Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Tres Faciunt Collegium

Vývoj aplikácie pre mobilný telefón

Tímový projekt

Študijný program: Počítačové a komunikačné systémy a siete
Mieste vypracovania: Ústav počítačových systémov a sietí, FIIT STU v Bratislave
Vedúci projektu: Ing. Peter Trúchly, PhD.

Apríl 2012

Obsah

1. Úvod 1

1.1. Prehľad dokumentu 1

2. Analýza problematiky 1

2.1. Analýza existujúcich riešení 1

2.1.1. PPT Remote 1

2.1.2. PowerPoint OpenOffice Remote 3

2.1.3. i-Clickr PowerPoint Remote 5

2.1.4. RemotePPT 7

3. Špecifikácia požiadaviek 8

3.1. Funkčná špecifikácia 8

4. Návrh riešenia 11

4.1. Server 11

4.1.1. Požiadavky na server 12

4.2. Klient 12

4.3. Komunikácia klient-server 13

4.3.1. Nájdenie servera 13

4.3.2. Komunikácia počas prezentácie 14

4.4. Prezentovanie 15

4.4.1. Módy prezentovania 16

5. Zmeny v návrhu 18

5.1. Server 18

5.2. Klient 18

6. Implementácia 19

6.1. Implementácia serverovej časti 19

6.1.1. Správa súborov 19
6.1.2. Modul na komunikáciu cez SOAP webservice 20
6.1.3. Modul na komunikáciu posielaním správ vo formáte JSON 20
6.1.4. Modul na úvodnú UDP komunikáciu 21
6.2. Implementácia klienta pre platformu Android 22
6.2.1. Komunikácia so serverom 22
6.2.2. Ovládanie aplikácie 23
6.3. Implementácia klienta pre platformu Bada 23
7. Zhodnotenie 26
8. Zdroje 27

Zoznam obrázkov

Obrázok 1 - Aplikácia PPT Remote 1
Obrázok 2 - Aplikácia PowerPoint OpenOffice Remote 3
Obrázok 3 - Aplikácia i-Clickr PowerPoint Remote 5
Obrázok 4 - Aplikácia RemotePPT 7
Obrázok 5 - Diagram navrhovaných tried prezentačnej časti aplikácie 16
Obrázok 6 - Schématický nákres GUI klientskej aplikácie 17
Obrázok 7 - Spracovanie príkazov vo formáte JSON 20

1. Úvod

Účelom tohto dokumentu je detailné zachytenie všetkých fáz vývoja vytváraného systému na predmet tímový projekt.

1.1. Prehľad dokumentu

Dokument pozostáva z úvodu, zdrojov a siedmych základných častí a záveru. Prvá časť popisuje analýzu problematiky a existujúce riešenia v oblasti vývoja aplikácie. Druhá časť dokumentácie bližšie špecifikuje vytváranú aplikáciu na základe na ňu kladených požiadaviek. V tretej časti dokumentácie je podrobne popísaný predbežný návrh riešenia systému. Nasleduje kapitola so zmenami v návrhu systému, ktoré vyplynuli z procesu implementácie – táto je opísaná v kapitole číslo 6. V poslednej kapitole je opísané zhodnotenie nášho projektu.

2. Analýza problematiky

Táto časť projektu sa zaoberá analýzou riešenej problematiky. Analýza je rozdelená do nasledujúcich oblastí: analýza existujúcich riešení a analýza vývojových prostredí.

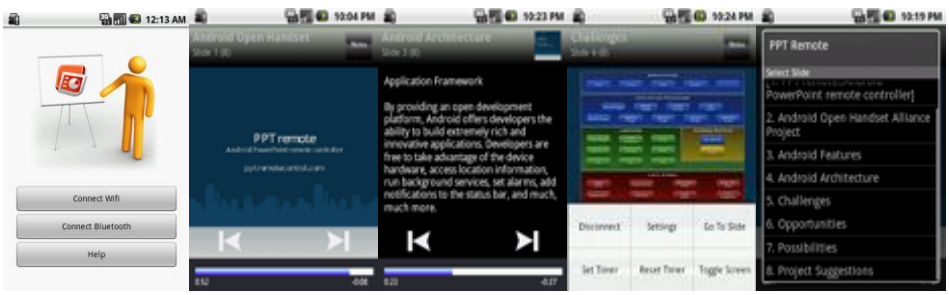
2.1. Analýza existujúcich riešení

V tejto časti sa nachádza analýza existujúcich aplikácií s podobným riešení problematiky. Ide zväčša o aplikácie pracujúce iba na jednej platforme.

2.1.1. PPT Remote

PPT Remote je aplikácia navrhnutá výlučne pre platformu Android. Ide o bezplatnú aplikáciu na uľahčenie ovládania prezentácií v prostredí Microsoft PowerPoint. Tento programový balík sa skladá z klientskej a serverovej časti. Serverová časť je navrhnuté výlučne pre operačný systém Windows XP a novšie. Prepojenie telefónu a počítača je dostupné pomocou technológie Bluetooth alebo WiFi.

Aplikácia využíva jednoduché používateľské rozhranie pre prepojenie telefónu s počítačom. Avšak aj pri poslednej zverejnenej verzii (Version 3.3.3) sa stále vyskytuje problém nadviazania spojenia pomocou technológie Bluetooth pri niektorých zariadeniach.



Obrázok 1 - Aplikácia PPT Remote

Používateľské rozhranie je navrhnuté veľmi jednoducho pre rýchlu a efektívnu prácu s aplikáciou. Následkom toho je okliešténá funkcionalita, ktorá poskytuje iba niekoľko rozšírených funkcií. Na hlavnej obrazovke sa nachádzajú dve tlačidlá, ktoré slúžia na prechádzanie snímkami prezentácie. Pri otočení displeja na šírku zaberajú tieto tlačidlá značnú časť obrazovky, čo sa môže negatívne odraziť na samotnom zobrazení prezentácie, ktorá nebude dostatočne viditeľná. Na obrazovke sa nachádza aj ďalšie tlačidlo slúžiace na zmenu prezentujúceho módu. Stlačením tlačidla sa zmení mód prezentácie na mód poznámok. Posledným rozširujúcim prvkom na obrazovke je efektívne umiestnený časovač v podobe grafického komponentu tzv. progress bar.

Ďalšou rozširujúcou funkciou volanou z menu je použitie čiernobielej farby na zobrazovanie obrázkov. Jej praktické využitie je diskutabilné, jediné pozitívum sme našli pri použití veľkej prezentácie s množstvom obrázkov a spojením pomocou Bluetooth, ktoré dosahuje omnoho nižšie prenosové rýchlosti ako WiFi alebo pri použití smartphonu s veľmi pomalým procesorom. Posledným rozšírením je posúvanie prezentácie s predstihom jednej stránky.

V bezplatnej verzii je nastavený obmedzený limit prezentácie na 15 snímok. Avšak k aplikácií existuje aj business verzia bez limitu, no za poplatok. Najväčšou nevýhodou je nutnosť inštalácie serverovej časti, čím sa stráca flexibilita použitia celého systému.

Vlastnosti aplikácie:

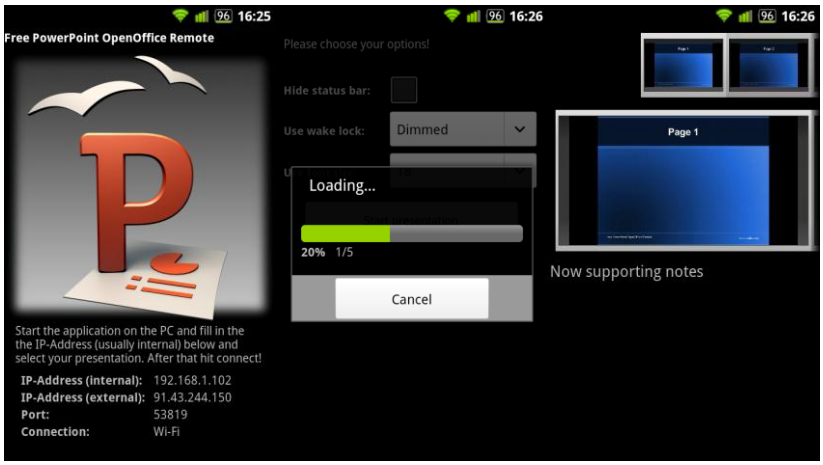
- Kontrola snímok PowerPoint prezentácie
- Možnosť prezerania v dvoch módoch - prezentácia alebo poznámky
- Možnosť prepojenia pomocou Bluetooth alebo WiFi
- Dotykové tlačidlá na prepínanie snímok
- Možnosť nastavenia časovača
- Režim obrazovky na výšku aj na šírku
- Jednoduchá ale nutná inštalácia
- Obmedzený počet snímok v bezplatnej verzii

Internetová stránka aplikácie je popísaná v zdroji [1].

2.1.2. PowerPoint OpenOffice Remote

PowerPoint OpenOffice Remote je aplikácia navrhnutá pre platformu Android. Táto aplikácia poskytuje vzdialené ovládanie pre Microsoft PowerPoint a OpenOffice prezentácie. Programový balík obsahuje klientsku a serverovú časť. Serverová časť programu je implementovaná v jazyku Java. To poskytuje jej použitie na viacerých, bežne používaných operačných systémoch ako Windows, Linux, či Mac.

Aplikácia poskytuje podporu pre súborové formáty .odt, .ppt a .pptx. Avšak pri .pptx vo verzii MS Office 2010 sa často vyskytujú chyby pri spracovaní prezentácie serverom.



Obrázok 2 - Aplikácia PowerPoint OpenOffice Remote

Pripojenie klientskej časti aplikácie k serveru prebieha za pomoci technológie WiFi. Chýba tu však podpora synchronizácie klienta so serverom. Pri výpadku signálu sa môže stať, že klientsky program bude v predstihu pred serverovou časťou. V novšej verzii je tento problém riešení nie moc efektívnym manuálnym synchronizovaním pomocou tlačidla na to určeného.

Po spustení programu a pripojení k serveru je potrebné najskôr načítať celú prezentáciu, čo limituje jej veľkosť na obmedzený počet snímok. Grafické rozhranie je navrhnuté veľmi intuitívne. V hornej časti sa nachádzajú miniatúrne zobrazenie všetkých snímok v podobe posuvného panelu. Pod týmto panelom sa nachádza obrázok aktuálnej snímky. Pri menšej obrazovke alebo pri menšom rozlíšení displeja však takáto veľkosť

snímky nemusí postačovať, čo spôsobí že snímka nemusí byť dostatočne viditeľná. V prémiovej verzií ja tento nedostatok riešení pomocou priblíženia snímky na celú obrazovku. Pod snímkou sa na obrazovke nachádzajú poznámky, no je tu možnosť nastavenia navigačných tlačidiel miesto nich. Navigácia je možná viacerými spôsobmi a závisí od verzie programu. Prepínanie medzi predchádzajúcimi a nasledujúcimi snímkami je riešené pomocou tzv. metódy ťahanie obrazovky, alebo spomínaných tlačidiel navigácie. V niektorých verziách sa po kliknutí na snímku prejde na ďalšiu snímku. Pomocou vrchného panelu môže prechádzať medzi všetkými snímkami.

Medzi rozšírené funkcie tejto aplikácie patrí nastavenie veľkosti písma poznámok, schovanie tzv. status bar panelu.

Veľkou nevýhodou je spracovanie servera, ktorý na prepínanie snímok používa simuláciu tlačidiel na klávesnici (key stroke). To môže v niektorých prípadoch vyvolať neželané reakcie, napr. pri použití niektorých prídavných prvkov v prezentácií ako video a podobne. Aplikácia je bezplatná, avšak ako už bolo spomínané, existuje Premium verzia, ktorá poskytuje ďalšie prídavné funkcie, ako napr. zmenu orientácie obrazovky. Táto verzia je však dostupná za poplatok.

Vlastnosti aplikácie:

- Podpora formátov MS PowerPoint a OpenOffice
- Prepojenie pomocou WiFi
- Rôzne spôsoby prepínania snímok
- Možnosť prejsť na ľubovoľnú snímku
- Bez nutnosti inštalácie servera
- Rozšírené funkcie v platenej Premium verzii

Internetová stránka aplikácie je popísaná v zdroji [2].

2.1.3. i-Clickr PowerPoint Remote

Najväčšou výhodou aplikácie i-Clickr PowerPoint Remote je jej podpora viacerých platforiem a viacerých operačných systémov. No vo svojej podstate ide o štyri absolútne rôzne aplikácie vyvíjané spoločnosťou Senstic pre platformy Apple iOS, Android, Windows Mobile 6 a Windows Phone 7. To znamená, že tieto aplikácie sú navzájom nekompatibilné. Treba zdôrazniť, že ide výlučne o platenú aplikáciu.

Softvérový balík pozostáva z dvoch komponentov. Klientskej časti určenej pre smartphone a serverovej časti určenej pre príslušný operačný systém. Nevýhodou tohto riešenia je nutnosť inštalácie serverovej časti, pričom treba zvoliť inštalačný súbor pre príslušný operačný systém. Tým sa stráca univerzálnosť a flexibilnosť použitia aplikácie. Podporované sú programy PowerPoint balíka Microsoft Office určeného pre Windows a Mac OS (od verzie 2004), oklieštená podpora pre novšie verzie balíka OpenOffice na systémoch Windows a program Keynote v rámci balíka iWork 09 pre systém Mac OS.

Obrázok 3 - Aplikácia i-Clickr PowerPoint Remote

Grafické rozhranie sa líši od použitej verzie. Pre všetky však platí, že v hornej časti obrazovky sa nachádzajú tlačidlá pre navigáciu medzi susednými snímkami. Navigácia je riešená pomocou dotykových tlačidiel. Niektoré verzie majú k dispozícii aj funkciu pre ľubovoľné prechádzanie medzi jednotlivými snímkami. Pod navigačným menu je zobrazená samotná aktuálna snímka prezentácie.

Medzi rozšírené funkcie patrí časovač s funkciou upozornenia pri vypršaní časového limitu. Pri niektorých verziách dokáže upozorniť aj predčasne pomocou vibrácie v stanovenom čase pred vypršaním časovača. Ďalšou funkciou je prepínanie medzi režimom prezentácie a režimom poznámok, ktoré sa zobrazujú na mieste snímok.

Prepojenie klientskej a serverovej časti je možné pomocou technológie Bluetooth alebo WiFi. Niektoré verzie podporujú aj spojenie pomocou Ad Hoc siete.

Vlastnosti aplikácie:

- Podpora prezentačných programov balíkov MS Office, OpenOffice, iWork
- Podpora pre rôzne operačné systémy a pre rôzne klientské platformy: iOS, Android, Windows Phone 7, Windows Mobile 6
- Prepojenie so serverovou časťou na počítači pomocou Bluetooth, WiFi, v niektorých verziách aj Ad Hoc
- Dotykové tlačidlá pre prepínanie snímok
- Možnosť nastavenia časovača
- Výlučne platená aplikácia
- Nutnosť inštalácie pre špecifický systém

Internetová stránka aplikácie je dostupná v zdrojoch [3].

2.1.4. RemotePPT

RemotePPT je momentálne (31.10.2011) pravdepodobne jediná aplikácia podporujúca systém Bada. Táto aplikácia je však stále vo vývoji. To sa odzrkadľuje hlavne na jej funkcionalite. V poslednej beta verzii poskytovala iba základné funkcie pre prechádzanie snímok. Okrem toho poskytuje možnosť zapnutia a vypnutia prezentácie.

Obrázok 4 - Aplikácia RemotePPT

Aplikácia je tvorená klientskou a serverovou časťou. Komunikácia medzi týmito zariadeniami prebieha pomocou technológie WiFi. Keďže aplikácia je iba v štádiu vývoja, komunikácia medzi zariadeniami nie je dostatočne otestovaná a prakticky funguje iba na simulátore.

Používateľské rozhranie tvorí obrazovka so štyrmi tlačidlami. Ako už bolo spomenuté, funkcia tlačidiel je obmedzená na prepínanie medzi snímkami a zapnutie a vypnutie prezentácie. Na obrazovke sa nenachádzajú snímky ani žiadne ďalšie funkcie. Serverovú časť aplikácie podporuje zatiaľ iba operačný systém Windows a balík Microsoft Office 2007 a 2010. Klientska časť podporuje iba platformu Bada.

Internetová stránka k aplikácii je priložená v zdroji [4].

3. Špecifikácia požiadaviek

3.1. Funkčná špecifikácia

Našim cieľom je vytvoriť aplikáciu na vzdialené riadenie prezentácií pomocou smartphonu. Táto aplikácia bude jednoduchá na ovládanie a zároveň bude integrovať veľké množstvo prídavných funkcií. Zároveň sa budeme snažiť zachovať univerzálnosť a flexibilitu jej použitia na rôznych systémoch. Softvérový balík aplikácie bude pozostávať z dvoch častí. Klientska časť bude učená pre smarthphone, ktorým slúži na ovládanie prezentácie a serverová časť sa bude nachádzať na počítači.

Na základe vyššie uvádzanej analýzy existujúcich riešení a našou tvorivou činnosťou sme dospeli k názoru, že navrhovaná aplikácia by mala spĺňať nasledujúce požiadavky:

- **Flexibilita a univerzálnosť**

Univerzálnosť aplikácie zabezpečíme použitím vývojového jazyka Java pre serverovú časť aplikácie. Tým zaistíme funkčnosť aplikácie na rôznych, bežne používaných operačných systémoch ako je Windows, Mac OS, či Linux.

Aplikácia bude vyvíjaná pre viaceré platformy. Konkrétne sa jedná o platformy Android a Bada.

- **Podpora viacerých formátov**

Jednou z hlavných požiadaviek je podpora viacerých dnes bežne používaných typov súborov. Samozrejmosťou je podpora súbor programu Microsoft PowerPoint, ktorý je najpoužívanejším na svete vo svojej kategórii. Niektorí používatelia však uprednostňujú voľne dostupné alternatívy, preto sme sa rozhodli pre podporu formátu balíka OpenOffice a ďalší populárny formát Pdf. Používateľ zároveň bude mať možnosť prezentovať snímky aj vo forme obrázkov.

- **Rôzne režimy prezentácie**

Používateľ aplikácie bude mať možnosť zvoliť si jeden z režimov prezentácie. Klasický režim zahrňuje zobrazenie aktuálnej snímky na klientskom zariadení. Ďalším režimom je textová štruktúra danej snímky, ktorú je vhodné zvoliť hlavne pri prezentáciách obsahujúce nadbytočné obrázky, ktoré majú skôr ilustračný charakter, alebo pri snímkach s veľkým množstvom textu. Tretím režimom bude tzv. Zoom, kde používateľ bude mať možnosť snímku priblížiť v určenej oblasti.

- **Možnosť vybrať prezentáciu**

Jednou z rozširujúcich funkcií je práca so súbormi. Používateľovi to umožní prepínať medzi obsahom, ktorý si nadefinuje pri inicializácií prezentovania. Zefektívni sa tak práca prednášajúceho, ktorý nebude natoľko závislý na riadiacej časti nachádzajúcej sa na serverovej časti programu.

- **Autentifikácia**

Autentizácia je nutná pre zaistenie bezpečnej práce s aplikáciou. Takáto funkcia absentuje vo všetkých analyzovaných existujúcich riešení. Bez autentifikácie by mohol ktokoľvek bez oprávnenia s nainštalovanou klientskou aplikáciou ovládať prezentáciu.

- **Master – slave**

Ďalšou rozšírenou funkciou je pripojenie viacerých nezávislých klientskych zariadení s tým, že len jedna bude mať možnosť aktívne ovládať prezentáciu. Ostatným používateľom sa prezentácia bude tiež zobrazovať na svojich staniciach, avšak budú iba pasívne, t.j. nebudú mať možnosť ju ovládať, až pokým im aktívne klientske zariadenie nedeleguje túto možnosť.

- **Poznámky**

Samozrejmosťou bude prepínanie medzi režimom zobrazenia snímok a poznámok. Pre túto funkciu bude vyhradené dotykové tlačidlo.

- **Ukazovateľ**

Ďalšou rozšírenou funkciou je možnosť ukazovateľa, ktorá bude nahrádzať klasický laserový ukazovateľ.

- **Časovač**

Dôležitou súčasťou bude aj funkcia časovača, ktorý bude používateľ môcť nastaviť a spustiť.

4. Návrh riešenia

Navrhovaná aplikácia pozostáva z dvoch hlavných častí:

- serverová časť
- klientská časť

V nasledujúcej kapitole bude rozobraná funkcia jednotlivých častí aplikácie a spôsob, akým medzi sebou komunikujú.

4.1. Server

Server je v našej aplikácii zariadenie (počítač), ktorý komunikuje s projektorom a pomocou neho zobrazuje prezentáciu. Zároveň prijíma požiadavky od klientov a reaguje na ne.

Ešte pred prvým spustením prezentácie, prezentujúci použije desktopovú aplikáciu servera na definovanie:

- zoznamu súborov, ktoré sa budú prezentovať (a poradie, v akom sa budú prezentovať)
- definuje bezpečnostné heslo, ktorým sa budú klienti autentifikovať voči serveru
- povolí klientom pripájať sa na server
- vyberie práve jedného klienta, ktorý bude pôsobiť ako tzv. *master (master* je klient, ktorý momentálne ovláda prezentáciu)
- natrvalo zakáže vybraným klientom pripájať sa na server
- nastaví prípadné doplňujúce informácie pre server: na ktorej obrazovke počítača sa bude zobrazovať prezentácia (v prípade, že projektor je zapojený do počítača v rôznych zobrazovacích režimoch); definuje podobu tzv. *pause screen*, t.j. obrazovka, ktorá sa zobrazí, keď master zastaví prezentáciu alebo ešte pred spustením prezentovania prvého súboru

Jedine ak je vybraný aspoň jeden súbor na prezentovanie a je určený *master*, môže byť spustený režim prezentovania. Spustením tohto režimu sa najprv na projektore zobrazí vyššie spomínaný *pause screen*. Od tohto okamžiku má master možnosť ovládať prezentáciu a server slúži len na interpretovanie a vykonávanie jeho príkazov.

4.1.1. Požiadavky na server

Server musí byť prepojený s projektorom a musí mať aktivované sieťové rozhranie, ktorým bude komunikovať s klientami (Wi-Fi, Bluetooth,...). Server musí disponovať nainštalovaným JRE vo verzií 1.6 a vyššie. Ak má server slúžiť na prehrávanie Microsoft Powerpoint (formáty ppt, pptx, pps, ppsx, potm, pot) alebo OpenDocument formáty [5], ktoré podporuje OpenOffice Impress (resp. LibreOffice Impress), napr. odt, sda, sdp, vor, a ďalšie.

4.2. Klient

Klientská aplikácia sa pripája na server a posiela mu požiadavky na ovládanie prezentácie, pričom platí, že vždy len jeden klient môže takýmto spôsobom ovládať prezentáciu. Takýto klient sa nazýva *master*. Klient sa môže stať *masterom* buď tak, že je nastavený ešte pred spustením prezentácie alebo už počas prezentácie, ak sa súčasný *master* „vzdá“ svojej funkcie. Ak dôjde k situácii, že súčasný *master* prestane vykonávať svoju funkciu, všetci pripojení klienti dostanú o tejto skutočnosti informáciu a prvý klient, ktorí pošle požiadavku serveru, že chce byť *masterom* sa ním automaticky stane. Všetci ostatní klienti dostanú o tejto novej situácii informáciu.

Takto navrhnutý princíp „volieb“ nového *mastera* vychádza z predpokladu, že prezentujúci sa na začiatku osobne dohodnú na poradí, v akom budú prezentovať, takže k súperenie o pozíciu *mastera* by nemalo nastať.

Navrhovaná aplikácia bude tiež podporovať rezervovanie si pozíciu *mastera* dopredu – ešte skôr, ako má dôjsť k „voľbám“. V čase, keď je *master* ešte aktívny, ľubovoľný iný klient môže poslať *masterovi* požiadavku, že práve on chce byť novým *masterom*. Súčasný master môže túto požiadavku:

- potvrdiť – súčasný *master* sa okamžite vzdá svojej funkcie a novým *masterom* je klient, ktorý o to požiadal
- odložiť na neskôr – súčasný *master* ostáva *masterom* až do momentu, kým sa nevzdá svojej funkcie, novým *masterom* sa následne automaticky stane klient, ktorý si túto pozíciu „rezervoval“
- odmietnuť požiadavku – nedôjde k žiadnej rezervácii alebo zmene *mastera*

Každý klient sa musí najprv autorizovať voči serveru – poskytne mu svoju MAC adresu, názov stroja a heslo. K úspešnej autorizácii dôjde len v prípade ak sú splnené všetky nasledovné požiadavky:

- klient nie je zablokovaný a teda nie je mu dovolené sa autorizovať
- heslo, ktorým sa klient autorizuje, je zhodné s heslom nastaveným na strane servera
- klient s danou MAC adresou ešte nebol autorizovaný

V prípade, že prebehne úspešná autorizácia, server vygeneruje jedinečný identifikátor, ktorý pošle klientovi. Tento identifikátor bude klient používať vždy, keď bude posielať serveru ľubovoľný príkaz a bude slúžiť na autentifikáciu klienta voči serveru. Na generovanie jedinečného identifikátora bude použitý algoritmus na tvorbu Universally Unique Identifier (UUID) typu 4 [6].

4.3. Komunikácia klient-server

V nasledovnej kapitole bude opísaný spôsob, akým klient a server spolu komunikujú. Komunikácia môže prebiehať počas prezentácie (posielanie navigačných príkazov a pod.) alebo ešte pred samotným začatím prezentácie.

4.3.1. Nájdenie servera

Na to, aby mohol klient komunikovať so serverom, je potrebné, aby došlo k vytvoreniu spojenia medzi nimi. Klient je v tomto prípade aktívne zariadenie, ktoré vyšle

UDP paket ako broadcast, ktorým hľadá všetky dostupné *serveri.* Server reaguje na prijatý UDP paket odoslaním TCP paketu, z ktorého klient zistí IP adresu servera. V prípade, že sa takto identifikuje viacero serverov, klient má možnosť vybrať si server, na ktorý sa má pripojiť.

4.3.2. Komunikácia počas prezentácie

Na posielanie príkazov od klienta k serveru, prípadne občasných informácii posielaných od servera, bude použitý formát JavaScript Object Notation (ďalej len *JSON)* [7]. Správa vo formáte JSON bude posielaná cez TCP spojenie prostredníctvom Wi-Fi, Bluetooth alebo iného spôsobu, ktorý podporuje server aj klient súčasne (fyzickým prenosom dát sa naša aplikácia nezaoberá, aplikácia potrebuje len prístup k TCP resp. UDP protokolom). Formát JSON je vhodný na prenos veľkého množstva dát v jednoduchom formáte [8]. V porovnaní s napríklad XML formátom prináša JSON úsporu prenesených dát, ktoré sú potrebné na XML značky (*tagy*). Použitie formátu JSON prináša aj ďalšie výhody – rýchle a nenáročné spracovanie na strane servera a klienta, otvorenosť formátu, platformová nezávislosť a dostupnosť množstva knižníc tretích strán, ktoré zjednodušia proces implementácie.

Posielanie príkazov od klienta k serveru prebieha v piatich základných krokoch:

1. klient vygeneruje JSON správu podľa vopred daných pravidiel a pošle ju serveru
2. server zachytí správu; podľa tých istých pravidiel správu interpretuje ako určitý druh príkazu
3. server vykoná príkaz
4. server vygeneruje a klientovi odošle odpoveď na príkaz (stavový kód vykonaného príkazu, náhľad prezentácie ako obrázok alebo informácie, ktoré si klient pýta a pod.); odpoveď má opäť JSON podobu a je vytvorené podľa určitých pravidiel
5. klient príjme odpoveď od servera a interpretuje ju.

Vďaka použitiu asynchrónnych JSON správ je komunikácia medzi klientom a serverom, čiže klient medzitým, čo čaká na odpoveď zo strany servera, môže pokojne pokračovať vo vykonávaní ďalších používateľových požiadaviek.

4.4. Prezentovanie

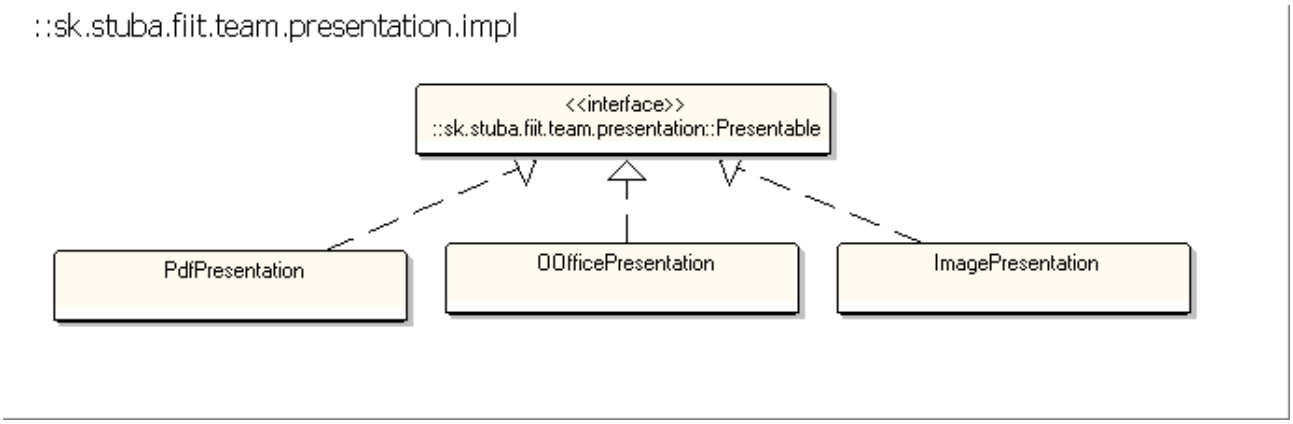
Proces prezentovania bude prebiehať na strane servera. Server bude posielať náhľady prezentácie na klientský stroj, ktorý je v stave *master*. Náhľady budú zmenšené na rozmery zodpovedajúce rozlíšeniu na strane klienta, budú zakódované do formátu JPEG a budú sa posielať ako pole bajtov.

Navrhovaná aplikácia bude podporovať 3 základné druhy prezentácii:

1. prezentácie podporované programom OpenOffice Impress (resp. LibreOffice Impress)
    - server spustí inštanciu programu OpenOffice (resp. LibreOffice) a bude s ním komunikovať prostredníctvom jeho programátorského rozhrania (UNO API [9]).
2. PDF dokumenty
    - bude použitý Swinglabs komponenta „PDF renderer“ na vykresľovanie PDF snímkov a knižnica Apache PDFBox na konverziu PDF dokumentov do textovej podoby
3. obrázky
    - obrázky budú zobrazované a bude s nimi manipulované štandardným spôsobom, aký poskytuje jazyk Java v balíku `java.awt.image` (t.j. triedy *Image*, *BufferedImage*, `AffineTransform`, a pod.)

Na zabezpečenie väčšej flexibility, udržiavateľnosti a schopnosti prispôsobiť sa potenciálnemu rozšíreniu o ďalšie prezentované typy súborov, sú tieto tri typy prezentácii reprezentované nasledovným spôsobom: exituje jednotné rozhranie *Presentable* (viď Obrázok 5), ktoré definuje základnú sadu príkazov, ktoré má podporovať každý typ prezentovaného súboru (napr. spusti prezentáciu, prechod na ďalší snímok, pauza, priblíženie, …). Zároveň je ale potrebné si uvedomiť, že napríklad prezentácia obrázku nie je schopná

vykonať príkaz na prechod na ďalší snímok. V takýchto prípadoch je odpoveďou na volanie tohto príkazu, vznik výnimky. Rozhranie *Presentable* teda definuje celkovú množinu príkazov, ale nie všetky príkazy sú schopné vykonať sa.



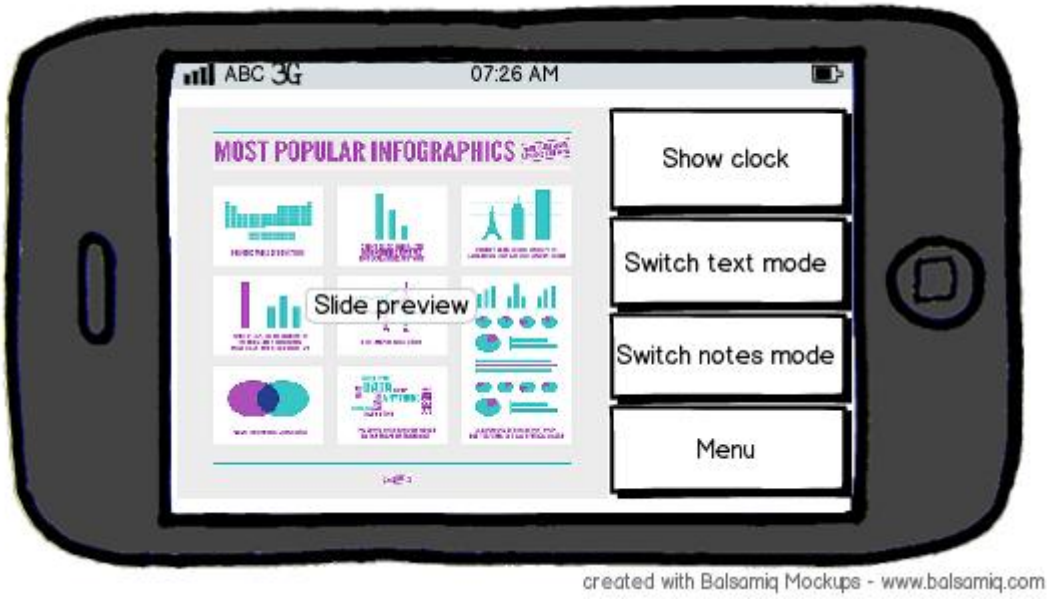
Obrázok 5 - Diagram navrhovaných tried prezentačnej časti aplikácie

4.4.1. Módy prezentovania

Navrhovaná aplikácia bude na strane klienta podporovať 4 módy (server a s ním spojená prezentácia nie je týmito módmi ovplyvnená):

1. prezentačný mód
    - klient zobrazuje náhľad prezentácie presne tak, ako ju zobrazuje server
2. textový mód
    - aktuálny snímok, ktorý sa zobrazuje je transformovaný do textovej podoby
3. mód pohybu
    - používateľ má možnosť pohybovať sa po snímke rôznymi smermi; tento mód je automaticky aktivovaný, keď používateľ priblíži snímok
4. súborový mód
    - špeciálny mód, kedy môže používateľ na klientovi zvoliť súbor, ktorý sa bude prehrávať; automaticky sa prehráva nasledujúci resp. predošlý súbor – týmto módom môže používateľ prehrávať ľubovoľný súbor, ktorý bol vopred definovaný na serveri

Na aktivovane určitého módu používateľ (na strane klienta) stlačením príslušného tlačidla vyšle príkaz serveru. Výnimkou je mód pohybu, ktorý sa aktivuje automaticky. Server následne odpovie s požadovanými informáciami – napr. ak klient nastaví textový mód, server spracuje prezeranú snímku do textovej podoby a pošle ju klientovi.



Obrázok 6 - Schématický nákres GUI klientskej aplikácie

5. Zmeny v návrhu

Počas procesu implementácie sme sa stretli s nepredvídateľnými problémami, kvôli ktorým sme boli nútení zmeniť pôvodný návrh nášho riešenia.

5.1. Server

Naším pôvodným zámerom bolo použiť program OpenOffice (alebo LibreOffice) v kombinácii s jazykom UNO na prezentovanie prezentácií vo formátoch ppt, pptx, odt a pod. Použitie tohto programu však nie je možné, keďže jazyk UNO na platforme Windows neposkytuje základnú funkciu na prezentovanie – funkciu na vrátenie sa o snímok späť. Preto sme sa rozhodli nepoužívať OpenOffice/LibreOffice, ale na prezentovanie tohto typu súborov použiť Java knižnicu Apache POI, ktorá dokáže zo súborov typu ppt, pptx, odt a pod. vztvoriť obrázky. Následne bude server pracovať s týmito obrázkami a zobrazovať ich používateľovi.

Výhodou tohto riešenia je tiež zbavenie sa zložitých závislostí, ktoré by musel používateľ nainštalovať, aby mohol využívať náš program.

5.2. Klient

Podľa pôvodného návrhu pre klientske aplikácie sa pri ich autorizácii voči serveru mala využívať MAC adresa daného zariadenia. Pri implementácii na jednotlivé platformy však vznikol problém so získavaním MAC adresy pre jednotlivé rozhrania, preto sme zvolili namiesto MAC adresy IMEI (International Mobile Equipment Identity). Jeho použitie je vhodnejšie aj preto, že jednoznačne identifikuje dané mobilné zariadenia, nie jednotlivé komunikačné rozhrania.

6. Implementácia

V nasledovnej kapitole sa budeme zaoberať procesom implementácie nášho projektu. Zvlášť budeme opisovať implementáciu oboch typov klientov (Android a Bada) a serverovej časti.

6.1. Implementácia serverovej časti

Implementácia serverovej časti bola rozdelená do troch nezávislých častí:

1. správa súborov,
2. modul na komunikáciu cez SOAP webservice,
3. modul na komunikáciu posielaním správ vo formáte JSON,
4. modul na úvodnú UDP komunikáciu.

6.1.1. Správa súborov

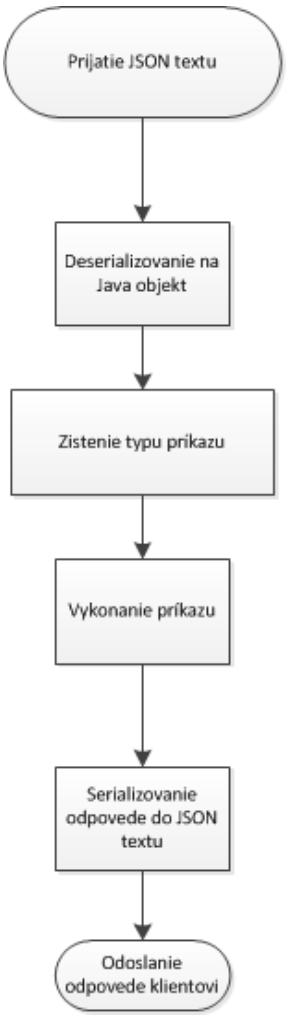
Používateľ si v grafickom rozhraní vyberie súbory, ktoré bude chcieť počas svojej prezentácie použiť. Následne je nutné tieto súbory identifikovať, či sa jedná o PDF súbor, obrázok alebo o prezentáciu typu Powerpoint (a jej podobné typy). Táto identifikácia je uskutočňovaná na základe prípony v názve súboru – PDF súbory končia na .pdf, obrázky môžu končiť na jpg, png, bmp, gif a pod. a rovnako prezentácie Powerpointového typu majú obmedzenú množinu povolených koncoviek. Tento spôsob rozoznávania typov súborov považujeme za dostatočný, keďže aj OS Windows rozoznáva súbory podľa koncoviek a aj na platforme Linux je zvykom (nie však nutnosťou) takéto typy súborov označovať príslušnými koncovkami.

6.1.2. Modul na komunikáciu cez SOAP webservice

Technológia SOAP webservice má v jazyku Java priamu podporu od verzie Java 6 a vyššie pod názvom JAX-WS, pričom v našom projekte sme túto podporu využili. Vďaka nej môže byť volanie webservice metód pre programátora úplne transparentné.

6.1.3. Modul na komunikáciu posielaním správ vo formáte JSON

Na spracovávanie prijatých správ vo formáte JSON a následné posielanie odpovedí v tomto formáte, bola použitá knižnica *google-gson*, ktorá dokáže serializovať Java objekt do text vo formáte JSON a opačne (z JSON textu do Java objektu). Na obrázku je zobrazený postup pri vykonávaní príkazov, ktoré server dostane od klienta vo formáte JSON.



Obrázok 7 - Spracovanie príkazov vo formáte JSON

Zistenie typu príkazu sa vykonáva na základe identifikačného čísla príkazu. Každý príkaz má toto číslo jedinečné a musí byť súčasťou každého JSON textu (príkazu na server alebo odpovede zo servera).

Príkaz sa vykonáva využitím návrhového vzoru adaptér (viď literatúra [10]) vzhľadom na webservice metódy. Vďaka tomuto riešeniu došlo k efektívnemu znovu-použitiu už napísaného a otestovaného kódu.

6.1.4. Modul na úvodnú UDP komunikáciu

UDP komunikácia medzi serverom a klientom (klientami) slúži primárne na zistenie IP adresy servera – túto IP adresu potrebujú zistiť klientské aplikácie, aby sa na server vedeli pripojiť cez webservice alebo socketovým spojením (ak používajú JSON príkazy). Zároveň server potrebuje vedieť svoju vlastnú IP adresu, aby dokázal spustiť webservice server. Server nedokáže zistiť túto IP adresu žiadnym iným spôsobom, ako tak, že sa ju dozvie od klienta – je možné zistiť IP adresy všetkých sieťových rozhraní servera, ale programátorsky nie je možné zistiť, na akom sieťovom rozhraní bude prebiehať komunikácia (WiFi rozhranie, Bluetooth rozhranie,....). Akonáhle klient zistí IP adresu severa, UDP komunikácia už nebude viac použitá.

Počas úvodnej UDP komunikácie je použitý nami navrhnutý protokol, kde prvý bajt dátovej časti UDP paketu je identifikátorom typu paketu:

Typ správy 1

Jedná sa o prvé pripojenie klienta. Súčasťou UDP dát je MD5 hash hodnota hesla zo strany klienta. Server porovná túto hash hodnotu s hash hodnotou jeho hesla – ak dôjde k zhode, vytvorí MD5 hash z hesla, mena klienta (ktoré je tiež súčasťou prijatého UDP paketu) a reťazca ACK. Ak nedôjde k zhode, namiesto reťazca ACK sa použije reťazec DENY.

Typ správy 2

Účelom tejto správy je informovanie servera o jeho IP adrese. Ak IP adresa ešte nebola nastavená (t.j. na server sa pripojil prvý klient), zároveň so zistením IP adresy servera dôjde k spusteniu webservice servera. Toto spustenie môže skončiť chybou – o nej sa klient dozvie tak, že server odpovedá UDP správou typu 3.

Typ správy 3

Jedná sa o odpoveď servera po spustení webservice servera. Odpoveďou môže byť 1 bajt s hodnotou 0, ak nedošlo k chybe a webservice server je úspešne spustený alebo nenulová hodnota, ak sa nepodarilo webservice server spustiť.

6.2. Implementácia klienta pre platformu Android

Aplikácia pre platformu Andorid je vyvíjaná v jazyku Java, na ktorom je táto platforma založená. Klientsku aplikáciu je možné rozdeliť do dvoch nezávislých častí:

- komunikácia so serverom
- grafické rozhranie a ovládanie aplikácie

6.2.1. Komunikácia so serverom

Pre vyhľadanie serverov v sieti a úvodnú komunikáciu s nimi sa využíva UDP komunikácia popísaná v rámci implementácie serveru v kapitole 6.1.4. Následne je na komunikáciu počas behu aplikácie využívaný modul serveru pre SOAP webservice popísaný v kapitole 6.1.2 implementácie serveru. Na rozdiel od priamej podpory webservice ktorú poskytuje Java pre server, pre klientsku aplikáciu na platforme Android táto podpora nie je natívna, bolo nutné využívať knižnicu kSOAP2 vo verzii 2.5.8 [11].

6.2.2. Ovládanie aplikácie

Jediným rozhraním pre ovládanie klientskej aplikácie je dotykový displej daného zariadenia. Tu je možné pohybmi po displeji ovládať prezentáciu, alebo pomocou menu zobrazeného na pravom okraji obrazovky využívať rozšírenú funkcionalitu. Presnejší popis je uvádzaný v používateľskej príručke. Pred samotným ovládaním prezentácie je potrebné potvrdiť resp. vyplniť autentifikačné dáta pre server zobrazené v okne klientskej aplikácie. Grafické rozhrania klientov na platformách Android a Bada sú vzhľadovo veľmi príbuzné a ich ovládanie si zachováva intuitívnosť a jednoduchosť.

6.3. Implementácia klienta pre platformu Bada

Implementácia aplikácie pre platformu Bada bola veľmi náročná a často krát sa menila z dôvodov, že vývojové prostredie nepodporovalo prostriedky stanovené v našej špecifikácii alebo návrhu, či nepodporovalo prostriedky, ktoré mala stanovené vo svojom rozhraní pre programovanie (API).

Aplikácia bola pôvodne navrhovaná všeobecne, t.j. s podporou všetkých verzií platformy Bada. Avšak neskôr vo fáze implementácie sa ukázalo lepšie použiť novšiu verziu (2.0). Tá poskytuje podporu viacerých prostriedkov, ktoré sme využili pri implementácii. Jednou z nich je napríklad explicitná podpora použitia štandardu JSON, ktorý je opísaný v návrhu riešenia. Avšak použitím novšej verzie sa vyskytlo aj niekoľko nečakaných problémov. Príkladom je nemožnosť získania MAC adresy. Táto funkcia bola podporovaná v predchádzajúcej verzii, avšak (že vraj) z bezpečnostných dôvodov bola pre potreby novšej verzie jej podpora odstránená. Je vhodné podotknúť, že táto informácia sa nenachádza nikde na oficiálnych stránkach, a ani API vývojového prostredia Bada.

Ďalším vážnym problémom ktorý sa vyskytol pri implementácií bola závislosť použitia niektorých funkcií a prostriedkov na tzv. privilégiách. Tieto privilégiá sú definované v manifeste aplikácie. V podstate ide o akési vymenovanie prostriedkov, ktoré sa majú povoliť pre potreby aplikácie. Ich použitie je potom potrebné používateľom potvrdiť pri inštalácii aplikácie. Avšak samotným problémom je, že tak ako API, tak aj vývojové prostredie (SDK) nedostatočne informuje programátora o ich použití a teda často sa stáva, že aplikácia pri testovaní sa správa nekorektne alebo padá z neznámych príčin. Na odstránení

tohto problému sa však momentálne pracuje, čo je vidieť v API, kde pri niektorých dôležitých funkciách pribudla informácia o nutnosti pridania príslušného privilégia. Najlepším zdrojov informácií o použití spomínaných privilégií ale aj tak zostávajú rôzne návody a manuály, ktoré sú sprevádzané príspevkami a komentármi programátorov, ktorí sa zväčša stretli s týmto problémom a s radosťou sa podelia o svoje cenné rady a skúsenosti.

Ďalšie problémy, ktoré sa pri implementácii vyskytli sú opísané v nasledujúcich častiach dokumentácie.

Samotná implementácia klientskej časti pre platformu Bada sa dá rozdeliť do niekoľkých samostatných častí:

- komunikačné prostredie
- prihlasovacia obrazovka
- hlavná obrazovka

6.3.1. Komunikačné prostredie

Pre komunikačné prostredie bol vybratý transportný protokol TCP. Jeho výhodou je potvrdzovanie prijímania dát, to znamená, že máme zaručené, že dáta sa prijmú korektne. Výber protokolu taktiež ovplyvnil fakt, že v samotnom vývojovom prostredí je protokol TCP viac podporovaný a ponúka viac možností ako alternatívny UDP protokol.

Komunikácia prebieha pomocou výmeny socketov. Socket je všeobecný mechanizmus pre komunikáciu, ktorý má pridelený jedinečný deskriptor. Je to v podstate kombinácia IP adresy a k nej prislúchajúcemu číslu portu. Pomocou tohto socketu sa vytvorí spojenie medzi klientskou stanicou a serverovou aplikáciou, ktoré sa používa sa na výmenu dát. Spojenie je definované IP adresou a špecifickým aplikačným portom, ktorý je v našej aplikácií implementovaný staticky, to znamená, že klient nemá možnosť meniť toto číslo portu. Nevýhodou tohto riešenia je, že niektorí používatelia môžu mať zablokované niektoré porty a medzi nimi aj náš vybraný port. Preto sme sa rozhodli zvoliť číslo portu, ktoré nie je používané bežnými aplikáciami komunikujúcimi po sieti, čím sme sa snažili minimalizovať efekt tejto nevýhody. IP adresa sa zadaná manuálne hneď po spustení aplikácie v prihlasovacej obrazovke.

Na identifikáciu stanice, resp. aplikácie sa používa jednoznačný identifikátor. V pôvodnom návrhu bolo uvedené, že identifikátor bude predstavovať MAC adresa zariadenia. Avšak ako už bolo spomenuté, pri implementácií bolo zistené, že túto funkciu novšia verzia platformy Bada nepodporuje, a teda bol tento identifikátor nahradený iným - IMEI číslom, ktoré sa používa na identifikáciu telefónu zvyčajne telefónnym, internetovým alebo satelitným operátorom. Toto číslo je zvyčajne jedinečné a teda spĺňa naše potreby pre identifikáciu zariadenia. Identifikátor sa používa na identifikáciu zariadenia serverom. Server na základe tejto informácie vie, ktorému zariadeniu má posielať dáta. Pre sprístupnenie IMEI čísla je potrebné pridať privilégiá *SYTEM_SERVICE* a knižnica *FSysSystemInfo*. Zaujímavosťou je, že podľa oficiálnej stránky, tak ako aj API portálu nie je IMEI číslo k dispozícii. Napriek tomu sa dá získať, aj keď trochu neštandardným, systémovým volaním príslušnej metódy spomenutej knižnice.

Pre socketovú komunikáciu bola použitá knižnica *FNet*, ktorá obsahuje rozhranie *ISocketEventListener*. Pomocou neho vieme posielať a odchytávať dáta posielané zo servera. Tieto dáta sú posielané vo formáte štandardu JSON. Taktiež bolo nutné pridať privilégiá *WIFI* a *SOCKET*.

Najskôr sa počítalo, že na posielanie dát sa bude využívať, podobne ako pre platformu Android našej aplikácie, knižnica gSOAP. Avšak ukázalo sa, že Bada neposkytuje všetky možnosť programovacieho balíka C++, medzi ktorými je aj podpora niektorých dôležitých externých knižníc pre používanie SOAP mechanizmu. Jedinou možnosťou by bolo importovanie týchto knižníc manuálne, avšak toto riešenie sa nám nezdalo ako štandardné a ani sa nám to nepodarilo implementovať, preto sme sa rozhodli pre použitie mechanizmu posielania JSON objektov.

Na posielanie dát vo formáte JSON je potrebná knižnica *FWebJson*. Táto knižnica je dostupná len pre novšiu verziu platformy Bada (t.j. 2.0 a vyššie). Vďaka nej vieme jednoducho spracovať prijaté údaje. Keďže JSON je v podstate nič iné ako naformátovaný text podľa istého štandardu, rozhodli sme sa pri vytváraní odosielaných dát nepoužiť spomínanú knižnicu pracujúcu s JSON objektmi, ale vytvárame tieto dáta manuálne. Dovoľuje nám to jednoduchosť našej aplikácie. Zvýšime tak rýchlosť vykonávania sa, pretože sa vyhneme zbytočnému vytváraniu JSON objektov.

7. Zhodnotenie

Výsledkom nášho je aplikácia uložená na Google Play a Samsung Apps portáloch, ktorá slúži na pohodlné prezentovanie s využitím smartphone telefónov na platformách Andoid a Bada. Vzhľadom na malý počet členov nášho tímu a problémy s vývojom klientskej aplikácie na platforme Bada, sme boli nútení niektoré pôvodne zamýšľané funkcie odstrániť (napr. približovanie strán prezentácie alebo písanie do prezentácie). Nemyslíme si však, že by sa tým zásadným spôsobom znížila použiteľnosť našej aplikácie.

8. Zdroje

[1] Android PowerPoint Remote Control. 10.11.2011. PPT Remote
http://pptremotecontrol.com

[2] v Ralle v. 10.11.2011. PowerPoint OpenOffice Remote
http://www.vrallev.net/do/apps/android/pptodp_remote/start

[3] Senstic. 10.11.2011. i-Clickr PowerPoint Remote
http://www.senstic.com/iphone/iclickr

[4] Youtube. 10.11.2011. remotePPT bada
http://www.youtube.com/watch?v=awwiJE4Q2gM

[5] Oasis. 10.11.2011. Open Document Format for Office Applications
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office

[6] IETF. 10.11.2011. rfc4122
http://www.ietf.org/rfc/rfc4122.txt

[7] JSON. 10.11.2011. Introducing JOSN
http://www.json.org/

[8] myDevNotes. 10.11.2011. JOSN vs XML
http://xphone.me/devnotes/2011/02/json-vs-xml-part-1-data-size/

[9] OpenOffice. 10.11.2011. UNO project
http://udk.openoffice.org/

[10] Adapter Design Pattern. 20.4. 2012
http://sourcemaking.com/design_patterns/adapter

[11] kSOAP2 - android. 22.4.2012
http://code.google.com/p/ksoap2-android/