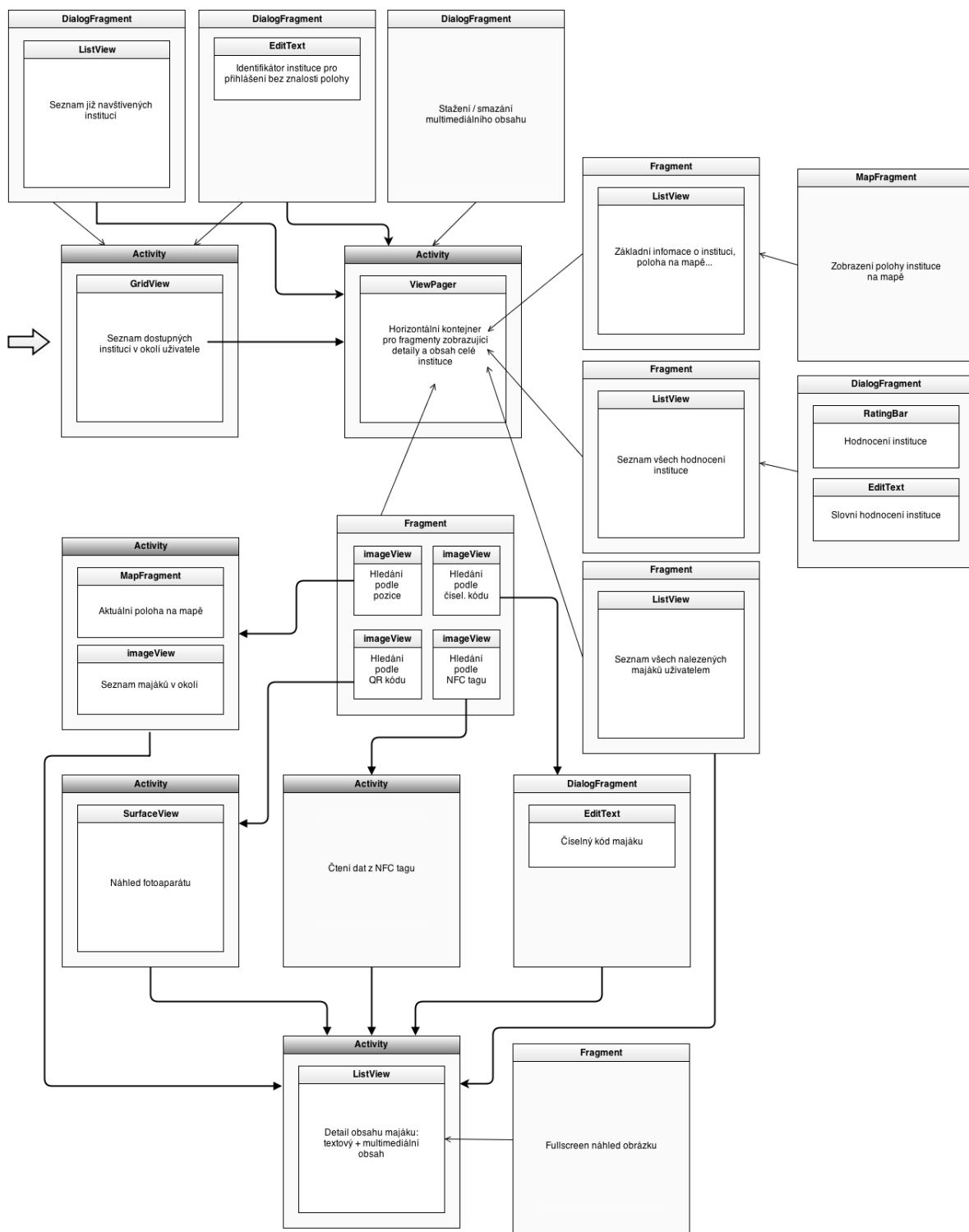
V (v pořadí) druhé aktivitě, která v sobě drží prvek *ViewPager*, jsou vloženy 4 *Fragmenty* horizontálně za sebou. K přepnutí mezi nimi dochází při gestu zvaném *swipe*. Obecně k přechodu na obrázku 5.1 dochází při uživatelském vstupu (př. dotek, stisk klávesy) nebo na základě události (př. přiblížení zařízení k NFC tagu, rozpoznání QR kódu apod.)

5.2.3 Komunikace se systémem Tagiee

V případě dostupnosti internetového připojení bude aplikace komunikovat se serverovou částí systému Tagiee, aby získala aktuální data. Požadavky i odpovědi budou ve formátu JSON. Komunikační rozhraní je na adrese [http:// api.tagiee.com](http://api.tagiee.com).

API systému Tagiee nabízí následující příkazy:

- `add_review` - tento příkaz se naplní jedinečným identifikátorem za-



Obrázek 5.1: Návrh komponent UI a jejich provázání

řízení, počtem hvězd, který uživatel instituci dal, a krátkým slovním ohodnocením.

- `beacon_found` - tento příkaz slouží k odesílání informace, že zařízení objevilo a zobrazilo konkrétní maják. Součástí je jedinečný identifikátor zařízení, identifikátor majáku, který byl nalezen, a způsob jakým byl nalezen.
- `device_visit_institution` - tento příkaz slouží k odeslání informace o úspěšném přihlášení uživatele k instituci. Příkaz obsahuje identifikátor zařízení uživatele a identifikátor instituce.
- `get_institution_by_uuid` - odesláním tohoto požadavku chce klient získat kompletní informace o dané instituci určené jejím identifikátorem. Odpověď v případě úspěchu obsahuje kompletní informace o instituci, metadata o všech informačních majících a popis veškerého multimediálního obsahu.
- `get_institution_list` - tento požadavek vrací jako odpověď seznam institucí využívající systém Tagiee v předem definované maximální vzdálenosti. Jako parametry požadavku jsou odesílány souřadnice aktuální polohy.
- `get_institution_revision` - jednoduchý požadavek na zjištění hodnoty čísla revize dat konkrétní instituce.
- `get_review` - odpovědí na tento požadavek klient získá seznam hodnocení dané instituce. Seznam může být kompletní či omezen shora nebo zdola.

Kompletní ukázka jednotlivých požadavků a odpovědí složených ve formátu JSON je uveden v příloze A.3.

5.2.4 Jedinečná identifikace zařízení

Pro systém Tagiee je kritické, aby bylo možné jednoznačně a unikátně identifikovat každé mobilní zařízení, které bude k systému přistupovat. Jelikož je systém navržen tak, aby k němu mohlo přistupovat více platforem (Android, iOS, Windows Phone) bylo nutné tuto identifikaci zobecnit.

K tomu lze využít knihovnu *openUDID*, která dokáže generovat jedinečný identifikátor pro každé zařízení i napříč platformami. Na platformě Android vytváří knihovna 32 znaků dlouhý hash, který je složen z MAC adresy WiFi rozhraní, MAC adresy Bluetooth rozhraní, IMEI a Android ID, což je 64-bitové číslo, které je náhodně vygenerované při prvním startu zařízení a zůstává stejné až do provedení obnovy do továrního nastavení.

Ukázka použití

Knihovna je pro platformu Android dostupná na adrese <https://github.com/jasonlamkk/OpenUDID>. Stačí stáhnout a do projektu vložit jedinou Java třídu `openUDID.java`.

Pro co nejlepší využití s minimální šancí duplicitních identifikátorů je vhodné do manifestu aplikace vložit následující oprávnění, jež dovolí zjistit MAC adresy WiFi a Bluetooth a IMEI telefonu:

```
1 <uses-permission android:name="android.permission.  
    ACCESS_WIFI_STATE"/>  
2 <uses-permission android:name="android.permission.BLUETOOTH"/  
    >  
3 <uses-permission android:name="android.permission.  
    READ_PHONE_STATE"/>
```

V samotné aplikaci se zjištění provádí následující sekvencí příkazů.

```
1 //staci provest pouze jednou za beh aplikace  
2 OpenUDID.syncContext(getApplicationContext());  
3  
4 //takto se ziska presistentni jedinecny identifikator pro  
    dane zarizeni v ramci domeny - com.tagiee  
5 String mUdid = OpenUDID.getCorpUDID("com.tagiee");
```

5.2.5 Algoritmus hledání polohy

V této sekci bude popsán algoritmus určení polohy uživatele a nejbližších majáků na základě okolního WiFi a Bluetooth vysílání a geografických souřadnic. Algoritmus vychází ze stavu, kdy má aplikace přehled o všech informačních majících instituce - tzn. zná buď jejich geografickou polohu nebo WiFi otisk či Bluetooth otisk, popřípadě jejich kombinaci.

Výběr metody

Jak pro WiFi, tak pro Bluetooth byla vybrána metoda otisků. U technologie Bluetooth nebyla kvůli omezeným možnostem Android SDK jiná volba, avšak u WiFi byla na výběr buď metoda otisků či metoda triangulace (viz sekce 3.4). Z těchto dvou metod byla zvolena obecnější metoda otisků. Zvolena byla právě z důvodu její obecnosti, jelikož i celý systém Tagiee je navrhován co nejvíce obecně. Dalším důvodem byla skutečnost, že metoda triangulace se využívá mnohem více ve spojení s mapovými podklady, což by vyžadovalo povinnost pro každou instituci dodat do systému plány jednotlivých pater (v případě, že bude mít informační majáky uvnitř budov).

Skenování okolí

Na začátku zjišťování polohy se spustí vlákno, které provádí průběžné skenování okolí a hledá viditelné WiFi a Bluetooth vysílače². Zároveň se aplikace zaregistruje pro odběr aktualizací polohy.

Proces skenování WiFi a Bluetooth zařízení je zcela odlišný. U WiFi je po zavolání funkce `getScanResults()` k dispozici celý seznam všech okolních WiFi sítí. Při zjišťování okolních Bluetooth zařízení se aplikace musí přihlásit k odběru *broadcast* zpráv, kdy při každém nově objeveném zařízení přichází jedna zpráva obsahující informaci o něm. Ukázky jsou v následujících dvou výpisech kódu.

```
1 //ziskani spravce WiFi
2 WifiManager mWifiManager = (WifiManager) context.
   getSystemService(Activity.WIFI_SERVICE);
3
4 if (mWifiManager.isWifiEnabled()) {
5     if (mWifiManager.startScan()) {
6         //ziskani vysledku skenovani WiFi siti
7         List<ScanResult> scanResultList = mWifiManager.
           getScanResults();
```

```
1 //ziskani rozhrani pro praci s Bluetooth
2 BluetoothAdapter mBtAdapter = BluetoothAdapter.
   getDefaultAdapter();
3
4 if (mBtAdapter != null) {
```

²pouze v případě, že má uživatel tyto technologie v zařízení dostupné a aktivní

```
5 //registrace odberu broadcast zprav
6 IntentFilter filter = new IntentFilter(BluetoothDevice.
    ACTION_FOUND);
7 context.registerReceiver(mReceiver, filter);
8
9 if (mBtAdapter.isEnabled()) {
10     if (!mBtAdapter.isDiscovering())
11         mBtAdapter.startDiscovery();
12     //sken trva 12 a vice vterin
13 }
14 }
15
16 ...
17
18 //definice prijimace bcast zprav s novym BT zarizenim
19 mReceiver = new BroadcastReceiver() {
20     public void onReceive(Context context, Intent intent) {
21         //zpracovani dat v intentu, v pripade ukonceni skenovani
            spustit znovu
22     }
23 }
```

Párování

Jakmile přijde informace o nové lokaci nebo nový seznam dostupných WiFi vysílačů či nově objevené Bluetooth zařízení, spustí se nové „párovací“ vlákno, pokud již neběží. Toto vlákno se snaží postupně porovnat pozice a *otisky* všech majáků v instituci s aktuálním stavem.

Pro každý průběh párování dostane každý maják *hodnotu páru* = 0. V případě, že má maják uložen údaj o jeho geografické poloze a aktuální vzdálenost uživatele od majáku je menší než 20 metrů, přičte se mu jeden bod k *hodnotě páru*.

V případě, že je maják definován WiFi otiskem, zjistí se nejprve relativní hodnota shody. To v praxi znamená, že se vezme velikost WiFi otisku - seznam BSSID seřazený sestupně dle síly signálu - a délkou tohoto seznamu se vydělí číslo 100. Tím získáme hodnotu, kterou nazýváme *relativní hodnota shody*. Po zjištění relativní hodnoty shody se provádí porovnávání s aktuálním stavem WiFi okolí. Za každou shodu prvků ve WiFi otisku majáku a v aktuálním seznamu WiFi vysílačů se přičte k celkové míře shody *relativní hodnota shody*. Pokud navíc souhlasí i pořadí v seznamu, přičítá se ještě malý bonus v podobě 5 bodů.

Pro Bluetooth otisk je situace analogická, zde se ovšem hledá pouze shoda

v Bluetooth otisku majáku a aktuální seznamu vysílajících Bluetooth zařízení. Jelikož Android programově neumožňuje zjistit sílu signálu, pořadí Bluetooth zařízení v otisku nehraje žádnou roli.

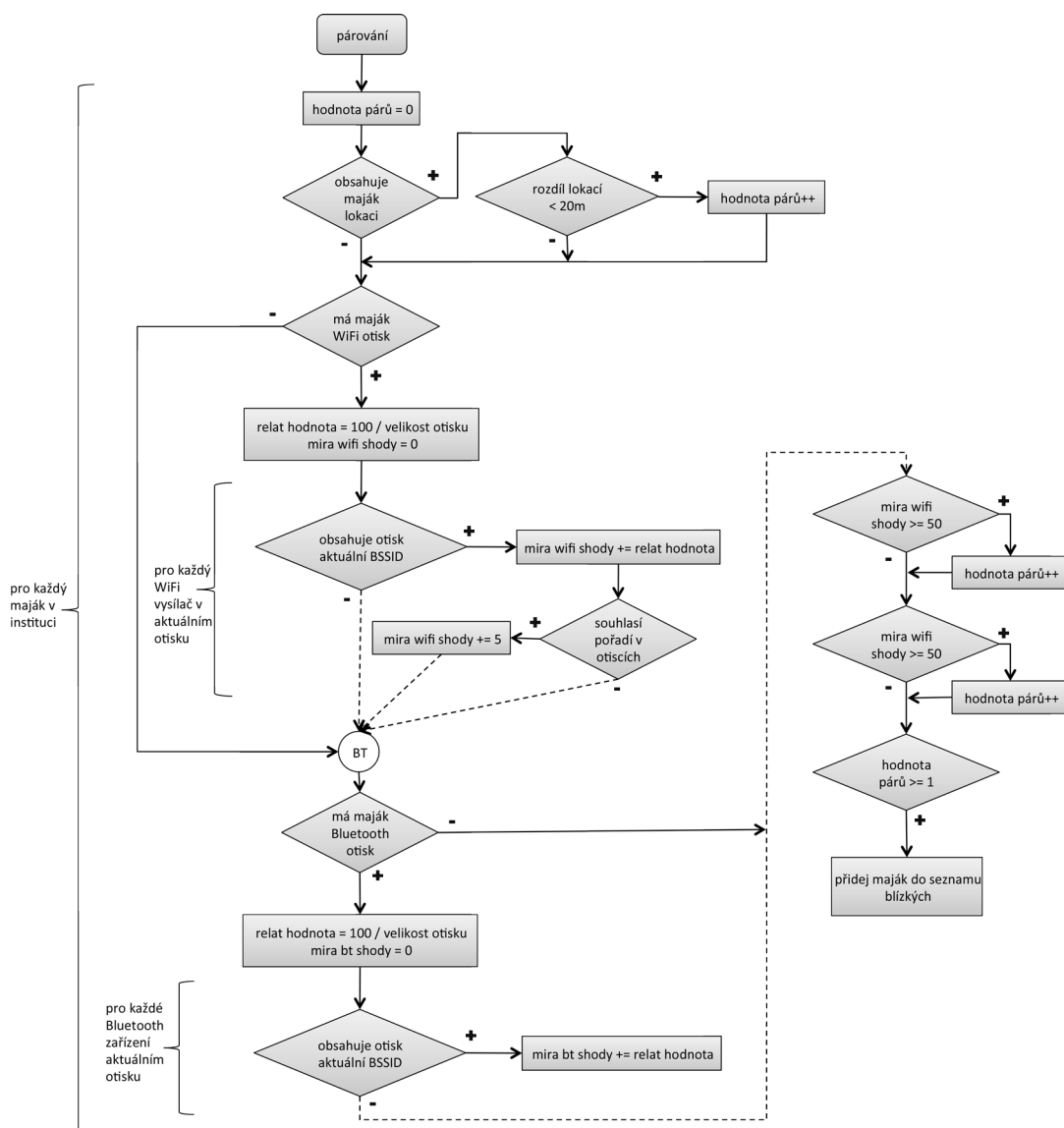
Jak pro WiFi, tak pro Bluetooth se na závěr provede vyhodnocení. Pokud celková míra shody přesáhne 49%, přičítá se k aktuální *hodnotě páru* daného majáku 1 bod. Všechny majáky instituce, které mají *hodnotu páru* větší než 0, jsou považovány za majáky vyskytující se v okolí uživatele. Čím větší je *hodnota páru*, tím je větší pravděpodobnost, že je maják v blízkosti uživatele. Aplikace vrací seznam okolních majáků seřazený právě podle této *hodnoty páru*.

Tento postup je postupně prováděn pro každý maják instituce. Graficky je znázorněn na obrázku 5.2.

5.2.6 Odůvodnění oprávnění aplikace

Aplikace pro správné fungování vyžaduje při instalaci 14 oprávnění. Následují jednotlivé odůvodnění, proč je dané oprávnění vyžadováno.

- `com.tagiee.permission.MAPS_RECEIVE`
- `com.google.android.providers.gsf.permission.READ_GSERVICES` - toto oprávnění včetně předchozího je třeba, aby aplikace mohla komunikovat s Google Maps Android API v2 a naopak.
- `android.permission.INTERNET` - umožňuje aplikaci připojit se k Internetu. To je třeba pro komunikaci s Tagiee serverem.
- `android.permission.WRITE_EXTERNAL_STORAGE` - povoluje aplikaci zapisovat na externí úložiště (nejčastěji SD karta). To je potřeba pro ukládání offline obsahu.
- `android.permission.ACCESS_COARSE_LOCATION`
- `android.permission.ACCESS_FINE_LOCATION` - toto i předchozí oprávnění je zapotřebí pro zjištění lokalizace zařízení.
- `android.permission.ACCESS_WIFI_STATE` - umožňuje pracovat s WiFi rozhraním, např. zjišťovat informace o okolních WiFi sítích.
- `android.permission.NFC` - je potřeba pro čtení NFC tagů.



Obrázek 5.2: Algoritmus hledání majáků

- `android.permission.ACCESS_NETWORK_STATE` - toto povolení je potřeba pro zjišťování dostupnosti připojení k Internetu před zahájením komunikace s Tagiee serverem.
- `android.permission.BLUETOOTH`
- `android.permission.BLUETOOTH_ADMIN` - toto i předchozí oprávnění je využíváno při zjišťování okolních Bluetooth zařízení.

- `android.permission.READ_PHONE_STATE` - je zapotřebí, aby aplikace mohla zobrazit výchozí dialer s telefonním číslem instituce.
- `android.permission.CAMERA` - umožní aplikaci používat fotoaparát při čtení QR kódu.
- `android.permission.WAKE_LOCK` - je využito, aby se zařízení neuspalo při vyhledávání okolních majáků.

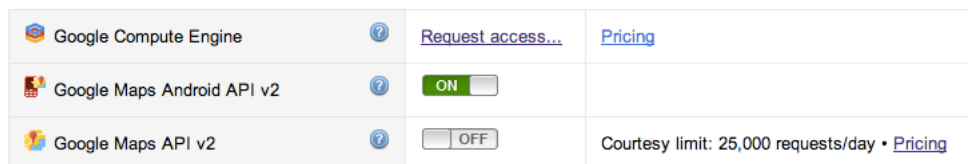
5.3 Google apis







Společnost Google nabízí přes 50 svých služeb, které nabízí volně nebo za úplatu vývojářům celého světa. V klientské aplikaci Tagiee budou využity 2 z nabízených služeb a to *Google Maps Android API v2* pro zobrazování map a *YouTube Data API v3* pro zobrazení streamovaného videa jako část obsahu informačního majáku.

Pro využívání výše uvedených služeb je potřeba se vůči Google apis ověřovat. K tomu je potřeba si buď vytvořit OAuth token nebo API klíč. Pokud nechceme přistupovat přes Google apis k uživatelským datům, stačí využívat API klíč, který slouží pouze jako identifikace projektu vůči Google apis.

Vytvoření API klíče

Kritické pro využívání Google apis je nutnost mít Google účet. Po přihlášení do Google účtu je na stránce <https://code.google.com/apis/console/> vývojářská konzole. V levém roletovém menu je nutné si vytvořit projekt. Po vytvoření projektu je v levém menu položka *Services*, pod kterou se skrývají všechny služby, které Google nabízí. Jejich zapínání / vypínání je vidět na obrázku 5.3.



 Google Compute Engine		Request access...	Pricing
 Google Maps Android API v2		<input checked="" type="checkbox"/>	
 Google Maps API v2		<input type="checkbox"/>	Courtesy limit: 25,000 requests/day • Pricing

Obrázek 5.3: Ukázka zapnutí / vypnutí služeb Google.

Posledním krokem ve vývojářské konzoli je třeba vytvořit již zmiňovaný API klíč. Využívání API klíče pro Android je možné omezit pouze na název balíku aplikace a *SHA-1 otisk* certifikátu vývojáře. V menu *API Access* na obrázku 5.4 je vidět API klíč pro přístup Android aplikace k Google apis. Z pochopitelných důvodů jsou důvěrné informace nečitelné. V prvním řádku je 39-ti místný API klíč. Jeho použití bude ukázáno v následujících sekcích 5.3.1 a 5.3.2. V druhém řádku je omezení API klíče - je zde *SHA-1 otisk* certifikátu vývojáře a jméno balíku aplikace, která smí tento klíč využívat pro přístup k Google službám.

```

Key for Android apps (with certificates)
API key:          [redacted]
Android apps:    [redacted]:com.tagiee
Activated on:     Feb 27, 2013 5:37 AM
Activated by:     hrom.jiri@gmail.com - you

```

Obrázek 5.4: API klíče pro Android aplikaci.

5.3.1 Google Maps Android API v2

Pro využívání této služby v Android aplikaci je třeba upravit manifest aplikace a vložit do něj výše vytvořený API klíč. Ten se vkládá přidáním následujícího kódu do sekce `<application>`:

```

1 <meta-data android:name="com.google.android.maps.v2.API_KEY"
   android:value="<--api_klic-->" />

```

Dále je ještě třeba aplikaci přidat příslušná oprávnění, aby mohla mapy od společnosti Google využívat:

```

1 <permission android:name="com.tagiee.permission.MAPS_RECEIVE"
   android:protectionLevel="signature" />
2 <uses-permission android:name="com.tagiee.permission.
   MAPS_RECEIVE" />
3 <uses-permission android:name="com.google.android.providers.
   gsf.permission.READ_GSERVICES" />
4 <!--pro načítání mapových podkladů z Internetu-->
5 <uses-permission android:name="android.permission.INTERNET"/>
6 <!--nactene mapy se ukladaji do vnitřního uloziste-->
7 <uses-permission android:name="android.permission.
   WRITE_EXTERNAL_STORAGE"/>
8 <!--pokud se mapy používají i pro navigaci uživatelé-->
9 <uses-permission android:name="android.permission.
   ACCESS_COARSE_LOCATION"/>

```

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

Google maps pro Android od verze 2 vyžadují OpenGL ES 2, které se povolí přidáním následujícího kódu do manifestu aplikace.

```
1 <uses-feature android:glEsVersion="0x00020000"  
android:required="true" />
```

Samotné mapy se do aplikace dají vložit do *View* jako samostatné *MapView* či jako *fragment* vložením třídy `com.google.android.gms.maps.MapFragment` nebo `com.google.android.gms.maps.SupportMapFragment` v případě, že je využita *Support Library*. Jak přidávat fragmenty do aplikace je vysvětleno v části 4.3

Jak již bylo zmíněno výše, všechny Google služby jsou buď zdarma nebo je jejich použití omezeno určitými limity, které lze za úplatu zvětšit. U Google map pro Android je použití zcela zdarma a bez jakýchkoliv limitů.

5.3.2 YouTube Data API v3

YouTube Data API v3 je experimentální verze, která je stále ve vývoji³, nicméně se předpokládá její brzké dokončení a minimální změny v API. Součástí je také YouTube Android Player API, který nabízí více možností, jak vložit přehrávač YouTube videí do aplikace.

Ověření opět probíhá výše vytvořeným API klíčem. Zde se ovšem API klíč nepřidává do manifestu aplikace ale přímo do programového kódu. Přehrávač se do aplikace dá opět přidat více způsoby - jako *View*, *Fragment* nebo se dá vyvolat zabudovaná samostatná aplikace pro YouTube videa pomocí *Intentu*.

V klientské aplikaci Tagiee je přehrávač vložen jako *view*. Nevýhodou je, že aktivita, která v sobě zobrazuje přehrávač (nebo má v sobě fragment, který zobrazuje přehrávač) musí dědit od třídy `YouTubeBaseActivity`, což nemusí být vždy vyhovující. Dalším požadavkem, který už není problémový, je nutnost implementovat rozhraní `YouTubePlayer.OnInitializedListener`. Použití je naznačeno na následující ukázce.

³údaj z dubna 2013

```
1 public static final String DEVELOPER_KEY = api_klic;
2
3 //vytvoreni view, vyhledani a vlozeni com.google.android.
  youtube.player.YouTubePlayerView
4 ...
5 //druhy parametr je trida implementujici
  OnInitializedListener rozhrani
6 youtubeView.initialize(DEVELOPER_KEY, this);
```

Rozhraní `OnInitializedListener` vyžaduje implementaci následujících dvou metod. Každé YouTube video je určeno svým jedinečným identifikátorem, který se využívá právě při inicializaci.

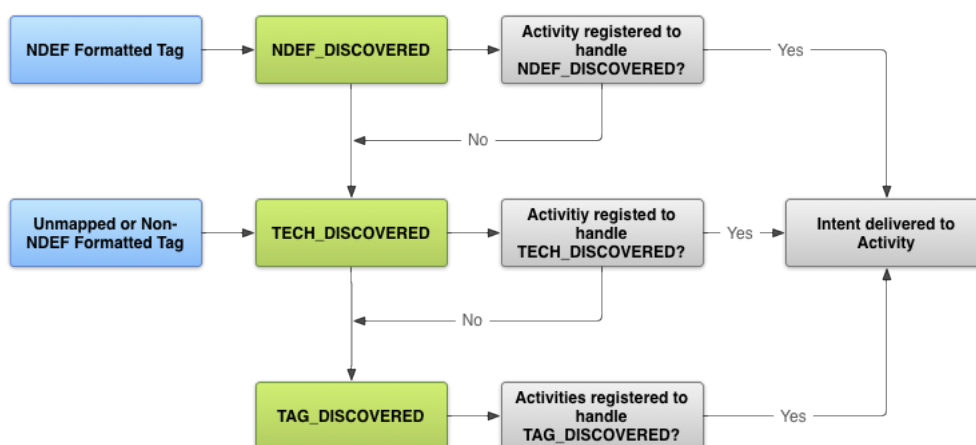
```
1 @Override
2 public void onInitializationFailure(Provider arg0,
  YouTubeInitializationResult arg1) {
3 //vyresit neuspesnou inicializaci
4 }
5
6 @Override
7 public void onInitializationSuccess(YouTubePlayer.Provider
  provider, YouTubePlayer player, boolean wasRestored) {
8     if (!wasRestored) {
9         player.cueVideo(id_vida);
10    }
11 }
```

Oproti Google mapám pro Android není použití YouTube Data API v3 zcela neomezené. Vývojář má v základu k dispozici 5 000 000 jednotek na den, kde si může nastavit jak velkou část přiřadí jednomu uživateli. V základu jsou jednotky rozděleny po 30 000 jednotek / uživatel, kde 1 jednotka odpovídá 1 vteřině videa.

5.4 Použití NFC

Prvním předpokladem pro využití NFC je zařízení, které tuto technologii podporuje. Druhým předpokladem je příslušné oprávnění, aby zařízení mohlo NFC využívat.

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

Obrázek 5.5: Rozhodování *Tag Dispatch Systemu*. [and(d)]

5.4.1 The Tag Dispatch System

V návrhu aplikace je dáno, že aplikace bude pouze data číst a to konkrétně ve formátu *NDEF*. Jak na NFC tagy zapisovat je ukázáno v sekci 5.8. Komunikaci s ostatními NFC zařízeními (ať už pasivními nebo aktivními) má na starosti tzv. *The Tag Dispatch System*, dále jen Dispatch System. Ten postupně hledá zaregistrované aktivity od „nejspecifičtějšího“ filtru až po ten nejméně obecný. Celé jeho rozhodovací schéma je znázorněno na obrázku 5.5.

V případě, že je zařízení v blízkosti NFC tagu, který obsahuje NDEF obsah, hledá Dispatch System aktivity, které jsou v systému zaregistrovány na filtr `NDEF_DISCOVERED`. V případě neúspěchu zkouší vyhledat aktivity, které registrují filtr `TECH_DISCOVERED` a jim *Intent* doručit. Pokud ani zde není nalezena aktivita, která by dokázala *Intent* přijmout a obsloužit, vyhledává všechny aktivity, které přijímají *Intenty* s filtrem `TAG_DISCOVERED`. Při jakémkoliv úspěchu během hledání aktivity schopné obsloužit *Intent*, je další hledání zastaveno a je jí doručen. V případě, že takových aktivit je v systému více, je zvolena buď výchozí nebo je uživateli dáno na výběr, aby zvolil ji on. V opačném případě, kdy v systému neexistuje jediná aktivita schopná obsloužit příchozí *Intent*, se neprovede žádná akce a *Intent* bude ignorován.

Obecně lze aplikaci pro komunikaci s NFC zaregistrovat v systému dvěma způsoby. Ten první je definicí *intent-filteru* pro danou aktivitu v manifestu aplikace a je vidět na následujícím výpisu kódu. Jedná se nejjednodušší

definici. Filtru lze samozřejmě nastavit technologie tagů, na které má reagovat apod.

```
1 <intent-filter>
2   <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
3   <data android:mimeType="text/plain" />
4 </intent-filter>
```

5.4.2 Foreground Dispatch System

Druhým způsobem je tzv. *Foreground Dispatch System*. Jedná se v podstatě o dynamickou registraci aktivity pro komunikaci s NFC tagem, která má zároveň nejvyšší prioritu. Tento systém může používat pouze aktivita, která je v danou chvíli aktivní, tudíž její přednost před ostatními aktivitami je opravdu absolutní. Použití je rámcově znázorněno v níže uvedeném výpisu kódu.

```
1 //ziskani objektu pro praci s nfc
2 NfcAdapter mAdapter = NfcAdapter.getDefaultAdapter(this);
3 //pripaveny intent k naplneni
4 PendingIntent mPendingIntent = PendingIntent.getActivity(this
5     , 0, new Intent(this, getClass()).addFlags(Intent.
6     FLAG_ACTIVITY_SINGLE_TOP), 0);
7
8 //datovy filtr urcujici na co vse ma aktivita reagovat
9 IntentFilter ndef = new IntentFilter(NfcAdapter.
10     ACTION_NDEF_DISCOVERED);
11 try {
12     ndef.addDataType("/*/*");
13 }
14 catch (MalformedMimeTypeException e) {
15     Log.e(Constants.LOG_TAG, e.getMessage());
16 }
17
18 mFilters = new IntentFilter[] { ndef };
19 //filtr technologii na NfcA (odpovida Mifare Classic 1k)
20 mTechLists = new String[][] { new String[] { NfcA.class.
21     getName() } };
```

Registrace resp. odregistrace by se měla provádět v příslušných metodách životního cyklu aktivity - tedy např. `onResume()` resp. `onPause()`. První parametr metody je odkaz na aktivitu, která má zájem o NFC komunikaci.

```
1 //registrace
2 mAdapter.enableForegroundDispatch(this, mPendingIntent,
3     mFilters, mTechLists);
```

```

3 //odregistrace
4 mAdapter.disableForegroundDispatch(this);

```

Pokud je aktivita úspěšně zaregistrovaná v *Foreground Dispatch Systemu* a správný NFC tag je k zařízení přiblížen, naplní se definovaný `mPendingIntent` a vyvolá se metoda `onNewIntent`, kterou je třeba v aktivitě překrýt.

```

1 @Override
2 protected void onNewIntent(Intent intent) {
3     super.onNewIntent(intent);
4     //takto lze z intentu dostat data z NFC tagu
5     Tag tag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
6     //libovolne zpracovani dat
7     ...
8     return;
9 }

```

5.4.3 NDEF formát

Pokud je v NFC tagu uložena zpráva ve formátu NDEF, lze její obsah definovat 3 bity označenými jako TNF. Ze všech podporovaných TNF je pro klientskou aplikaci nejzajímavější hodnota `TNF_WELL_KNOWN`, která říká, že obsah NFC tagu je buď URI nebo MIME typ, který je ještě dále určen tzv. RTD. V tabulce 5.1 jsou uvedeny vybrané RTD typy, z nichž některé jsou použity v klientské aplikaci. Kompletní přehled TNF a RTD hodnot je dostupný na stránce <http://developer.android.com/guide/topics/connectivity/nfc/nfc.html>.

RTD	mapování
<code>RTD_SMART_POSTER</code>	URI podle parsování obsahu tagu
<code>RTD_TEXT</code>	MIME typ <code>text/plain</code>
<code>RTD_URI</code>	URI podle obsahu tagu

Tabulka 5.1: Přehled některých podporovaných RTD

5.5 Implementace QR čtečky

Jelikož je mezi zvolenými technologiemi QR kód, bylo třeba do klientské aplikace implementovat čtečku, která dokáže QR kódy dekodovat. V sekci 3.3

jsou popsány vybrané knihovny, které mají požadovanou funkčnost. Pro implementaci byla vybrána knihovna **Zbar** z důvodu její snadné implementace a velmi rychlému dekódování QR kódů.

Knihovnu je třeba zahrnout do projektu aplikace a dekódování QR obrazce lze vyvolat pouhým *Intentem*. Není třeba, aby měl uživatel v zařízení nainstalovanou žádnou aplikaci 3. strany na skenování QR kódů.

```
1 if (isCameraAvailable()) {
2     //pokud je kamera dostupna, lze skenovat QR kod
3     Intent intent = new Intent(getActivity(),
4         ZBarScannerActivity.class);
5     startActivityForResult(intent, ZBAR_SCANNER_REQUEST);
6 }
```

Jelikož byla aktivita vyvolána metodou `startActivityForResult(...)`, tak po úspěšném skenování QR kódu se vrátí zpět do volající aktivity s výsledkem obsaženým v *Intentu*.

```
1 @Override
2 public void onActivityResult(int requestCode, int resultCode,
3     Intent data) {
4     switch (requestCode) {
5     case ZBAR_SCANNER_REQUEST:
6     case ZBAR_QR_SCANNER_REQUEST:
7         //vysledek QR ctecky
8         if (resultCode == Activity.RESULT_OK) {
9             //dekodovany obsah QR kodu
10            String qrMessage = data.getStringExtra(ZBarConstants.
11                SCAN_RESULT);
12            //libovolne zpracovani
13            ...
14        }
15    }
```

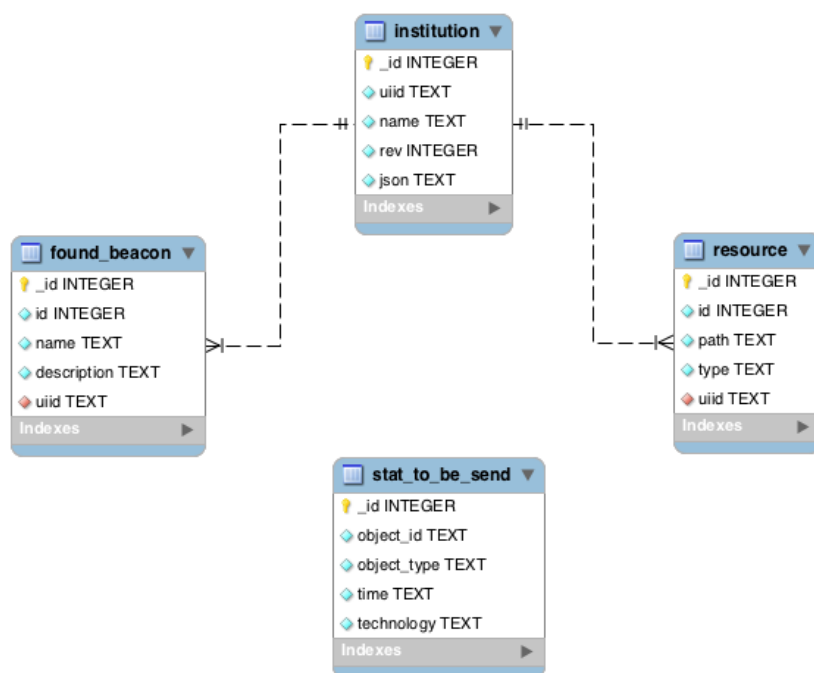
5.6 Databáze

V aplikaci byla použita databáze pro persistenci dat. Na obrázku 5.6 je vidět datový model databáze. Model není nijak zvlášť komplexní, nicméně je pro klientskou aplikaci vyhovující. Sloupce `_id` jsou vnitřní identifikátory *SQLite* databáze na platformě Android. Případné sloupce `id` jsou identifikátory ze systému Tagiee.

Tabulky `found_beacon` a `resource` jsou propojeny s tabulkou `institu-`

`tion` prostřednictvím cizího klíče. Tabulka `resource` udržuje všechny informace o multimediálních souborech dané instituce. Tabulka `found_beacon` v sobě udržuje již navštívené majáky v rámci dané instituce.

Tabulka `stats_to_be_send` je využita pouze pro ukládání statistických dat pro server, která nemohla být odeslána z důvodu nedostupnosti internetového připojení.



Obrázek 5.6: Datový model databáze

5.7 Verze pro tablety

V rámci Oborového projektu byla vytvořena verze klientské aplikace Tagiee pro zařízení s větším displejem tzn. tablety apod. Stavební prvky aplikace - již zmíněné *Fragmenty* - jsou pro tyto účely ideálním prostředkem. V případě, že jsou navrženy a vytvořeny jako znovupoužitelné, lze je poskládat do nového rozložení, které více vyhovuje větším zařízením při zachování stejné funkčnosti. Dalším prostředkem jsou *konfigurační kvalifikátory*, o kterých je uvedeno více informací v následující části.

5.7.1 Podpora různých obrazovek

Ačkoliv platforma Android nabízí některé prvky UI, které se umí automaticky přizpůsobit změnám obrazovky, neposkytují tyto často nejlepší rozhraní pro každou velikost obrazovky. Z tohoto důvodu se využívají již zmíněné *konfigurační kvalifikátory*, které umožňují automaticky při běhu aplikace zvolit příslušné zdroje pro aktuální konfiguraci.

Pro definici rozložení prvků UI se používají XML soubory - tzn. *layout*. Standardní umístění souborů je v adresáři `/res/layout/`, kde jsou uloženy všechny obecné XML konfigurační soubory. Konfigurační soubory pro konkrétní konfiguraci se umísťují do adresářů, které mají ve svém názvu konfigurační kvalifikátory oddělené pomlčkou - tzn. pro zařízení s větší obrazovkou⁴ systém hledá konfigurační XML soubory například na cestě `/res/layout-large/`.

Tento kvalifikátor pro velkou obrazovku je nicméně příliš obecný a nedostačující. Proto od verze 3.2 nabízí Android mnohem jemnější kvalifikátory (viz část 4.1.4), nicméně pro zařízení s verzí systému nižší než 3.2 jsou tyto kvalifikátory neznámé a tak pro zpětnou kompatibilitu by se musely udržovat XML soubory s rozložením UI prvků jak pro obecné, tak pro jemné kvalifikátory. Z toho důvodu byly zavedeny *aliasy*.

Příklad použití

Klientská aplikace používá standardně jedno-panelové rozhraní, přičemž pro tablety je dostatek prostoru pro použití dvou-panelového rozhraní. S využitím *aliasů* stačí vytvořit nový XML soubor definující rozložení UI prvků pro větší zařízení, který se uloží na standardní umístění (`/res/layout/`). Situace je tedy následující:

- `fragment_found_beacons.xml` pro jedno-panelové rozložení
- `fragment_found_beacons_tablet.xml` pro dvou-panelové rozložení

Dalším krokem je vytvoření aliasů. Ty mohou být v libovolně pojmenovaném XML souboru, který musí být uložen ve správném adresáři dle konfiguračních kvalifikátorů. Je tedy pořád nutné vytvořit dva XML soubory,

⁴7ⁿa více

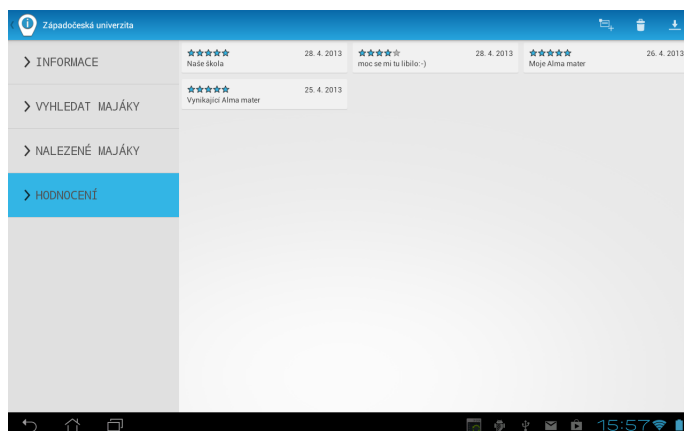
jak pro obecný kvalifikátor (`large`), tak pro nový jemnější (např. `sw600dp`), avšak stačí udržovat pouze jeden *layout* XML soubor pro větší zařízení, což je pro další vývoj výhodnější. Následující ukázka kódu je z XML souboru uloženého v adresáři `/res/layout-large/` resp. `/res/layout-sw600dp`. Je zde vidět mapování souboru `fragment_found_beacons_tablet` na `fragment_found_beacons`, přičemž obsah obou souborů je odlišný.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <item name="fragment_found_beacons" type="layout">@layout
4     /fragment_found_beacons_tablet</item>
5 </resources>
```

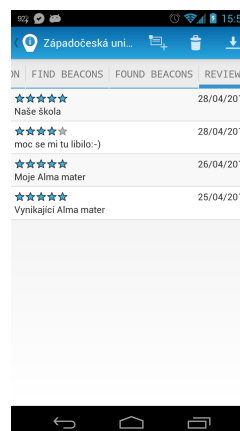
V kódu aplikace poté stačí při vytváření UI otestovat, které UI prvky jsou dostupné a zjistit tak aktuální konfiguraci zařízení. V následující ukázce kódu je vidět vytvoření uživatelského rozhraní ze zdroje pojmenovaného `fragment_found_beacons`, což bude systém díky vytvořeným aliasům načítat z odlišných XML souborů dle aktuální konfigurace.

```
1 @Override
2 public View onCreateView(LayoutInflater inflater, ViewGroup
3   container, Bundle savedInstanceState) {
4   //vytvoreni View ze zdroje, který je definovan aliasem
5   View root = inflater.inflate(R.layout.
6     fragment_found_beacons, container, false);
7   //obsahuje layout pro telefony
8   mListView = (ListView) root.findViewById(R.id.
9     foundBeaconsListView);
10  //obsahuje layout pro tablety
11  mGridView = (GridView) root.findViewById(R.id.
12     foundBeaconsGridView);
13  if (mListView != null) {
14    //jedno-panelovy layout
15  }
16  if (mGridView != null) {
17    //dvou-panelovy layout
18  }
19  return root;
20 }
```

Na obrázku 5.7 je zobrazen příklad různého rozložení UI v závislosti, na konfiguraci zařízení. Je patrné, že pro tablety je použito dvou-panelové rozhraní, kde jsou navíc jednotlivá hodnocení zobrazena ve 3 sloupcích. Pro menší zařízení jsou jednotlivá hodnocení zobrazena přes celou šířku displeje v jednom sloupci a celá navigace se změnila z vertikální na horizontální. Je zde



(a) Zobrazení hodnocení dané instituce na tabletu.



(b) Zobrazení hodnocení dané instituce na telefonu.

Obrázek 5.7: Ukázka aplikace pro různé konfigurace zařízení

také vidět, že si aplikace poradí s různě nastaveným jazykovým prostředím v zařízení. Na tabletu je nastaveno kompletně české prostředí a na telefonu je anglické prostředí ovšem s českým formátem času a data.

5.8 Tagiee Admin

V rámci předmětu *KIV/MKZ* byla vytvořena administrační aplikace pro serverovou část systému Tagiee. Účelem aplikace je dát správci institucí mobilní nástroj pro zobrazení rychlého přehledu statistik, hodnocení, přidávání nových lokací a přípravu NFC tagů pro jednotlivé majáky.

5.8.1 Funkce aplikace

Administrátorská aplikace byla navržena, aby poskytovala následující funkčnost:

Přihlášení

Každý administrátor má v systému Tagiee svoje uživatelské jméno a heslo, pod kterým může spravovat jím vytvořené instituce. Zadáním svého jména

a hesla, které je shodné s údaji ve webové administraci, se administrátor přihlásí a jsou mu k dispozici následující funkce pro jeho instituce. Následující funkce jsou k dispozici **pouze** po úspěšném přihlášení administrátora

Přehled institucí

Aplikace nabídne administrátorovi přehled všech jeho institucí, jejich hodnocení a stručné statistiky. Jedná se o statistiky typu celkový počet návštěvníků, počet návštěvníků za měsíc. Dalším statistickým údajem bude poměr platformem zařízení, které danou instituci navštívila.

Přehled hodnocení

Aplikace umožní administrátorovi zobrazit všechna uložená hodnocení pro konkrétní instituci seřazená sestupně podle času přidání.

Zobrazení majáků a statistik

Administrační klient umožní uživateli zobrazit seznam všech aktivních i neaktivních majáků v dané instituci. U každého majáku budou statistické údaje o tom, kolikrát byl maják objeven a jakou metodou / technologií byl objeven. Poslední údaj bude zobrazen v podobě koláčového grafu.

Příprava NFC tagu

Aplikace dále umožní pro vybraný maják vytvořit NFC tag - resp. zapsat na něj potřebné údaje. Uživatel bude mít na výběr, zda chce NFC tag trvale zamknout proti přepsání a zda chce přidat do tagu volitelný údaj - stručný popis majáku - pro univerzální NFC čtečky.

Uložení lokace

Klient umožní uživateli přidávat k instituci nové lokace, které jdou přidávat jednotlivým majákům. Lokace je určena jménem a libovolnou kombinací

aktuální polohy, seznamem okolních vysílajících WiFi zařízení seřazených sešupně dle síly signálu a seznamem okolních vysílajících Bluetooth zařízení.

5.8.2 Příprava NFC tagu

Předpoklady a základy pro komunikaci s ostatními NFC zařízeními jsou uvedeny v části 5.4. V této části bude pouze stručně popsán způsob zápisu *NDEF* zpráv na NFC tagy.

V praxi mohou při zápisu *NDEF* zprávy nastat 2 situace. První situace je bezproblémová a na NFC tag lze zprávu zapsat. V následujícím kódu je vidět zápis krátkého identifikátoru ve formátu *SMART POSTER* a obyčejného textu. U obyčejného textu je třeba určit, jakým jazykem je napsán. Kódy jazyků jsou určeny podle organizace *IANA*. Podle specifikace *NFC fóra* je každý *NDEF Well Known Type* textový záznam formátován následovně. Po prvním bajtu, jehož složení je vidět v tabulce 5.2, následuje *IANA* kód jazyka záznamu. Počet bytů kódu jazyka je určen právě v prvním bajtu záznamu. Zbytek je zpráva samotná.

bity	význam
7	0: UTF-8, 1: UTF-16
6	musí být nula
0..5	délka <i>IANA</i> jazykového kódu

Tabulka 5.2: Ukázka složení prvního bajtu textového NFC RTD záznamu.

```

1 //nejdriv je treba dostat byty z promennych ubid a desc
2 byte[] payload = ubid.getBytes(Charset.forName("UTF-8"));
3 byte[] payloadDescPrep = desc.getBytes(Charset.forName("UTF-8
  "));
4 byte[] finalPayload = new byte[payloadDescPrep.length + 3];
5 //LSB 0100000 znamena ze IANA kod je 2 byty dlouhy a zprava
  je v UTF-8
6 finalPayload[0] = 0x02;
7 //zprava bude v ceskem jazyce, kteremu odpovida kod cs
8 finalPayload[1] = 0x63; //c
9 finalPayload[2] = 0x73; //s
10
11 System.arraycopy(payloadDescPrep, 0, finalPayload, 3,
  payloadDescPrep.length);
12
13 //pripraveni NDEF zaznamu na zapis

```

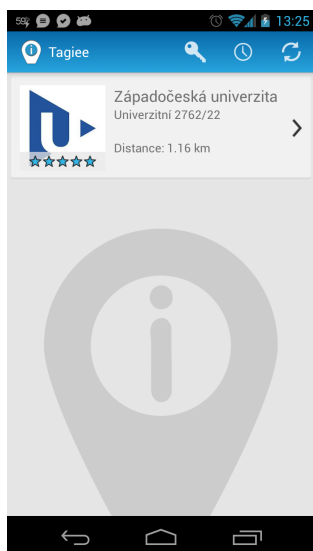
```
14 NdefRecord record = new NdefRecord(NdefRecord.TNF_WELL_KNOWN ,  
    NdefRecord.RTD_SMART_POSTER, null, payload);  
15 NdefRecord recordDesc = new NdefRecord(NdefRecord.  
    TNF_WELL_KNOWN, NdefRecord.RTD_TEXT, null, finalPayload);
```

V druhém případě se může stát, že na tag nepůjde zapsat. Jednou z příčin může být skutečnost, že tag není na zápis NDEF zpráv připraven. To tag dává na vědomí tím, že podporuje technologii *NDEFFormatable*. Pak lze takový tag naformátovat a NDEF zprávu zapsat. Pokud na tag nelze NDEF zprávu zapsat ani není *NDEFFormatable*, je pro tyto účely tag zpravidla nepoužitelný.

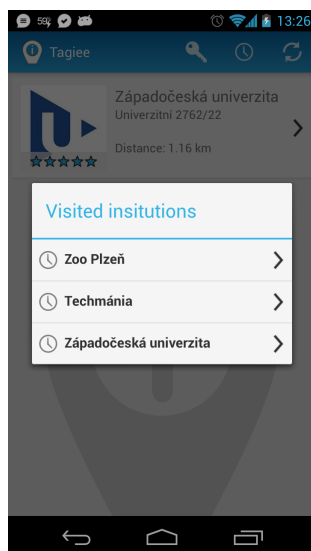
5.9 Ukázky aplikací

V této kapitole jsou nasnímané obrazovky při používání jak klientské aplikace Tagiee, tak administrátorské aplikace Tagiee Admin. Ukázka aplikace na tabletu je na obrázku 5.7. Pro ukázku aplikace byla použita zařízení *Samsung Galaxy Nexus* a *Asus Transformer TF101*.

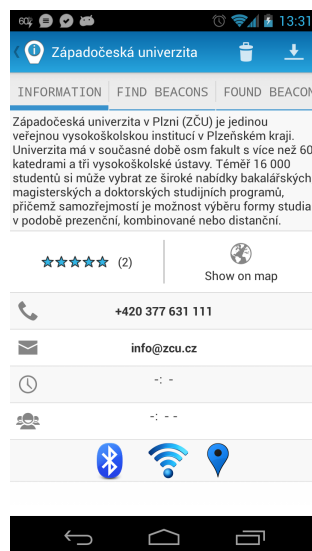
5.9.1 Tagiee



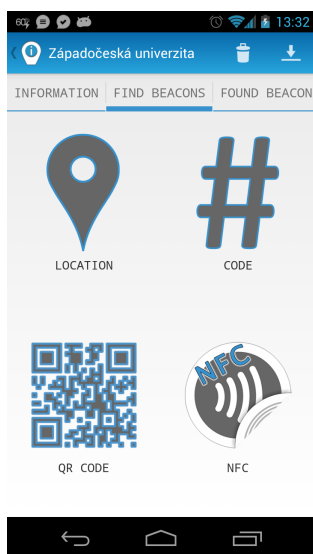
(a) Seznam všech institucí v okolí



(b) Přihlášení k instituci z historie



(c) Základní informace o instituci



(d) Možnost vyhledávání majáků

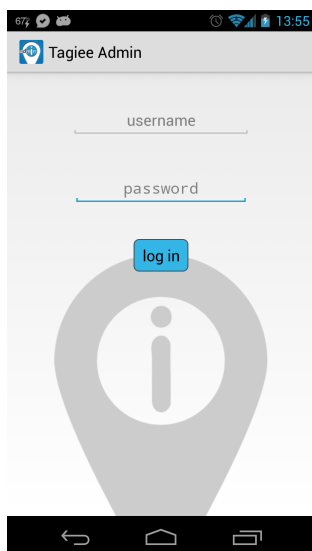


(e) Seznam již navštívených majáků

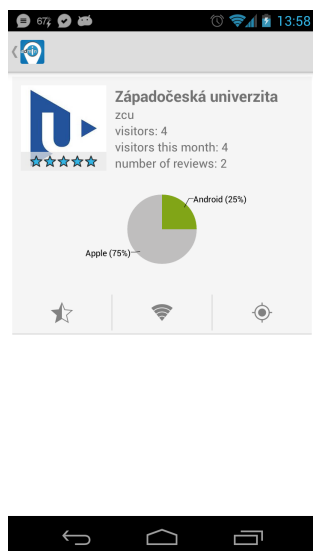


(f) Zobrazení obsahu konkrétního majáku

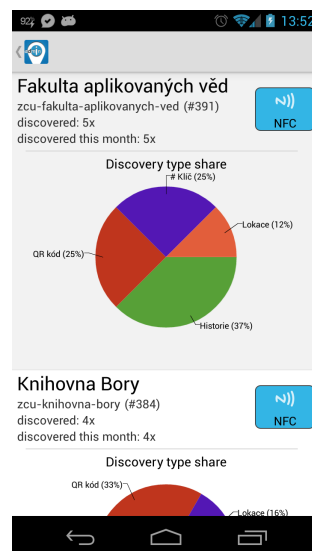
5.9.2 Tagiee Admin



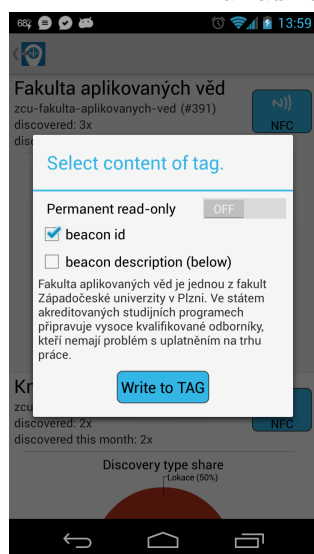
(a) Přihlašovací obrazovka



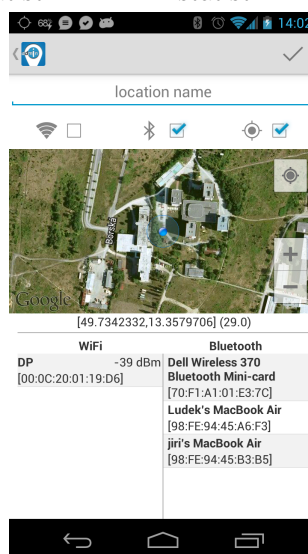
(b) Seznam institucí přihlášeného správce včetně základních statistik



(c) Seznam všech majáků konkrétní instituce včetně statistik



(d) Příprava obsahu, který bude zapsán na NFC tag



(e) Rozhraní pro uložení lokace do systému. Ta může být poté přiřazena k majáků v administraci.

6 Ověření funkčnosti

6.1 Modelová situace - DOD areál ZČU Bory

System Tagiee by se ideálně dal použít v areálu ZČU Bory při Dni otevřených dveří.

Den otevřených dveří je akce, při níž do areálu ZČU přichází zejména velké množství zájemců o budoucí studium na Západočeské univerzitě v Plzni. Ti si procházejí všechny budovy, ve kterých jsou pro ně připraveny různé ukázky a soutěže, zjišťují informace o průběhu studia apod. K tomu je třeba, aby byly na všech „stanovištích“ informační letáky a personál, který by podal dodatečné informace.

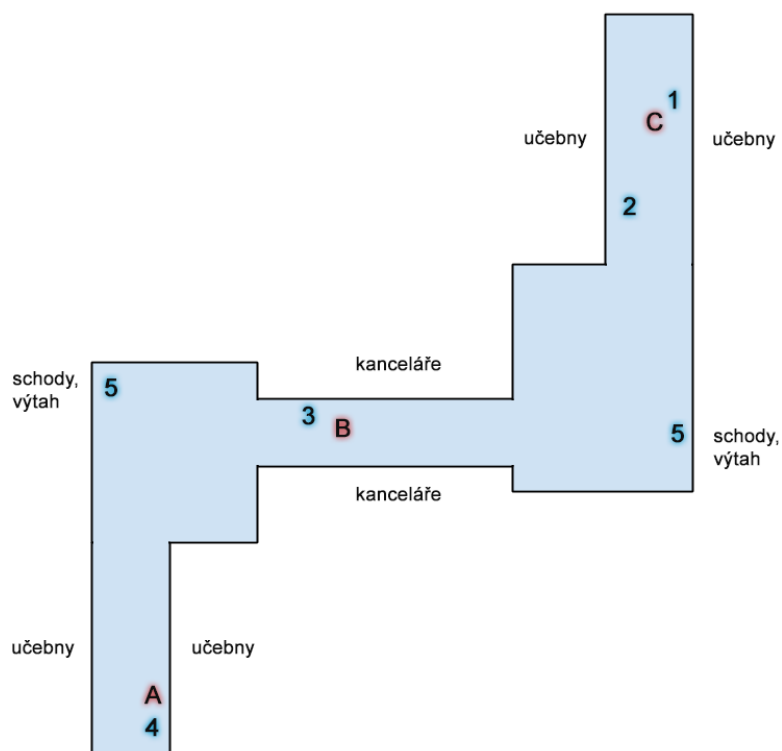
S využitím systému Tagiee bude každá budova v areálu zavedená v systému, tudíž návštěvníci budou na mapě vidět popis a polohu jednotlivých budov. Každé patro v každé budově bude mít u vstupu nalepený QR kód identifikující jednotlivé katedry sídlící na patře. U všech významnějších místností (laboratoře atd.) budou také u dveří QR kódy případně doplněné o NFC tagy. Dále lze využít stávající infrastrukturu WiFi sítě pro určení přibližné polohy uvnitř budovy a nabídnout návštěvníkovi nejbližší informační majáky.

6.2 Ověření funkčnosti - Areál ZČU Bory

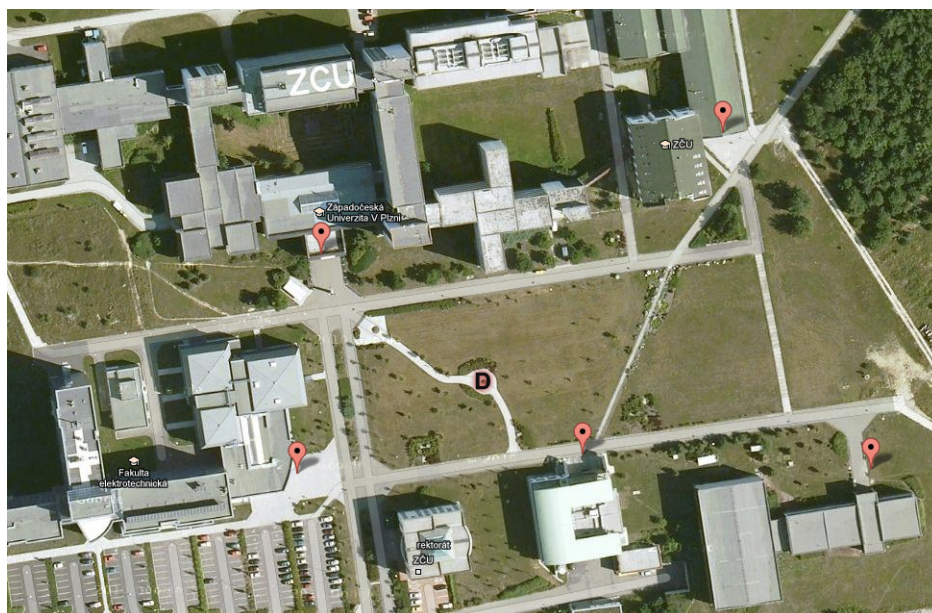
Pro ověření funkčnosti klientské aplikace a celého systému byla vytvořena instituce Západočeská univerzita, pro kterou byla vytvořena sada informačních majáků. Majáky jsou vytvořeny pro jednotlivé budovy, jak je vidět na obrázku 6.2. Dále jsou vytvořeny informační majáky pro Katedru informatiky a výpočetní techniky ve 4. patře budovy FAV, které jsou znázorněny na zjednodušeném plánu na obrázku 6.1.

Seznam venkovních majáků

- Fakulta aplikovaných věd
- Fakulta elektrotechnická



Obrázek 6.1: Mapa 4. patra budovy FAV.



Obrázek 6.2: Venkovní mapa areálu ZČU Bory.

- Knihovna Bory
- Katedra tělesné výchovy a sportu
- Menza ZČU Bory

Seznam vnitřních majáků

1. Učebna počítačové grafiky
2. Učebna počítačových sítí
3. Kancelář Doc. Ing. Přemysl Brada, MSc. PhD.
4. EEG ERP učebna
5. Katedra informatiky a výpočetní techniky

Speciálním majákem, který není na mapách znázorněn, je *Laboratoř digitální fabriky*, která je umístěna na stejné pozici jako maják číslo **1**, ovšem o patro níže.

Všechny vnitřní majáky jsou definovány WiFi signály sítě *eduroam* a *zcu-mobile*. Jediný maják číslo 3 je určen i Bluetooth signálem. Venkovní majáky jsou určeny pouze svou geografickou polohou, kromě majáku budovy, kde sídlí *Katedra tělesné výchovy a sportu*¹. Ten obsahuje krom geografické polohy i seznam signálů WiFi.

6.2.1 Výsledky ověření

Pro ověření správné funkčnosti klientské aplikace byla použita následující zařízení:

- **Samsung Galaxy Nexus** - smartphone, 4,65", Android 4.2.2, 1,2 GHz dual-core Cortex-A9, 1 GB RAM
- **Sony Ericsson Xperia Neo V** - smartphone, 3,7", Android 4.0.3, 1 GHz Snapdragon MSM8255, 512 MB RAM

¹na obrázku 6.2 se jedná o maják vpravo dole

- **Samsung Google Nexus S** - smartphone, 4", Android 4.1.2, 1 GHz Cortex-A8, 512 MB RAM
- **ASUS Eee Pad Transformer TF101** - tablet, 10,1", Android 4.0.3, NVIDIA® Tegra2™, 1 GB RAM

Na obrázcích 6.1 a 6.2 jsou kromě majáků vyznačeny ještě pozice, kde probíhalo zjišťování polohy a dostupných majáků v okolí. Tyto místa jsou označeny písmeny A - D.

Na obrázku 6.3 jsou ukázány výsledky nalezených majáků při průzkumu jak uvnitř budovy, tak venku v areálu univerzity. V části (c) a (d) jsou výsledky z téhož místa, avšak výsledky nejsou zcela shodné. V části (c) je navíc *Laboratoř digitální fabriky*, která je ovšem o patro níže. Díky tomu, že WiFi signál projde skrz podlahu, mohou být WiFi otisky dvou majáků, které spolu sousedí (ve smyslu pater), velmi podobné a může tak dojít k takovéto chybné identifikaci. Pro tento případ je vhodnější využít technologii Bluetooth pro identifikaci majáků, či vhodně omezit sílu vysílaného WiFi signálu, aby neprošel skrz podlahu.

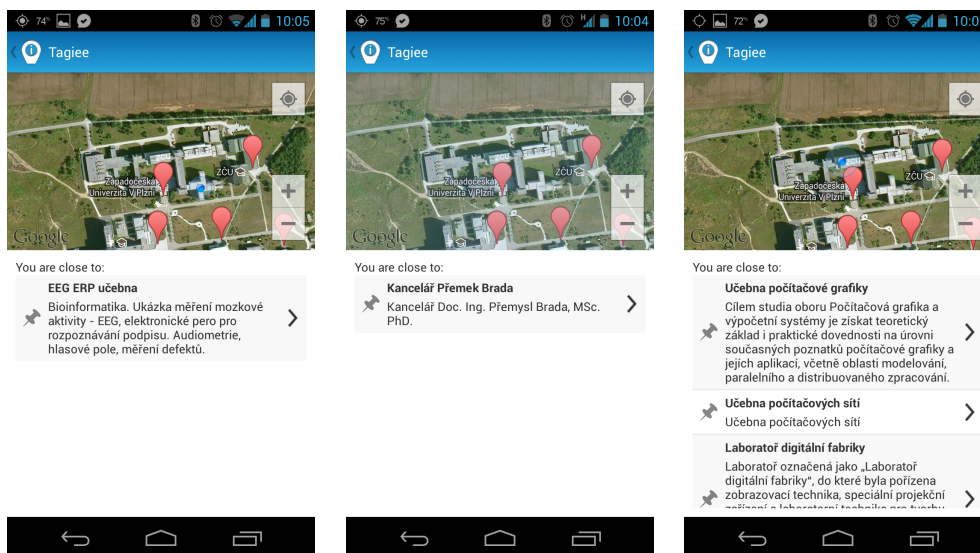
Další nepřesný výsledek je vidět na obrázku v části (e), kde je venku detekován maják *Katedry tělesné výchovy a sportu*, ačkoliv je od něj uživatel dostatečně vzdálen². Jedná se o hraniční pozici, pokud se vzdálenost zvětší o dalších pár metrů, maják již není identifikován. Tato nepřesnost je způsobena již zmíněným faktem, že tento jediný maják je možné identifikovat i podle WiFi otisku. Ve venkovních prostorách, kde šíření WiFi signálu nebrání žádné překážky jako např. zdi, je dosah majáku natolik významný, že může způsobovat tyto chybné identifikace.

Ostatní majáky - tzn. venkovní budovy určené pouze geografickou polohou, patra a učebny určené QR kódem, NFC tagem, Bluetooth signálem, WiFi signálem či jejich kombinacemi - byly identifikovány korektně. Výše uvedené chybné identifikace byly způsobeny vlastnostmi technologií, které nelze nijak změnit. Při dodržování následujících zásad, lze tyto chybné identifikace eliminovat.

- Venkovní majáky definovat pouze pomocí geografické polohy.
- Při velké hustotě WiFi vysílačů je vhodné doplnit definici majáků o Bluetooth otisk.

²poloha uživatele je znázorněna modrou šipkou

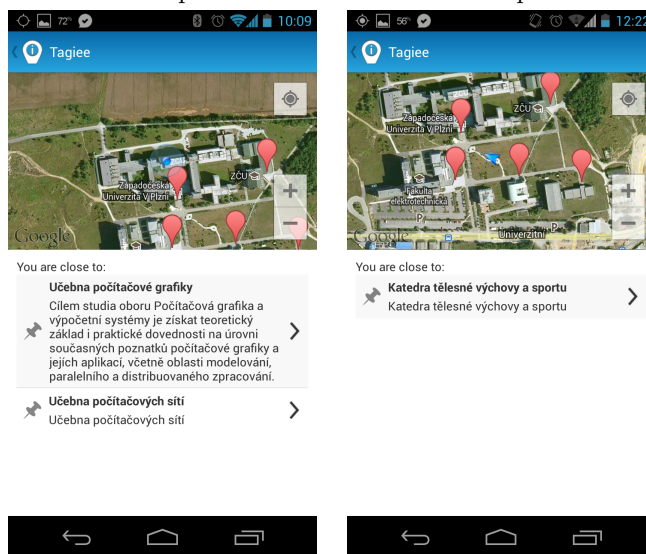
- Pokud spolu majáky sousedí ve smyslu pater, je vhodné buď omezit výkon WiFi vysílačů či definovat majáky pouze Bluetooth otiskem.



(a) Dostupné majáky na pozici A.

(b) Dostupné majáky na pozici B.

(c) Dostupné majáky na pozici C.



(d) Dostupné majáky na pozici C, situace 2.

(e) Dostupné majáky na pozici D.

Obrázek 6.3: Ukázka výsledků průzkumu okolních majáků.

7 Závěr

Cílem této diplomové práce bylo prozkoumat trh existujících systémů informačních majáků, zjistit jaké technologie a způsoby využívají pro identifikaci majáků a dále navrhnout další možnosti, jak lze informační majáky identifikovat využitím mobilního zařízení. Na základě zjištěných informací byla navržena a realizována mobilní aplikace pro systém informačních majáků na platformě Android.

Během analýzy bylo nalezeno a prozkoumáno celkem šest existujících systémů. U každého z nich byla popsána základní možnost použití systému včetně technologií, které používá pro identifikaci majáků. Dále bylo navrženo celkem deset metod / technologií, jež se dají využít pro identifikaci informačních majáků mobilním zařízením, ať se jedná o identifikaci majáků uvnitř budovy, či ve venkovních prostorech. Vzhledem k obecnosti navrženého systému Tagiee a velké otevřenosti Android SDK (oproti ostatním majoritním mobilním platformám) bylo vybráno celkem šest metod / technologií pro identifikaci majáků, které budou implementovány v klientské aplikaci systému Tagiee.

Klientská aplikace byla vytvořena tak, aby se dala pohodlně používat jak na telefonech, tak na tabletech. Správná funkčnost aplikace včetně algoritmů pro identifikaci informačních majáků byla ověřena na modelové situaci v areálu Západočeské univerzity v Plzni. Instituce čítá 5 venkovních a 6 vnitřních majáků, jež byly záměrně navrženy tak, aby se otestovaly všechny možnosti identifikace majáků. Aplikace identifikovala většinu majáků korektně, chybné identifikace byly způsobeny nevhodným použitím technologií. Detailní výsledky ověření jsou uvedeny v kapitole 6.

Systémy informačních majáků jsou v současné době rozvíjejícím se oborem a lze očekávat jejich častější využití ve veřejné, vzdělávací i komerční sféře. Celý systém Tagiee včetně mobilní klientské aplikace se ukázal jako použitelný a určitě má potenciál k dalšímu rozvoji. Mezi jedno z možných rozšíření klientské aplikace patří implementace služby *Google Cloud Messaging*, díky které by pak správci institucí byli schopni zasílat krátké zprávy na zařízení uživatelů, kteří jejich instituci navštívili, a mohli je tak například informovat o nových exponátech či probíhajících akcích.

Seznam obrázků

2.1	Zobrazení plánů jednotlivých pater nákupního střediska Arcaden, Hamburg, Německo	6
2.2	Ukázka natrénovaného systému. Zelené body určují polohu uložených „otisků“.[Bolliger()]	8
2.3	Ukázka aplikace TripAdvisor pro hlavní město Praha - seznam turistických atrakcí.	10
2.4	Ukázka klientů systému informačních majáků pro konkrétní instituce realizované na platformě Android	13
3.1	Způsob rozložení a význam informací uvnitř QR kódu. [qr-(2000)]	16
3.2	Ukázka QR kódů	18
3.3	Příklad použití WiFi otisků.	20
3.4	Jak vypadá NFC tag uvnitř. [nfc(b)]	24
4.1	Ukázka UI prvku Action Bar[and(a)]	34
4.2	Ukázka UI prvku Title Bar	34
4.3	Porovnání podílu jednotlivých verzí OS Android	36
4.4	Životní cyklus fragmentu.[and(c)]	37
4.5	Ukázka vícenásobného použití fragmentu.[and(c)]	38
5.1	Návrh komponent UI a jejich provázání	46
5.2	Algoritmus hledání majáků	52
5.3	Ukázka zapnutí / vypnutí služeb Google.	53
5.4	API klíče pro Android aplikaci.	54
5.5	Rozhodování <i>Tag Dispatch Systemu</i> . [and(d)]	57
5.6	Datový model databáze	61
5.7	Ukázka aplikace pro různé konfigurace zařízení	64
6.1	Mapa 4. patra budovy FAV.	71
6.2	Venkovní mapa areálu ZČU Bory.	71
6.3	Ukázka výsledků průzkumu okolních majáků.	74

Seznam tabulek

3.1	Maximální dosah a příkon Bluetooth zařízení dle třídy.[bt()]	21
4.1	Nové kvalifikátory.[and(b)]	35
4.2	Příklad použití kvalifikátorů.[and(b)]	35
5.1	Přehled některých podporovaných RTD	59
5.2	Ukázka složení prvního bajtu textového NFC RTD záznamu.	66

Přehled použitých zkratek a značení

Zkratka	Vysvětlení
API	Application Programming Interface - jedná se o sadu tříd, funkcí či protokolů nějaké knihovny.
BSSID	Basic Service Set Identification - unikátní identifikace bezdrátového přístupového bodu.
BTS	Base Transceiver Station - zařízení usnadňující bezdrátovou komunikaci mezi sítí operátora a mobilním zařízením.
DOD	Den otevřených dveří.
dp	Density-independent Pixel - virtuální jednotka pixelu, která je ekvivalentem jednoho fyzického obrazového bodu na obrazovce s 160 dpi. Při
dpi	dots per inch - počet obrazových bodů, které se vejdou do délky jednoho palce.
Hz	Hertz - jednotka frekvence v soustavě SI.
IMEI	International Mobile Equipment Identity - unikátní patnáctimístné číslo přiřazené výrobcem telefonu.
LED	Light-Emitting Diode - polovodičový světelný zdroj.
MAC	Media Access Control - jedinečný identifikátor síťového zařízení.
NFC	Near Field Communication - sada standardů definující architekturu bezdrátové technologie pro komunikaci na krátké vzdálenosti.
QR	Quick Response - nejčastěji použítá ve spojení s QR kódy.
RSSI	Received Signal Strength Indication - označení pro jednotku označující kvalitu signálu pro bezdrátová zařízení.
SDK	Software Development Kit - vývojová sada umožňující vývoj aplikací pro OS Android.
UI	User Interface - uživatelské rozhraní.
ZČU	Západočeská univerzita v Plzni.

Literatura

- [and(a)] *Action Bar* [online]. a. [cit. 20.4.2013]. Dostupné z: <http://developer.android.com/design/patterns/actionbar.html>.
- [and(b)] *Android 3.2 APIs* [online]. b. [cit. 20.4.2013]. Dostupné z: <http://developer.android.com/about/versions/android-3.2.html>.
- [and(c)] *Fragments* [online]. c. [cit. 20.4.2013]. Dostupné z: <http://developer.android.com/guide/components/fragments.html>.
- [and(d)] *NFC Basics* [online]. d. [cit. 23.4.2013]. Dostupné z: <http://developer.android.com/guide/topics/connectivity/nfc/nfc.html>.
- [and(e)] *Eric Schmidt: Google now at 1.5 million Android activations per day* [online]. e. [cit. 19.4.2013]. Dostupné z: <http://www.engadget.com/2013/04/16/eric-schmidt-google-now-at-1-5-million-android-activations-per/>.
- [bt()] *A Look at the Basics of Bluetooth Wireless Technology* [online]. [cit. 16.4.2013]. Dostupné z: <http://www.bluetooth.com/Pages/Basics.aspx>.
- [bt-()] *New Alliance helps you find 'needle in a haystack'* [online]. [cit. 19.4.2013]. Dostupné z: <http://conversations.nokia.com/2012/08/23/new-alliance-helps-you-find-needle-in-a-haystack/>.
- [byt()] *ByteLight* [online]. [cit. 13.4.2013]. Dostupné z: <http://www.bytelight.com/faqs>.
- [ind()] *Indoor Maps availability - Google Maps for Mobile Help* [online]. [cit. 10.4.2013]. Dostupné z: <http://support.google.com/gmm/bin/answer.py?hl=en&answer=1685827&topic=1685871&ctx=topic>.

- [loc()] *Technology* [online]. [cit. 17.4.2013]. Dostupné z: <http://locata.com/technology/>.
- [nfc(a)] *NFC Forum* [online]. a. [cit. 16.4.2013]. Dostupné z: http://www.nfc-forum.org/aboutnfc/nfc_andy_contactless.
- [nfc(b)] *NFC Tag Antennas* [online]. b. [cit. 16.4.2013]. Dostupné z: http://rapidnfc.com/nfc_tag_antennas.
- [nfc(c)] *Reverse-Engineering a Cryptographic RFID Tag* [online]. c. [cit. 17.4.2013]. Dostupné z: <http://www.cs.virginia.edu/evans/pubs/usenix08/usenix08.pdf>.
- [nfc(d)] *NFC Tag Types* [online]. d. [cit. 16.4.2013]. Dostupné z: <http://www.nfc.cc/technology/nfc-tag-types/>.
- [qr-(2000)] *Information technology — Automatic identification and data capture techniques — Bar code symbology — QR Code* [online]. 2000. [cit. 14.4.2013]. Dostupné z: http://raidenii.net/files/datasheets/misc/qr_code.pdf.
- [wim()] *WiMAX vs. LTE: Should You Switch?* [online]. [cit. 18.4.2013]. Dostupné z: <http://www.pcmag.com/article2/0,2817,2403490,00.asp>.
- [Bolliger()] BOLLIGER, P. *RedPin - Adaptive. Zero-Configuration Indoor Localization through User Collaboration* [online]. [cit. 13.4.2013]. Dostupné z: <http://www.vs.inf.ethz.ch/publ/papers/bolligph-redpin2008.pdf>.
- [Cooksey()] COOKSEY, D. *Understanding the Global Positioning System* [online]. [cit. 14.4.2013]. Dostupné z: <http://www.montana.edu/gps/understd.html>.

A Přílohy

A.1 Struktura CD

Příložené CD obsahuje následující adresáře:

- `diplomova-prace`: V tomto adresáři jsou všechny soubory potřebné k úspěšnému přeložení textu diplomové práce.
 - `img` - obsahuje všechny obrázky použité v diplomové práci
- `javadoc-dokumentace`: Obsahuje všechny soubory vygenerované JavaDoc dokumentace.
- `knihovny`: Tento adresář obsahuje knihovny použité při vývoji klientské aplikace.
 - `Commons-lang3` - z této knihovny je použita funkce na escapování národních znaků
 - `Gson` - knihovna pro parsování JSON zpráv
 - `JakeWharton-Android-ViewPagerIndicator-8cd549f` - knihovna pro zobrazení identifikátorů prvku UI *ViewPager*
 - `YouTubeAndroidPlayerApi-1.0.0` - knihovna pro přehrávání streamovaného videa z YouTube
- `ZBarScannerLibrary` - knihovna čtečky QR kódů jako Android projekt
- `zdrojove-kody`
 - `Tagiee` - tento adresář obsahuje projekt celé klientské aplikace. V případě importování projektu (lze pouze do Eclipse IDE) je třeba mít nainstalované Android SDK a nainportovanou knihovnu *google-play-services-lib*. Po importování projektu je třeba v jeho nastavení nastavit správné cesty ke všem výše uvedeným knihovnám.

A.2 Uživatelská příručka

A.2.1 Instalace aplikace

Klientskou aplikaci si lze stáhnout na adrese `http://dev.tagiee.com` v sekci Android. Aplikace (.apk soubor) je zde ke stažení v několika verzích, avšak doporučeno je stáhnout nejnovější verzi 1.0.0. Aplikace není v současné době umístěna v oficiálním portálu aplikací *Google Play*, tudíž je nutné, aby měl uživatel povoleno instalovat aplikace z jiných zdrojů, než je *Google Play*. Povolení se v zařízení nastavuje následovně¹: Nastavení -> Zabezpečení -> Správa zařízení -> Povolit neznámé zdroje.

Pro úspěšnou instalaci je třeba mít mobilní zařízení s OS Android ve verzi 3.0+ a alespoň 10 MB volného místa. Pokud si uživatel stáhne instalační soubor do vnitřní paměti telefonu, lze ho pomocí libovolného správce souborů spustit a aplikace se nainstaluje. V případě, že si uživatel stáhne instalační soubor do počítače, je třeba ho buď přkopírovat do mobilního zařízení, kde se spustí dříve uvedeným způsobem, nebo ho nainstalovat využitím programu *adb* příkazem uvedeným níže. Tento postup předpokládá, že má uživatel na svém počítači nainstalován Android SDK, konkrétně balík *Android SDK Platform-tools*.

```
adb install Tagiee_1_0_0.apk
```

Analogickým způsobem lze nainstalovat i administrátorskou aplikaci *Tagiee-Admin*, která je dostupná na stejné adrese v sekci Android.

A.2.2 Používání aplikace

Po úspěšné instalaci a spuštění aplikace jsou okamžitě vyhledávány instituce v okolí. V případě, že zařízení nemá povoleny služby pro zjišťování polohy, je nutné se k instituci přihlásit ručně. Vytvořená instituce *Západočeské univerzity v Plzni* má svůj unikátní identifikátor **zcu**. Po přihlášení může uživatel s aplikací dále pracovat. Rozmístění majáků v areálu je znázorněno v kapitole 6 na obrázcích 6.1 a 6.2.

Uživatel si také může vytvořit svou vlastní instituci s vlastními majáky

¹Postup se může mírně lišit v závislosti na verzi OS Android a výrobci zařízení.

využitím webového rozhraní vytvořeného v rámci diplomové práce Bc. Lud'ka Vlka na adrese <http://www.tagiee.com>. K tomu je ovšem nutné si vytvořit vlastní uživatelský účet.

Aby mohl správce využít veškerou funkčnost administrátorské aplikace (prohlížení hodnocení a statistik instituce a majáků, příprava NFC tagů), musí se nejprve přihlásit svým uživatelským jménem a heslem, které zadal při registraci v systému Tagiee (na adrese <http://www.tagiee.com>).

Pro otestování klientské aplikace následuje několik QR kódů včetně unikátních číselných kódů s popisem, co značí.



(a) Fakulta aplikovaných věd, #391



(b) Katedra informatiky a výpočetní techniky, #169



(c) Učebna počítačových sítí, #805



(d) EEG ERP učebna, #850

A.3 Seznam JSON požadavků a odpovědí

V této části budou popsány jednotlivé požadavky na server, tzn. co vše je potřeba vyplnit do požadavku a následně bude popsána odpověď serveru.

A.3.1 add_review

Tento požadavek je na server odeslán vždy, když uživatel přidá hodnocení instituce, ke které je přihlášen.

```
[type] => add_review
[udid] => string - unikatni identifikator zarizeni
[uiid] => string - instituce, ke ktere se hodnoceni pridava
[review] => string - kratke slovni hodnoceni
[rating] => int
[platform] => string {ios|android|win}
[version] => string - ve formatu x.x.x
```

ODPOVED

```
[status] => boolean
```

A.3.2 beacon_found

Tento požadavek odesílá zařízení automaticky při každém zobrazení detailu informačního majáku. Na základě takto získaných dat server vypočítává různé statistiky.

```
[type] => device_found_beacon
[udid] => string - unikatni identifikator zarizeni
[uiid] => string - instituce, ke ktere se pozadavek tyka
[tech] => string {gps|qr|nfc|key} - pouzita technologie, neni
    povinna
[discovered_datetime] => datetime
[platform] => string {ios|android|win}
[version] => string - ve formatu x.x.x
```

ODPOVED

```
[status] => boolean
```

A.3.3 device_visit_institution

Stejně jako předchozí požadavek je i tento pouze pro statistické údaje. Je odeslán při každém přihlášení uživatele k instituci.

```
[type] => device_visit_institution
[udid] => string - unikatni identifikator zarizeni
[uiid] => string - instituce, ke ktere se pozadavek tyka
[discovered_datetime] => datetime
[platform] => string {ios|android|win}
[version] => string - ve formatu x.x.x
```

ODPOVED

```
[status] => boolean
```

A.3.4 get_institution_by_uiid

Jedná se o nejdůležitější požadavek, který vrací kompletní informace o dané instituci. Odpověď je personalizovaná podle uživatele, co se týče příznaků, zda-li uživatel instituci hodnotil a které majáky již navštívil.

```
[type] => get_institution_by_uiid
[udid] => string - unikatni identifikator zarizeni
[uiid] => string - instituce, ke ktere se pozadavek tyka
[platform] => string {ios|android|win}
[version] => string - ve formatu x.x.x
```

ODPOVED

```
[status] => boolean
[response]=>
(
  [revision] = int
  [logo] => string - url obrazku
  [uiid] => string
  [name] => string
  [description] => string
  [address] => string
  [lat] => double
  [lon] => double
  [tel] => string
  [support_email] => string
  [prices] =>
  (
    [1..N array]
    [type] => string
    [price] => string
    [currency] => string
```



```
)
[opening_hours] =>
(
  [1..N array]
    [days] => string
    [time] => string
  )
[allow_offline] => boolean
[allow_uploads] => boolean
[allow_reviews] => boolean
[number_reviews] => int
[rating] => float
[user_posted_review] => boolean
[beacons] =>
(
  [1..N array] =>
    (
      [ubid] => string
      [name] => string
      [description] => string
      [lat] => double
      [lon] => double
      [key] => int
      [additional] =>
        (
          [1..N array] => - síla signalu určena pořadím
            (
              [tech] => string
              [data] => string
            )
          )
        )
      [content] =>
        (
          [1..N array] =>
            (
              [type] => string {image|text|audio|video|
                youtube|href}
              [heading] => string
              [content] => string
              [resource_id] =>
                (
                  [0..N array] => int
                )
            )
          )
        )
      [discovered] => boolean
    )
  )
)
[resources] =>
(
  [1..N array] =>
    (
      [id] => int
      [type] => string
      [size] => double
    )
  )
)
```


tému znova nebo zda-li stačí použít lokální uložená data.

```
[type] => get_institution_revision
[udid] => string - unikatni identifikator zarizeni
[uuid] => string - instituce, ke ktere se pozadavek tyka
[platform] => string {ios|android|win}
[version] => string - ve formatu x.x.x
```

ODPOVED

```
[status] => boolean
[response]=>
(
    [revision] = int
)
```

A.3.7 get_reviews

Tímto požadavkem aplikace žádá server o vrácení seznamu hodnocení dané instituce.

```
[type] => get_reviews
[udid] => string - unikatni identifikator zarizeni
[uuid] => string - instituce, ke ktere se pozadavek tyka
[platform] => string {ios|android|win}
[version] => string - ve formatu x.x.x
[from] => int - spodni limit
[to] => int - horni limit
```

ODPOVED

```
[status] => boolean
[response]=>
(
    [1..N array]
        [added] => string
        [rating] => double
        [review] => string
)
```