

České vysoké učení technické v Praze  
Fakulta elektrotechnická

katedra počítačů

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Roman Fürst**

Studijní program: Softwarové technologie a management  
Obor: Softwarové inženýrství

Název tématu: **Měření a vizualizace signálu bezdrátových sítí v interiéru**

Pokyny pro vypracování:

Seznamte se s vývojem aplikací pro OS Android. Vytvořte aplikaci pro měření a vizualizaci pokrytí vnitřku budov bezdrátovými sítěmi Wi-Fi. Cílem aplikace bude určit pokrytí různých míst budovy bezdrátovými sítěmi.

Aplikace bude schopna zobrazit seznam sítí a k nim náležejícím přístupové body, ke kterým bude vztaženo měření. Po vybrání jednoho nebo více bodů spustí aplikaci detailní měření včetně agregovaného zobrazení. Dle specifik vývoje použijte buď některou knihovnu pro dynamickou tvorbu grafů, nebo vytvořte vlastní funkci zobrazení grafu. Provedená měření budou perzistentní s možností exportu (např. do Matlab). Aplikace bude pracovat s jednoduchým plánem podlaží budovy, měření bude možno vztáhnout k různým místům.

Vývoj aplikace provádějte v souladu s obecně platnými pravidly pro zajištění kvalitního kódu, dokumentace a testování. Ověřte funkčnost aplikace změřením dostupných sítí v reálném prostředí.

Seznam odborné literatury:

Allen, Grant. Android 4: průvodce programováním mobilních aplikací. Computer Press, 2013

Yaghmour, Karim. Embedded Android. O'Reilly, 2013

Stallings, William. Wireless communications and networks. Pearson, 2014

Pechač, Pavel. Modely šíření vln v zástavbě: [modely pro plánování mobilních rádiových systémů : WLAN, WiFi, DECT, GSM, UMTS]. BEN - technická literatura, 2005

Lee, Byeong Gi. Broadband wireless access and local networks: mobile WiMax and WiFi. Artech House, 2008

Vedoucí: Ing. Jakub Doležal

Platnost zadání: do konce zimního semestru 2016/2017



doc. Ing. Filip Železný, Ph.D.  
vedoucí katedry



prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 8. 10. 2015





**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

Fakulta elektrotechnická

Katedra počítačů

## **Měření a vizualizace signálu bezdrátových sítí v interiéru**

Bakalářská práce

Studijní program: Softwarové technologie a management

Studijní obor: Softwarové inženýrství

Vedoucí práce: Ing. Jakub Doležal

**Roman Fürst**

---

Praha 2015



## **Poděkování**

Poděkování patří Ing. Jakubu Doležalovi, vedoucímu této bakalářské práce, za odborné vedení, věcné i praktické rady během konzultací, a vstřícnost během vypracovávání této bakalářské práce. Dále bych chtěl vyjádřit obrovské poděkování své rodině za vytvoření podmínek ke studiu.



## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů pro přípravu vysokoškolských závěrečných prací.

V Praze dne 11.1.2016

Podpis .....





## **Abstrakt**

Práce se zabývá studiem, analýzou a popisem způsobu měření, vizualizace a využití signálu bezdrátových sítí v interiéru. Pro tyto účely je provedena implementace mobilní aplikace pro Android, která zobrazuje dostupné bezdrátové sítě v okolí, graficky interpretuje a agreguje jejich signál, a naměřená data pak ukládá do csv souboru. Následně proběhlo testování funkčnosti aplikace.

## **Abstract**

This thesis' goal is to deliver evaluation tool for wireless networks in indoor environments. For this purpose a novel mobile application for Android was designed and implemented. The application is capable to detect all available wireless networks and to conduct signal strength measurements on selected subset of networks. The most significant benefit of the application is the capability to render measurements visible through linear, aggregated and heatmap charts



## **Klíčová slova**

Android, mobilní aplikace, vizualizace, bezdrátové sítě, síla signálu

## **Key words**

Android, mobile application, visualization, wireless networks, signal strength



# Obsah

<b>1. ÚVOD .....</b>	<b>15</b>
1.1 ÚVOD DO PROBLEMATIKY .....	15
1.2 DEFINICE PROBLÉMU .....	16
1.3 O PROJEKTU .....	16
<b>2. PROBLEMATIKA BEZDRÁTOVÝCH SÍTÍ .....</b>	<b>17</b>
2.1 PRINCIP FUNGOVÁNÍ WIFI .....	17
2.1.1 Základní pojmy .....	17
2.1.2 Standard IEEE 802.11 .....	17
2.1.3 Kolize .....	18
2.1.4 Bezpečnost .....	19
2.2 INTERPRETACE SIGNÁLU .....	21
2.3 VLIV PROSTŘEDÍ NA ŠÍŘENÍ SIGNÁLU .....	23
<b>3. DOSTUPNÝ SOFTWARE PRO MĚŘENÍ A VIZUALIZACI SIGNÁLU .....</b>	<b>25</b>
3.1 UŽIVATELSKÉ APLIKACE .....	25
3.1.1 WiFi Signal Strength .....	25
3.1.2 WiFi Analyzer .....	26
3.2 KNIHOVNY PRO VIZUALIZACI .....	26
3.3 VÝBĚR PLATFORMY A JAZYKA PRO BP .....	27
3.3.1 Přehled platforem .....	27
<b>4. SOFTWAROVÁ ANALÝZA .....</b>	<b>29</b>
4.1 POŽADAVKY .....	29
4.1.1 Funkční .....	29
4.1.2 Nefunkční .....	29
4.2 SCÉNÁŘE UŽITÍ .....	30
4.3 DIAGRAM AKTIVIT .....	35
4.4 STAVOVÝ DIAGRAM .....	37
4.4.1 NetworkListActivity .....	37
4.4.2 MeasureActivity .....	38
4.5 DOMÉNOVÝ MODEL .....	39
4.5.1 NetworkListActivity .....	39
4.5.2 MeasureActivity .....	40
<b>5. IMPLEMENTACE .....</b>	<b>41</b>
5.1 PRŮBĚH IMPLEMENTACE .....	41
5.1.2 GraphView .....	42
5.1.3 HeatmapView .....	43
5.1.4 TouchEventHandler .....	45
5.2 POUŽITÉ KNIHOVNY .....	48
<b>6. TESTOVÁNÍ .....</b>	<b>49</b>
6.1 TESTOVÁNÍ ANDROID APLIKACÍ .....	49
6.1.1 Mockování .....	49
6.2 TESTY .....	50

6.2.1 Akceptační test case .....	50
6.2.2 Unit .....	53
6.3 SHRNU TÍ TESTŮ .....	53
<b>7. ZÁVĚR.....</b>	<b>55</b>
<b>LITERATURA .....</b>	<b>56</b>
<b>A. OBSAH PŘILOŽENÉHO CD .....</b>	<b>57</b>

# 1. Úvod

Tato bakalářská práce se věnuje tématu měření a vizualizace signálu bezdrátových sítí. Následující kapitola popisuje obsah a náplň práce, definuje konkrétní problematiku, určuje, co bude výstupem práce a jaký bude mít přínos. Stručně shrnuje problematiku úlohy a nastiňuje její řešení.

## 1.1 Úvod do problematiky

Pro měření signálu bezdrátových sítí lze najít mnoho praktických důvodů. Jedním z nejaktuálnějších témat, se kterým se měření bezdrátových sítí spojuje, je indoor navigace (indoor navigation) [1], neboli navigace uvnitř budov, obecně v uzavřených prostorách. Hlavní problém indoor navigace tkví v tom, že zařízení není schopné (nebo velmi špatně) navázat spojení s GPS [1] satelitem a určit tak svou polohu. K těmto účelům se dost často využívá potenciál WiFi [2] technologie, a to hlavně jejího základního parametru a tím jsou radiové vlny. Přesněji řečeno jejich intenzita. Princip řešení spočívá v tom, že na základě aktuální hodnoty přijímaného signálu může buď výpočtem nebo porovnáním s předem naměřenými daty, určit svou polohu vzhledem k vysílačům v okolí.

První metoda používá matematický výpočet pro určení vzdálenosti vysílač-přijímač dle naměřeného útlumu. Je-li takto k dispozici více vysílačů v okolí (router, přístupový bod), lze spočítat vzdálenosti ke každému z nich a tím určit polohu vůči těmto zařízením a tím i v prostoru. Předpokladem je, že poloha vysílačů je předem známá a statická.

Druhá varianta pracuje s předem naměřenými hodnotami, kdy probíhá měření na určitých pozicích po určitou dobu, a získané hodnoty se ukládají společně s pozicí, ke které se hodnota vztahuje. Čím více dat naměříme na hustším poli měřících míst, tím přesnější představu o rozložení signálu dostaneme. Takto naměřená data pak můžeme poskládat do takzvané RSSI (Received Signal Strength Indication) mapy [1]. V podstatě jde o databázi, mapu, seznam souřadnic míst a k nim příslušných hodnot síly signálu. Při pohybu v uzavřeném prostoru pak vezmeme aktuálně naměřené hodnoty ke všem vysílacím zařízením v dosahu a porovnáme údaje s předem sestavenou mapou. Z toho jsme pak schopni odvodit relativně přesně naši aktuální pozici vzhledem k vysílacím jednotkám v okolí.

Problematika se tedy dá rozdělit na dvě části. Nejdříve, v první části, je třeba sesbírat potřebná data a provést v širším časovém úseku co nejvíce měření na co největším počtu míst, poté údaje vhodně zpracovat a uložit. Následně, v druhé části, se uložená data mohou použít k sestavení mapy pro zjištění pokrytí bezdrátovými sítěmi, nebo k lokalizaci polohy uživatele ve spojení s dalšími technologiemi.

Tato práce dává základ první části problematiky týkající se indoor navigace [1]. Následným rozšířením v rámci dalších projektů by tak mohl vzniknout software, který bude mít plnohodnotné uplatnění při vývoji indoor navigace.

## 1.2 Definice problému

V úloze vědeckého charakteru, jako je například tato, jejíž součástí je měření údajů a sběr dat, by měl být velký důraz kladen na vhodnou interpretaci naměřených hodnot. Lepší uchopení a vizualizace naměřených dat nám pak pomůže lépe pochopit souvislosti a nahlédnout na řešení problému ze správného úhlu. Je proto vhodné navrhnout nástroj, který bude nejen kvalitně a spolehlivě měřit požadované údaje, ale stejně tak dobře je bude interpretovat a to tak aby byly pochopitelné i bez znalosti širších souvislostí tématu.

Během řešení této bakalářské práce bude tedy kladen důraz na vhodnou, přehlednou, srozumitelnou a jasnou vizualizaci měřených dat, a na to, aby aplikace nabídla pokročilejší způsob zobrazení než konkurenční aplikace podobného charakteru.

## 1.3 O projektu

Výsledkem této práce by měla být mobilní aplikace, která nabídne lepší způsob měření a hlavně vizualizace signálu bezdrátových sítí oproti ostatním aplikacím s podobným zaměřením. Oproti konkurenci by měla poskytovat širší funkcionalitu během vizualizace dat, a po ukončení měření nabídnout i možnost exportu získaných hodnot do CSV souboru. Rozsah projektu a aplikace je možné v budoucnu rozšířit a doplnit novou funkcionalitu. Postupně by se tak mohla stát plnohodnotným nástrojem pro vývoj indoor navigace.

Náplní toho projektu je rešerše na téma bezdrátových sítí, srovnání konkurenčních aplikací a grafových knihoven. Následně je provedena softwarová analýza, podle které vzniká implementace aplikace pro zvolenou platformu. Implementovány by měly být hlavně vizualizační algoritmy zobrazující signál v podobě lineárního grafu a heatmapy. Správná funkcionalita softwaru je poté otestována simulací běžného provozu.



## 2. Problematika bezdrátových sítí

Tato práce je specifická kombinací problematiky bezdrátových Wi-Fi sítí se softwarovým vývojem. Z tohoto důvodu tato kapitola vysvětluje průřezově všechny relevantní technologie nezbytné k porozumění a výsledné implementaci.

### 2.1 Princip fungování WiFi

#### 2.1.1 Základní pojmy

V dnešní době se na většině míst, ať již veřejných či privátních, nachází možnost bezdrátového připojení k síti. Proto je vhodné zmínit další z termínů ze světa WiFi [2], a tím je HotSpot [3], neboli místo, na kterém je k dispozici bezdrátová síť. Většina sítí dále vysílá takzvané SSID(Service Set Identifier) [3], což je název a identifikace dané sítě, díky kterému se zařízení může k této síti připojit. Ovšem abychom mohli zařízení (laptop, mobilní telefon, tablet, ...) připojit, musí obsahovat nezbytnou hardwarovou podporu zajištěnou bezdrátovým adaptérem. Pro připojení vícero zařízení do sítě nám poslouží router (směrovač). Router řídí výměnu paketů mezi sítěmi a podsítěmi překladem IP (Internet Protocol) adres na MAC [3] adresy. Tomuto procesu se říká routování (směrování). Setkat se můžeme také s pojmem Acces Point (AP) [3] neboli přístupový bod. Přístupový bod nám zprostředkuje připojení do sítě, ale neprovádí směrování jako router.

#### 2.1.2 Standard IEEE 802.11

Zkratka WiFi (Wireless Fidelity) se v informatice používá pro označení bezdrátové komunikace mezi zařízeními. Komunikace probíhá hlavně na základě standardu 802.11 [2] vyvinutým Institute of Electrical and Electronic Engineers (IEEE) [2]. Odtud se tedy bere známé označení IEEE 802.11. Standard 802.11 ve skutečnosti obsahuje celou sadu standardů, které se vyvíjeli na základě vylepšení předchozí verze nebo modifikace použitých technologií. Původně se pojem WiFi používal pouze pro standard s označením 802.11b, ale v dnešní době tato zkratka platí prakticky pro veškeré členy rodiny IEEE 802.11. V tabulce 2.1 můžeme vidět přehled verzí protokolu IEEE 802.11.

Jak lze z *tabulky 2.1* [2] vyčíst, přenosová rychlost se pohybuje od 1 Mbps až ke 140 Mbps. Kterou z možností si uživatel zvolí pro svou síť, záleží na jeho potřebách a možnostech.

Tabulka 2.1 : zobrazuje verze standardu IEEE 802.11 a jejich stručný popis [2]

Verze standartu	Stručný popis
802.11 1997	Původní specifikace podporující přenos 1 nebo 2 Mbps. Pracuje s frekvencí 2,4 GHz, a rychlost přenosu reguluje buď přeskokováním mezi frekvencemi (Frequency Hopping Spread Spectrum - FHSS) nebo pomocí techniky přímého rozprostřeného spektra (Direct Sequence Spread Spectrum - DSSS)
802.11a 1999	Poměrně častá modifikace standardu, podporuje přenosovou rychlost 54 Mbps využívající k tomu 5GHz frekvenci spolu s vylepšenou přenosovou technikou, tzv. Ortogonální Frekvenční Multiplex (Orthogonal Frequency Division Multiplexing - OFDM)
802.11b 1999	Zřejmě nejrozšířenější a neznámější verze, původně známá jako WiFi. Přenosová rychlost se může přizpůsobit špatnému připojení, ale standartní hodnota je 11 Mbps. Frekvence je 2,4 GHz s technikou přímého rozprostření spektra (DSSS)
802.11g 2003	Upgrade k 802.11a . Taktéž používá OFDM avšak rychlost na frekvenci 2,4 GHz dosahuje až 54 Mbps.
802.16 2001	Znám pod zkratkou WiMax. Důvodem je mnohem větší dosah oproti ostatním verzím. Frekvenční pásmo se pohybuje mezi 10-66 GHz a přenosová rychlost může být vyšší než 70 Mbps.
802.11n 2009	Zatím nejrychlejší standard využívající dvojnásobnou šířku pásma (40MHz) díky vícenásobnému streamování. Kódovací schéma je taktéž OFDM. Rychlost až 140 Mbps

### 2.1.3 Kolize

Pokud již máme navázané spojení a zařízení je připojeno k síti, bezdrátová komunikace se může setkat s jedním z mnoha problémů - takzvané kolizi [2]. Nejenom, že každá WiFi síť operuje na stejné frekvenci 2,4 nebo 5 GHz [3], ale i další zařízení jako například Bluetooth, mikrovlnné trouby, dálkové ovladače od garáže pracují na téže frekvenci, a proto nastává kolize, respektive dochází ke kolizi při vysílání v rámci stejného frekvenčního pásma. Tento problém můžeme vidat při používání radiokomunikačních zařízení, kdy jeden momentálně hovořící účastník blokuje frekvenci a tudíž znemožňuje komunikaci zbylým účastníkům. Nabízí se řešení pro každé zařízení určit vlastní frekvenční kanál, na kterém by komunikovalo. Tato varianta

má ovšem nevýhodou v tom, že velká část frekvenčního rozsahu by byla po většinu času nevyužitá. Lepší cesta je použít při přenosu dat celý frekvenční rozsah – tzv. rozprostřené spektrum [2]. Díky roztažení přenosu přes celé pásmo dojde při kolizi pouze na určitých frekvencích, a tudíž ke ztrátě jen části přenášených informací. Příjímač detekuje ztrátu a pošle požadavek o znovu poslání chybějících dat. Čím více vzájemně kolizních zařízení v okolí je, tím více kolizí nastane, následuje větší četnost ztrát a znovu poslání dat, a tím pádem pomalejší přenos informací. Přehled modulačních technik využívajících rozprostřené spektra můžeme vidět v tabulce 2.2 [2].

Tabulka 2.2: přehled modulačních technik rozprostřené spektra [2]

Název	Princip
Frequency Hopping(FH)	Příjímač a vysílač mění frekvenci v předem určením postupu
Direct Sequence(DS)	Matematicky určí rozložení dat ve frekvenčním pásmu
Orthogonal Frequency Division Multiplexing(OFDM)	Signál je vysílán na více vzájemně ortogonálních frekvencích

#### 2.1.4 Bezpečnost

Bezdrátové zabezpečení lze zvažovat ze dvou důvodů – zabezpečení a soukromí. První nám dává kontrolu nad tím, kdo, kdy a jak se může k síti přihlásit (tj. autentizace a autorizace), díky čemuž není třeba mít fyzickou kontrolu nad přípojkami, a druhé nám šifruje přenášená data, aby nikdo nepovolaný nemohl získat přístup o obsahu komunikace. Hovoříme-li o WiFi sítích můžeme jmenovat čtyři základní typy zabezpečení, viz tabulka 2.3 [4].

Tabulka 2.3: přehled zabezpečujících standardů a protokolů [4]

Typ zabezpečení	Stručný popis
WEP	Standard, který se vyskytuje buď v 64bitové nebo 128bitové verzi. Rozdíl je v délce hesla, které se používá k šifrování. WEP (Wired Equivalent Privacy) se považuje za velice slabé a snadno prolomitelné zabezpečení. Sdílený klíč WEP se dá snadno zjistit útokem na síť během několika okamžiků a proto se v dnešní době nahrazuje mnohem účinnějšími šifrovacími standardy.
WPA-PSK	WPA-PSK (WiFi Protected Acces – Preshared Key) pracuje se šifrovacím protokolem TKIP (Temporal Key Integrity Protocol). Využívá kryptografický algoritmus RC4 a na rozdíl od svého předchůdce, standardu WEP, využívá klíče o délce 256 bitů.
WPA2-PSK	Šifrování buď pomocí TKIP nebo pomocí novější AES (Advanced Encryption Standard). AES je symetrická šifra nahrazující DES (Data Encryption Standard). TKIP se kvůli slabé bezpečnosti používá jen u starších zařízení nepodporujících WPA-PSK (AES), což je prozatím nejbezpečnější varianta zabezpečení.
802.1x	Jedná se protokol obsluhující především přístup k síti, tedy autentizaci. V celém procesu autentizace máme tři účastníky – supplicant (klient), autentizátor (síťová aplikace) a autentizační server

Princip autentizačního protokolu 802.1x [5] pro ověřování klienta je pak následující:

1. Ihned po připojení klienta k portu je síťový port blokován a prochází jím pouze autentizační rámce.
2. Autentizátor (nejběžněji router) vyšle klientovi žádost o autentizaci. Klient žádost přijme, vyhodnotí a odpoví.
3. Autentizátor (router) zprávu přijme, zabalí do správného rámce a odešle autentizačnímu serveru. Autentizační server může být například RADIUS (Remote Authentication Dial-in User Service).
4. Autentizační server si pak přes autentizátor vyžádá od klienta autentizační údaje a pošle je opět přes autentizátor (router) zpět autentizačnímu serveru (RADIUS).

5. V případě, že klientem zadané údaje jsou validní, odešle autentizační server autentizátoru zprávu o povolení přístupu, ten ji vyhodnotí, odblokuje port a pošle klientovi hlášení o úspěšné autentizaci. V tuto chvíli je proces u konce.

## 2.2 Interpretace signálu

Všechna WiFi zařízení odesílají data standardizovaným protokolem 802.11 [2] a pro samotný přenos se využívá rádiových vln – čili data se šíří volným prostorem. Nosným médiem je tedy vzduch. Každé z těchto zařízení má odlišnou sílu při vysílání, a naopak citlivost pro přijímání rádiových vln. Síla rádiového signálu, který se přenáší éterem může být narušována mnoha okolními vlivy. Nejdůležitější pro kvalitní signál, jak již bylo zmíněno, je citlivost a výkon antény vysílače/přijímače, ovšem úbytek na síle signálu může mít na svědomí i například nábytek, zeď, dveře, okno a v neposlední řadě i člověk. Můžeme si uvést pár příkladů útlumů: sádrokartonová zeď nám odebere 4 dB z intenzity signálu, cihlová zeď už 8 dB a betonová 10-15 dB (k samotné definici decibelu se dostaneme později). Dokonce i samotný vzduch, kterým se vlny šíří, snižuje intenzitu signálu. Během cesty od vysílače k přijímači se část energie signálu ztratí ve vzduchu. Platí tedy pravidlo, že čím dále je přijímač od vysílače, tím slabší signál k němu dorazí. Tomuto jevu v angličtině říkáme Free Space Path Loss (ztráty při šíření otevřeným prostorem) [11] a dá se bez problému stanovit výpočtem. Samotná síla signálu se pak odborně označuje jako Signal Strength Indicator, častěji se však setkáme se zkratkou RSSI. Typicky by se při větším snížení signálu měla automaticky snížit rychlost přenosu dat, aby se udrželo stabilní spojení (kvůli velké chybovosti). To už je ale záležitost konkrétního hardwaru.

Pokud je řeč o samotné interpretaci síly signálu, neboli RSII [1], měli bychom určitě zmínit v jakých jednotkách se signál interpretuje. Nejčastěji se setkáme se jednotkou dBm [6], neboli decibel anebo mW, což je zkratka pro miliwatt. Mezi oběma jednotkami existuje matematický vztah, který popisuje následující vzorec [6]:

$$dBm = 10 * \log_{10}(mW) \quad (2.1)$$

Opačný vztah je popsán takto:

$$mW = 10 \left( \frac{dBm}{10} \right) \quad (2.2)$$

S první zmíněnou veličinou, decibelem, se v praxi zřejmě budeme setkávat častěji. Decibel je bezrozměrná jednotka a udává v podstatě jen poměr mezi dvěma veličinami.

V našem případě poměr mezi dvěma výkony. Písmeno  $m$  ve značení decibelu (dBm) indikuje, že se jedná o decibel vztažený na 1 miliwatt výkonu. Po dosazení do vzorce pro výpočet decibelu pak lze dojít k závěru, že bude-li naměřený výkon signálu větší než 1 miliwatt, výsledná hodnota decibelu bude kladná. V reálném světě je však hodnota menší než 1 miliwatt, tudíž se při měření signálu vždy setkáváme se zápornými hodnotami decibelu. Nulový decibel pak odpovídá naměřené hodnotě rovné přesně 1 miliwatt. Obecný vzorec decibelu:

$$dB = 10 * \log_{10} \frac{P_1}{P_2} \quad (2.3)$$

Do vzorce 2.3 [6] pak můžeme dosadit hodnotu ke které se decibel vztahuje, v našem případě 1mW, viz vzorec 2.4 [6].

$$dBm = 10 * \log_{10} \frac{P_1}{1mW} \quad (2.4)$$

Můžeme tedy uvést příklad, že pro naměřený výkon 0,001 mW budeme pracovat s hodnotou -30 dBm. Nicméně každý výrobce WiFi čipů může implementovat vlastní způsob pro zobrazení síly přijímaného signálu. Kupříkladu Cisco Systém [8] na svých kartách vrací hodnoty 0 až 100, pro značku Atheros [8] je to pak 0 až 127. Stejně tak každý z výrobců garantuje svou vlastní přesnost a citlivost svého čipu při měření aktuální síly signálu.

Do signálů vstupuje mnoho vedlejších rušivých elementů, které ho ovlivňují. Jedná se především o takzvaný šum. Šumem rozumíme jakékoli vedlejší rušivé rádiové vlny, které se přičtou k přenášenému signálu. Zdrojem pak může být okolní prostředí, přijímač, ale i samotné zařízení, které signál vysílá. Jedním se zdrojů může být například Bluetooth adaptér či přímo WiFi adaptér. Každé zařízení tak generuje svůj vlastní šum a při přijímání by mělo být schopné rozeznat, do jaké míry se jedná o šum a kdy už jde o plnohodnotný signál. V praxi se můžeme setkat s veličinou Signal-to-Noise Ration [2] ( poměr signál/šum ), která nám sděluje, že pro kvalitní příjem by měl být poměr mezi signálem a šumem větší než 22 dB.

## 2.3 Vliv prostředí na šíření signálu

WiFi technologie používá pro přenos dat rádiové vlny. Je tedy nutno počítat s tím, že se signál na cestě od vysílače k přijímači setkává s mnoha vlivy, které ho narušují a odebírají signálu na síle. Hned z kraje můžeme jmenovat pár příkladů z běžného života jako jsou zdi, budovy, stromy, člověk ale třeba i samotný vzduch, kterým se vlnění šíří. Pokud je signál moc slabý dochází k poklesu připojení, popřípadě k jeho destabilizaci nebo v případě úplné ztráty signálu, logicky i ke ztrátě připojení. Samozřejmě jinak se bude signál šířit v otevřeném prostoru, jinak v budově. Jinak se bude chovat v prostředí, kde se nachází spousta fyzických překážek a jinak ve volném prostředí.

První, a základní vlastnost, kterou musíme brát v potaz za všech okolností a podmínek, je vzdálenost mezi vysílačem a přijímačem. Platí, že čím větší je vzdálenost mezi nimi, tím slabší signál bude. Vztah signálu ke vzdálenosti je matematicky popsán jako převrácená hodnota kubické hodnoty vzdálenosti. V následujícím vzorci je tedy výsledná proměnná *strength* síla signálu, a proměnná *distance* je vzdálenost:

$$strength = \frac{1}{distance^3} \quad (2.5) [9]$$

Lze tedy konstatovat, že čím dále jsme od vysílače, tím slabší přijímáme signál, a případně musíme počítat i s poklesem rychlosti přenosu dat (závisí na více faktorech než jen na RSII). Další překážkou na cestě jsou samozřejmě fyzické předměty běžného života. Signál procházející hmotou ztrácí na intenzitě, a proto se můžeme setkat se situací, že i na malé vzdálenosti přijímač-vysílač může docházet k velkým ztrátám. Například ve firmě se spoustou kanceláří, kde signál musí na relativně krátké dráze od zdroje k cíli překonat mnoho překážek v podobě zdí, skříní, stolů, dveří... V tabulce 2.4 [10] jsou uvedeny příklady materiálů a k nim související úbytek intenzity signálu.

Tabulka 2.4: srovnání materiálů a jejich vliv na úbytek signálu[10]

Vliv na prostředí	Materiál	Ztráta (dB)
Nízký	Čiré sklo, dřevěné dveře, tvárnice zdi, omítka	2-4
Střední	Cihlové zdi, mramor, drátěný plot, tmavé sklo	5-8
Vysoký	Betonové zdi, papír, tvrzené sklo	10-15
Velice vysoký	Kov, zrcadlo	>15

Signál předměty nejen prochází a přichází tak o svou sílu, ale zároveň se od nich také odráží. Každá takto odražená vlna nejen ztrácí na intenzitě, ale dojde i k jejímu fázovému posuvu. Několikeré odrazení signálu způsobí relativní prodloužení jeho cesty k cíli. V momentě kdy při odrazu dojde k fázovému posuvu a všechny odražené (i přímé) signály konečně dorazí k přijímači, jejich fáze se vzájemně ovlivní – vyruší se - a dojde k redukci signálu. Na druhou stranu může dojít i k tomu, že se fáze setkají a po jejich sečtení dojde k navýšení síly signálu. V anglickém jazyce se tomuto problému odborně říká multi-path propagation [11]. Všechny tyto aspekty mají vliv na výslednou hodnotu RSSI přijatou přijímačem. Jedním z možných řešení, ať již k lepšímu či horšímu, může být přemístění prvků okolí (nábytek), změna polohy vysílače, výměna za výkonnější vysílač nebo změna polohy přijímače.



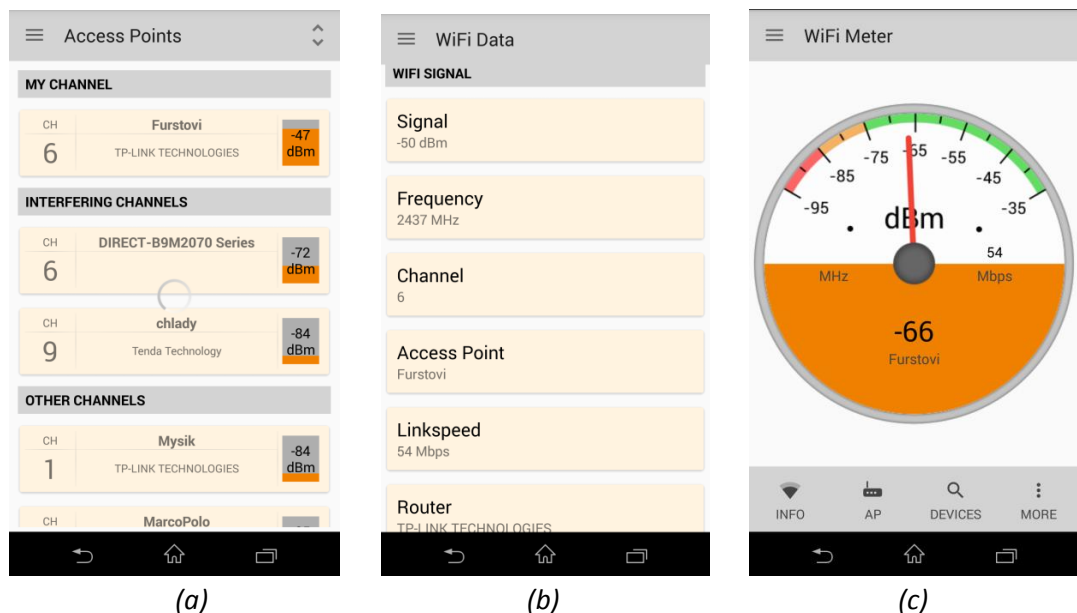
## 3. Dostupný software pro měření a vizualizaci signálu

### 3.1 Uživatelské aplikace

Na oficiálním online obchodě s aplikacemi pro Android (Google Play) [12] je k dispozici ke stažení nesčetné množství aplikací týkajících se WiFi společně s měřením signálu. Lze jmenovat příklady jako: WiFi Signal Strength, WiFi Signal, WiFi Checker, Network Signal Info, WiFi Scanner, WiFi Analyzer. Všechny tyto aplikace a mnoho dalších, jsou si v mnohém podobné, jejich název se liší jen nepatrně a nabízejí téměř totožnou funkcionalitu. Mění se pouze uživatelské rozhraní. Celou tuto množinu zmíněných aplikací však můžeme kategorizovat do dvou skupin.

#### 3.1.1 WiFi Signal Strength

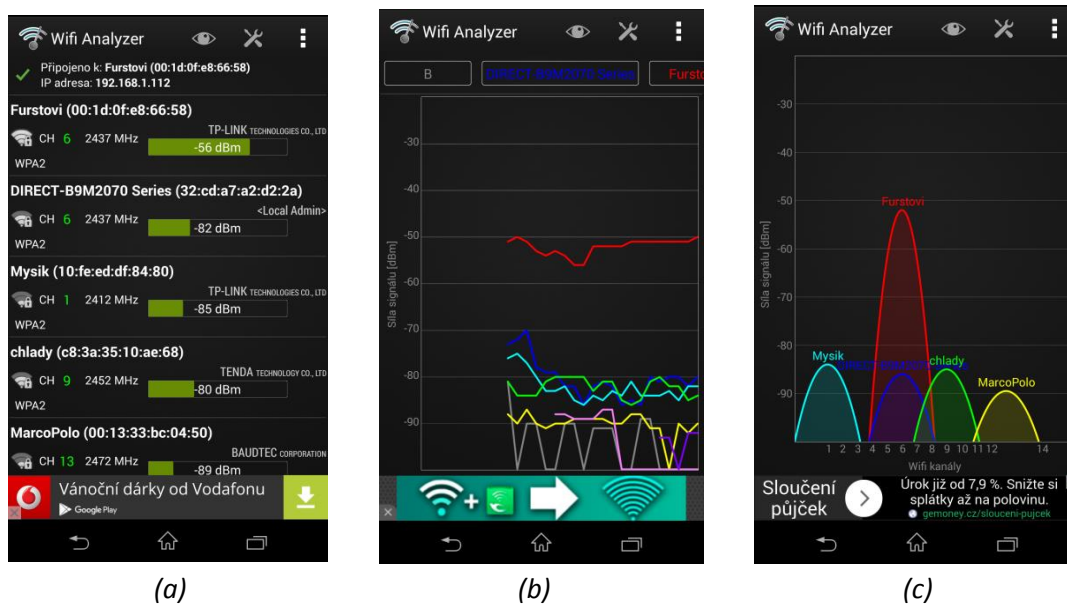
Tato kategorie aplikací obecně nabízí seznam dostupných vysílačů v okolí, základní informace k nim a samozřejmě zobrazení aktuálního stavu signálu. Může také zvlášť zobrazovat sílu k aktuálně připojenému Access Point (AP). Chybí však možnost zobrazit průběh signálu v předchozím čase – od začátku měření nebo možnost perzistence dat. Jako zástupce této skupiny můžeme jmenovat aplikaci *WiFi Signal Strength* [13] nebo *WiFi Checker*.



Obrázek 3.1: ukázka aplikace „WiFi Signal Strength“ [13]. Zobrazení dostupných AP (a), zobrazení aktuálního připojení v (b), grafické znázornění aktuální síly signálu

### 3.1.2 WiFi Analyzer

Aplikace z této skupiny nabízí rozšířenější sortiment možností, jak studovat průběh intenzity WiFi signálu. Kromě zobrazení dostupných zařízení, informace k nim a síly signálu, nabízí lepší grafické zobrazení naměřených hodnot v podobě grafů. Lze tak pozorovat průběh signálu v čase. Funkce pro filtrování AP, jejich řazení nebo nastavení intervalu měření nejsou výjimkou. Největší výhodou oproti aplikacím, které pouze měří momentální sílu signálu je, možnost z grafu vyčíst změny signálu za delší časový úsek. Časový úsek je však většinou omezen šířkou grafu a tak se po čase ztrácí dříve naměřené hodnoty. Velkou výhodou je tedy vizualizace kanálu každého vysílače, takže je možné sledovat kolize. Příklady jsou: *WiFi Analyzer* [14], *iMapper Wifi*, *Wi-Fi Scanner*



Obrázek 3.2: ukázka aplikace „WiFi Analyzer“ [14]. Zobrazení dostupných AP (a), graf zobrazení průběhu signálu (b), grafické znázornění rozsahu kanálů

## 3.2 Knihovny pro vizualizaci

V Android aplikacích, ve kterých je zapotřebí interpretovat data pomocí grafu, lze využít několik grafických knihoven třetích stran. Jednou z nich je například open-source knihovna GraphView [15]. Knihovna umožňuje zobrazování dat v reálném čase, zobrazení několika datových linek v jednom grafu, zoom osy Y. Z prvopočátku se její použití zdálo být dostačující, ale posléze se ukázalo, že nespĺňuje požadavky vhodné pro aplikaci vyvíjenou v této práci. Jedním z hlavních příčin byla nízká škálovatelnost při velkém počtu dat a měřených sítí. Knihovna měla vysoké výpočetní paměťové nároky a při náročnějších operacích docházelo k omezení plynulosti běhu aplikace. Důvodem může být provádění všech úkonů v rámci hlavního aplikačního vlákna.



Obrázek 3.3: ukázka knihovny GraphView [15]





## 3.3 Výběr platformy a jazyka pro BP

### 3.3.1 Přehled platforem

Dle zadání bakalářské práce bylo třeba vytvořit aplikaci pro mobilní zařízení, tedy pro telefon či tablet. Na řadě byl proto výběr platformy, pro kterou bude aplikace vyvíjena. Nabízelo se hned několik nejrozšířenějších mobilních operačních systémů: iOS, Android, Windows Phone, BlackBerry. Vývoj aplikací pro iOS [16]. má tu nevýhodu, že použití je možné pouze na přístrojích značky Apple, tedy na iPhoneu či iPadu. Další podmínka tkví v tom, že vývoj může probíhat pouze na operačním systému Mac OS X a za použití iOS SDK (Software Development Kit) [16]. Nejčastěji používaný programovací jazyk je potom Swift.

Další vysoce zvažovanou možností byl Windows Phone [16].. Vyvíjet můžeme pod Windows a většinou za pomoci jazyků Visual C++ nebo Visual C#. Jak Windows Mobile tak BlackBerry [16]. však mají slabé procentuální zastoupení na trhu, proto bylo i od této myšlenky upuštěno. Nejlepší variantou je proto platforma Android [16]. Tvorba aplikace s Android SDK může probíhat jak na Windows, iOS tak i na Linuxu. Nejčastější spojení s Androidem je programovací jazyk Java. Vývoj pro operační systém Android je zcela zdarma, nehledě na velké množství volně dostupných informací. Další velké plus lze připsat tomu, že Android je v dnešní době nejrozšířenější platformou pro mobilní zařízení. *Tabulka 3.1 [16]* zobrazuje mobilní platformy, a jejich základní vlastnosti.

Tabulka 3.1: přehled mobilních platforem [16]

Název	<b>iOS</b> 	<b>Android</b> 	<b>Windows Phone</b> 	<b>BlackBerry</b> 
<b>Společnost</b>	Apple Inc.	Open Handset Alliance	Microsoft	BlackBerry Ltd.
<b>Podíl na trhu</b>	18,3%	78%	2,7%	0,3%
<b>Jádro</b>	Darwin	Linux	Windows NT	QNX
<b>Jazyk</b>	Swift(C,C++)	Java (C,C++)	NET C# (C,C++)	C/C++
<b>SDK</b>	Mac OS X a iOS SDK	Linux, Mac OS X, Windows	Windows	Windows, Mac OS X, Linux

### 3.3.2 Zvolená cílová platforma

Jako cílová platforma pro tuto bakalářskou práci byl zvolen Android. Vývoj probíhá za pomoci Android SDK společně s IDE Eclipse (vývojové prostředí), fungujících na operačním systému Windows. Zařízení použité k testování je Sony Xperia Z s verzí Android 4.3.

## 4. Softwarová analýza

Aplikace, která by měla být výstupem této práce, klade důraz nejen na měření dat, ale především na jejich vhodnou interpretaci. Při vizualizaci je kladen důraz na jasnost, stručnost, a přesto komplexnost a korektnost naměřených dat. Měla zobrazovat průběh aktuálně měřených dat ale oproti ostatním aplikacím podobného charakteru, i starších, agregovaných, dat. Při měření je pak cílem nasbírat co nejvíce užitečných informací, které by se daly využít při zkoumání průběhu WiFi signálu.

Je tedy zapotřebí analyzovat a navrhnout způsob, jakým skloubit tyto dvě skupiny požadavků a společně s vhodnou interakcí uživatele tak vyvinout kvalitní software.

### 4.1 Požadavky

#### 4.1.1 Funkční

- **F1: seznam sítí** – aplikace by měla zobrazit všechny sítě a její přístupové body v dosahu. Ke každé z nich dodat základní info (SSID sítě, MAC adresa AP, zabezpečení, frekvence, síla signálu )
- **F2: seznam sítí** - pokud se některá ze sítí ztratí z dosahu, měla by v seznamu zůstat avšak být označena
- **F3: zobrazení dat** – aplikace by měla zobrazovat naměřená data v reálném čase ve formě lineárního grafu s tolerančními intervaly
- **F4: zobrazení dat** – aplikace by měla zobrazovat naměřená data v reálném čase ve formě heatmapy
- **F5: agregace dat** – aplikace by měla zobrazovat všechna naměřená data od počátku měření, dříve naměřená data je nutné agregovat z důvodu snížení paměťových nároků aplikace, i zachování přehlednosti zobrazovaných grafů.
- **F6: zobrazení legendy** – během měření může uživatel vypnout/zapnout vykreslování vybraných sítí
- **F7: export dat** – aplikace by měla být schopna exportovat naměřená data do souboru ve formátu Comma-Separated Values (CSV)

#### 4.1.2 Nefunkční

- **N1: platforma** – Android (doporučená verze 4.3 a vyšší)
- **N2: škálovatelnost** – vykreslování a vizualizace dat by měla být adekvátně náročná tak, aby rychle reagovala i při velkém počtu měřených sítí a to i na průměrně výkonném přístroji
- **N3: paměť** - snažit se minimalizovat zbytečnou zátěž operační paměti telefonu tak, aby bylo možné naměřit co největší objem dat

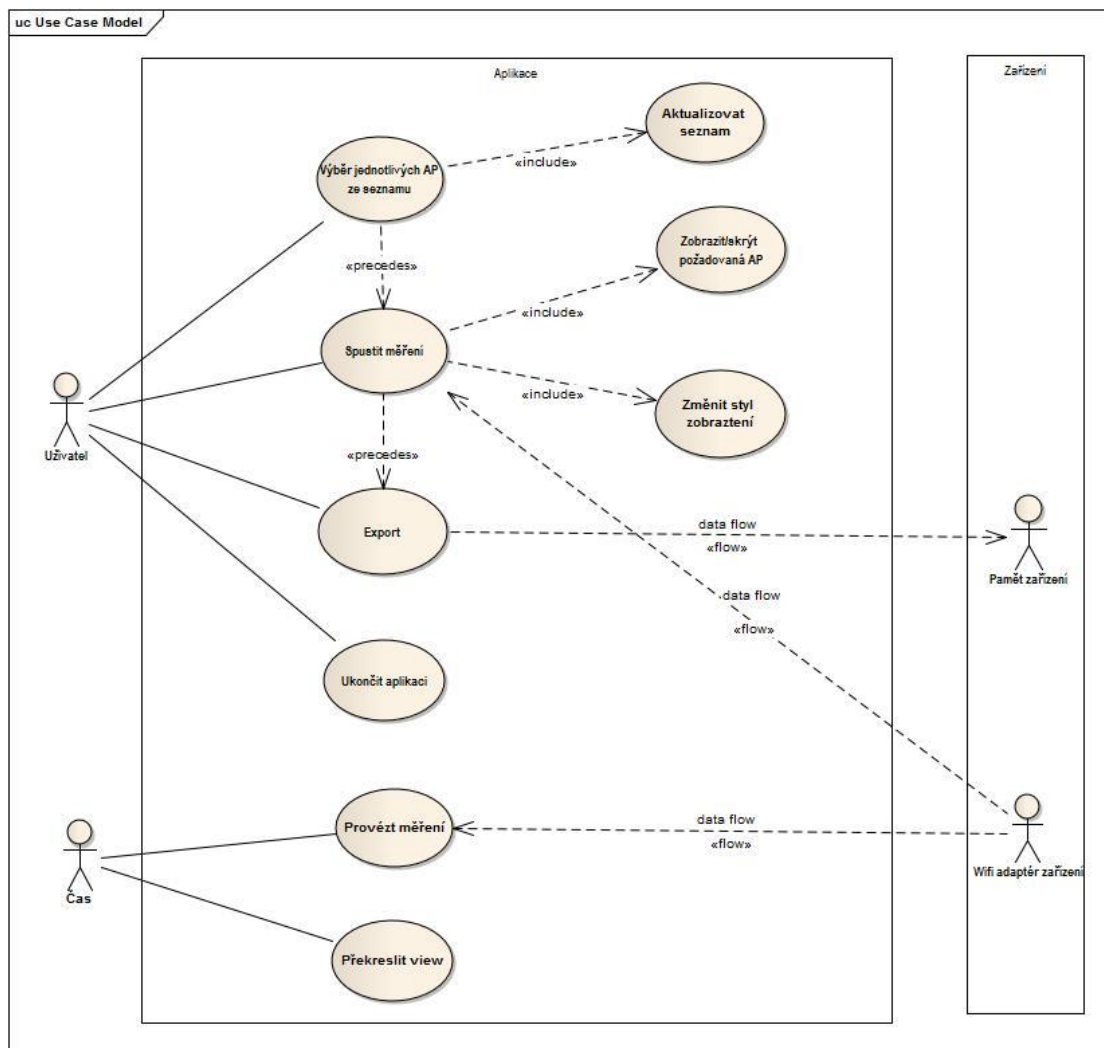
- **N4: běh na pozadí**– pokud je aplikace překryta jinou aplikací, měla by stále běžet avšak za minimálních podmínek. Budou tedy pozastaveny všechny funkce s výjimkou samotného načítání dat.
- **N5: licence** – všechny použité knihovny musí umožňovat bezplatné použití pro komerční aplikace

## 4.2 Scénáře užití

Na následující případech užití se účastní tyto aktéři:

- **Uživatel** - běžný uživatel aplikace, ovládá aplikaci pomocí gest na displeji
- **Čas** - čas je vnější faktor, periodicky spouští měření a vykreslování dat
- **Paměťové uložení** - nevstupuje do běhu aplikace přímo, využívá se pouze pro uložení dat
- **WiFi adaptér** - hardware zařízení, díky kterému získáváme nová data a informace skenováním okolí, sám o sobě do aplikace nezasahuje, o nové hodnoty si musíme říci voláním funkce

Obrázek 4.1 zobrazuje grafické schéma případů užití. Ze schématu lze vyčíst, že uživatel jako účastník případu užití může zahájit výběr položek ze seznamu. Může také samotný seznam aktualizovat. Dále může spustit měření, které následuje výběru AP (Access Point). Schéma také znázorňuje, že během měření může uživatel měnit styl zobrazení dat (lineární graf, heatmapa, gesta prstem, vypnutí tolerančních intervalů) a může taktéž vypínat zobrazení určitých AP za chodu. Na měření pak navazuje případ užití, ve kterém probíhá export dat do souboru. Nechybí ani znázornění případu, kdy uživatel chce běh aplikace ukončit. Diagram případů užití zachycuje další tři aktéry. Čas, který cyklicky spouští měření dat a iniciuje překreslení view, WiFi adaptér, ze kterého získáváme nová data, a paměťové zařízení, do kterého probíhá export dat.



Obrázek 4.1 : schéma případů užití

Přehled jednotlivých scénářů užití je popsán v následujících tabulkách:

Tabulka 3.1 : scénář případu užití „Vybrat jednotlivá AP“

Případ užití	Vybrat jednotlivá AP
Aktéři	Uživatel
Tok událostí	1. aplikace zobrazí seznam přístupových bodů v dosahu 2. uživatele kliknutím vybere požadované AP. Tlačítkem může vybrat všechny, nebo naopak žádné AP. 3. uživatel potvrdí svůj výběr 4. proběhne validace výběru
Vstupní podmínky	Spuštěná aplikace
Výstupní podmínky	Po potvrzení a validaci výběru se spustí nová aktivita

Tabulka 3.2 : scénář případu užití „Aktualizovat seznam“

Případ užití	Aktualizovat seznam
Aktéři	Uživatel
Tok událostí	1. uživatel klikne na tlačítko aktualizace seznamu 2. proběhne nový WiFi sken 3. zobrazí se nový, aktualizovaný seznam s AP
Vstupní podmínky	Tento scénář je součástí scénáře ‚Vybrat jednotlivé AP‘
Výstupní podmínky	Aktualizovaný seznam

Tabulka 3.3 : scénář případu užití č.3

Případ užití	Spustit měření
Aktéři	Uživatel
Tok událostí	1. Uživatel potvrzením výběru AP spustil novou aktivitu 2. Proběhne inicializace proměnných a vláken 3. Značně měření dat
Vstupní podmínky	Validní výběr a potvrzení ve scénáři užití ‚vybrat jednotlivé AP‘
Výstupní podmínky	Spuštění měření dat pro zvolené AP

Tabulka 3.4 : scénář případu užití „Zobrazit/skrýt požadovaná AP“

Případ užití	Zobrazit/skrýt požadovaná AP
Aktéři	Uživatel
Tok událostí	1. Uživatel klikne na položku v legendě grafu 2a. Při odznačení se vybrané AP přestane zobrazovat mezi naměřenými daty 2b. Při označení se vybrané AP opět začne zobrazovat mezi naměřenými daty
Vstupní podmínky	Tento scénář je součástí scénáře ‚Spustit měření‘
Výstupní podmínky	Změna interpretace naměřených dat



Tabulka 3.5 : scénář případu užití „Změnit styl zobrazení #1“

Případ užití	Změnit styl zobrazení #1
Akteři	Uživatel
Tok událostí	<ol style="list-style-type: none"> <li>1. Uživatel spustí měření</li> <li>2a. Uživatel kliknutím na tlačítko panelu může přepnout na zobrazení z grafu na heatmapu</li> <li>2b. Uživatel kliknutím na tlačítko panelu může přepnout na zobrazení z heatmapy na graf</li> <li>3c. Uživatel kliknutím na tlačítko panelu může přepnout na zobrazení Legendy</li> <li>3. změní se view de výběru</li> </ol>
Vstupní podmínky	Tento scénář je součástí scénáře „Spustit měření“
Výstupní podmínky	Změna interpretace naměřených dat

Tabulka 3.6 : scénář případu „Změnit styl zobrazení #2“

Případ užití	Změnit styl zobrazení #2
Akteři	Uživatel
Tok událostí	<ol style="list-style-type: none"> <li>1. Uživatel spustí měření</li> <li>2. Zobrazí se view s lineárním grafem</li> <li>3. Uživatel klikne na tlačítko „menu“ a vybere položku „zobrazit intervaly“</li> <li>4. V bodové části grafu se u scatterů zobrazí intervaly minima a maxima</li> </ol>
Vstupní podmínky	Tento scénář je součástí scénáře „Spustit měření“
Výstupní podmínky	V bodové části grafu se u scatterů zobrazí/vypnou intervaly minima a maxima

Tabulka 3.7 : scénář případu užití „Změnit styl zobrazení #3“

Případ užití	Změnit styl zobrazení #3
Aktéři	Uživatel
Tok událostí	<ol style="list-style-type: none"> <li>1. Uživatel spustí měření</li> <li>2. Zobrazí se view s lineárním grafem</li> <li>3a. Uživatel tahem jednoho prstu scrolluje křivky grafu v čase</li> <li>3b. Uživatel mění gestem dvou prstů pozici hraniční čáry</li> <li>4. Změní se zobrazený časový úsek a agregace dat</li> </ol>
Vstupní podmínky	Tento scénář je součástí scénáře „Spustit měření“
Výstupní podmínky	Změna pozice hraniční čáry, změna zobrazeného časového úseku a agregace starších dat

Tabulka 3.8 : scénář případu užití „Export do csv“

Případ užití	Export do csv
Aktéři	Uživatel, paměťová karta zařízení
Tok událostí	<ol style="list-style-type: none"> <li>1. Uživatel v liště menu vybere možnost export do CSV</li> <li>2. Ukončí se měření</li> <li>3. Spustí se nová aktivita</li> <li>3. Uživatele zadá název nového souboru</li> <li>4. Proběhne validace vstupu a kontrola přístupu úložného zařízení</li> <li>5. Proběhne zápis dat do nového souboru</li> </ol>
Vstupní podmínky	Přístup k paměti zařízení
Výstupní podmínky	Zápis naměřených dat do nově vzniklého souboru

Tabulka 3.7 : scénář případu užití „Ukončit aplikaci“

Případ užití	Ukončit aplikaci
Aktéři	Uživatel
Tok událostí	1. Uživatel může v kterémkoliv momentě kliknout na menu v liště a kliknutím ukončit aplikaci
Vstupní podmínky	Spuštěná aplikace
Výstupní podmínky	Ukončená aplikace

Tabulka 3.8 : scénář případu užití „Proved' měření“

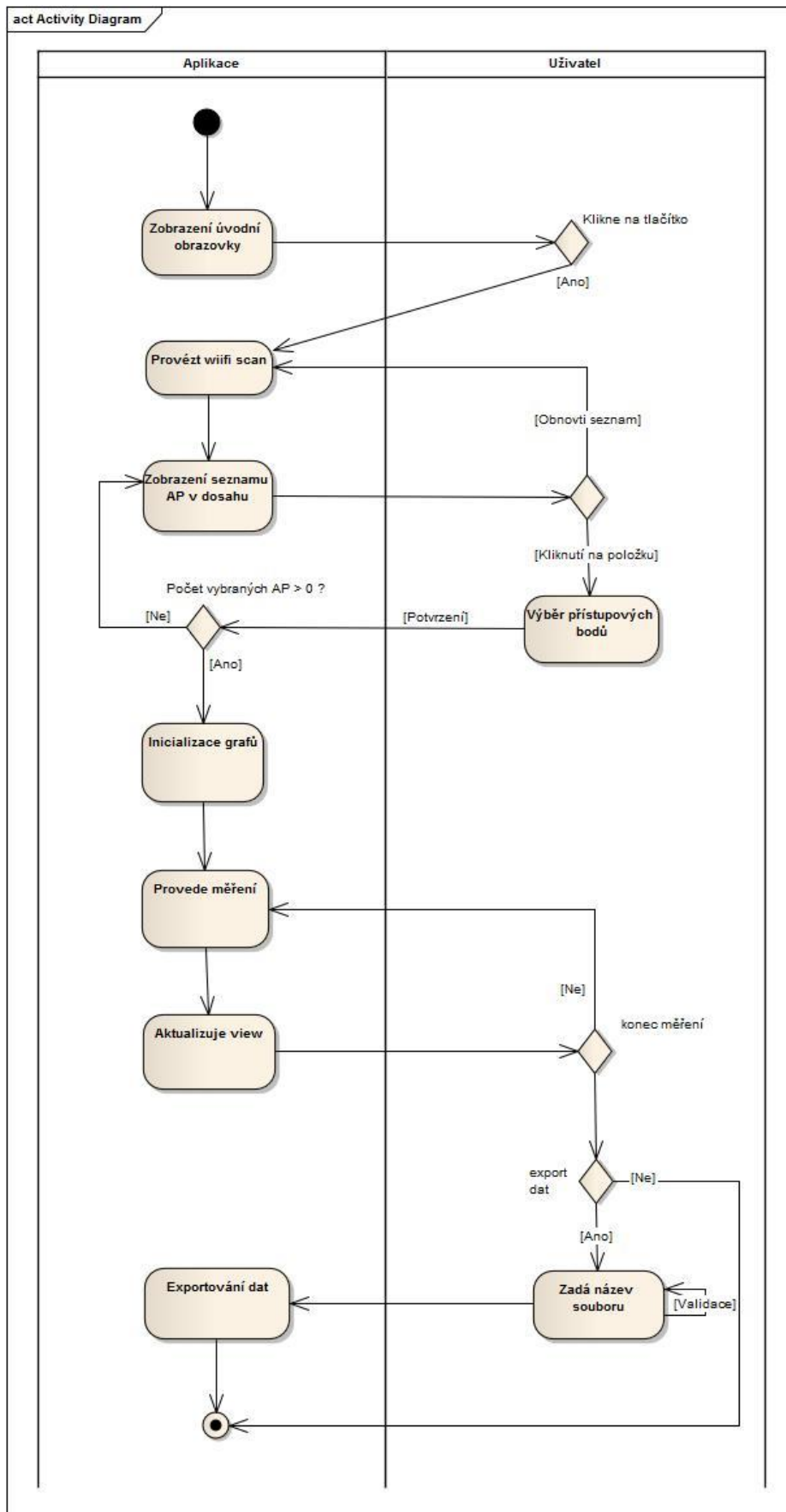
Případ užití	Proved' měření
Aktéři	Čas
Tok událostí	1.V předem stanoveném časovém intervalu aplikace provádí sken WiFi sítí v dosahu 2. Uloží data do datové struktury
Vstupní podmínky	Spuštěná aktivita měření signálu
Výstupní podmínky	Nově naměřená data, přístup k WiFi adaptéru

Tabulka 3.9 : scénář případu užití „Překreslit view“

Případ užití	Překreslit view
Aktéři	Čas
Tok událostí	1.V předem určeném časovém intervaly probíhá překreslování view grafu a mapy
Vstupní podmínky	Spuštěná aktivita měření signálu
Výstupní podmínky	Překreslené view

### 4.3 Diagram aktivit

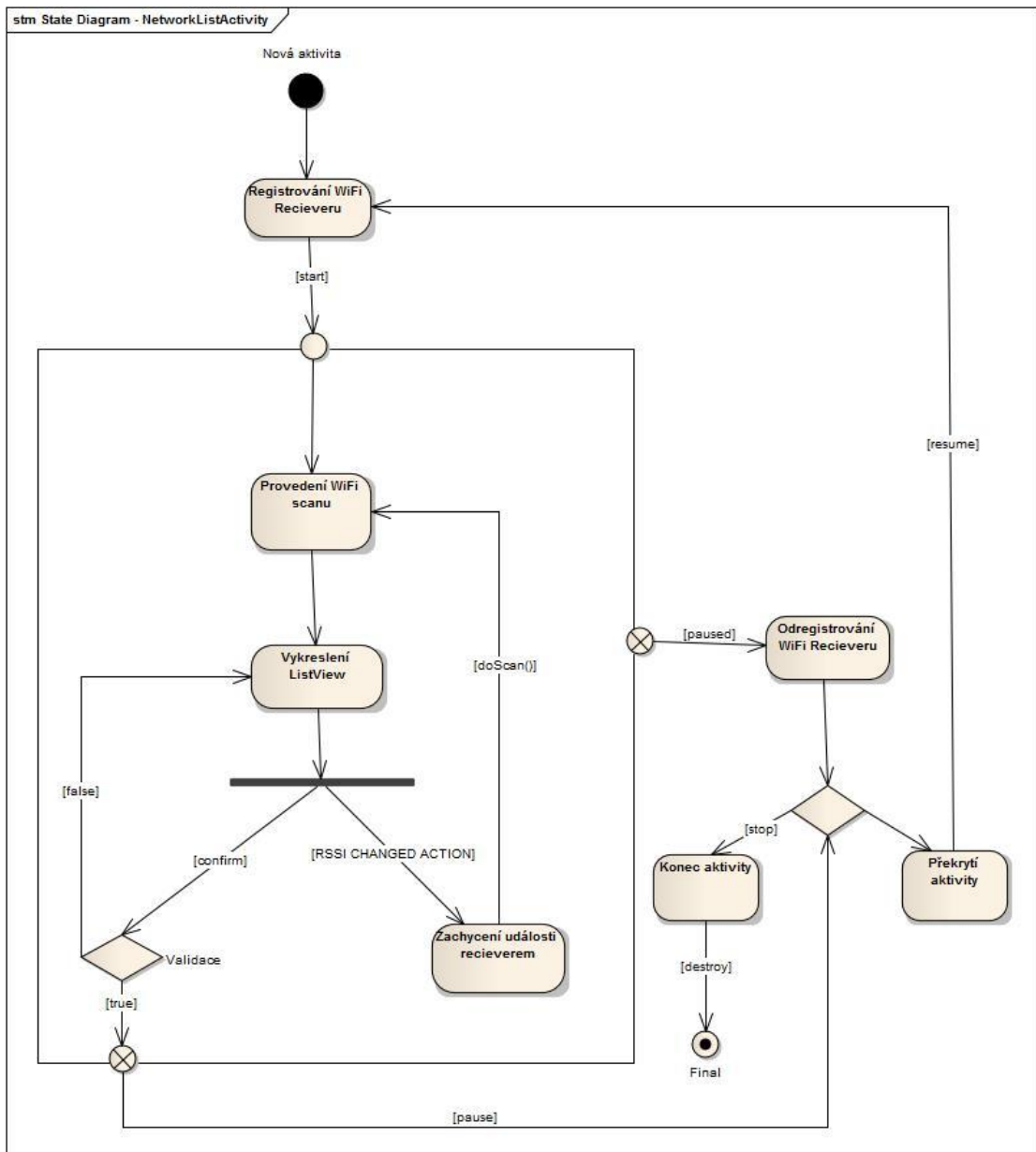
Obrázek 3.2 znázorňuje schéma aktivit aplikace. Diagram začíná spuštěním aplikace. Poté následuje zobrazení úvodní obrazovky aplikace. Uživatel kliknutím na tlačítko postoupí na seznam dostupných WiFi sítí v dosahu a následně vybere požadované položky. Projde-li výběr validací, aplikace spustí měření signálu vybraných AP, která probíhá do doby, než ho uživatel ukončí kliknutím na tlačítko. Před ukončením celého procesu může uživatel vybrat možnost exportu naměřených dat.



Obrázek 4.2 : diagram aktivit

## 4.4 Stavový diagram

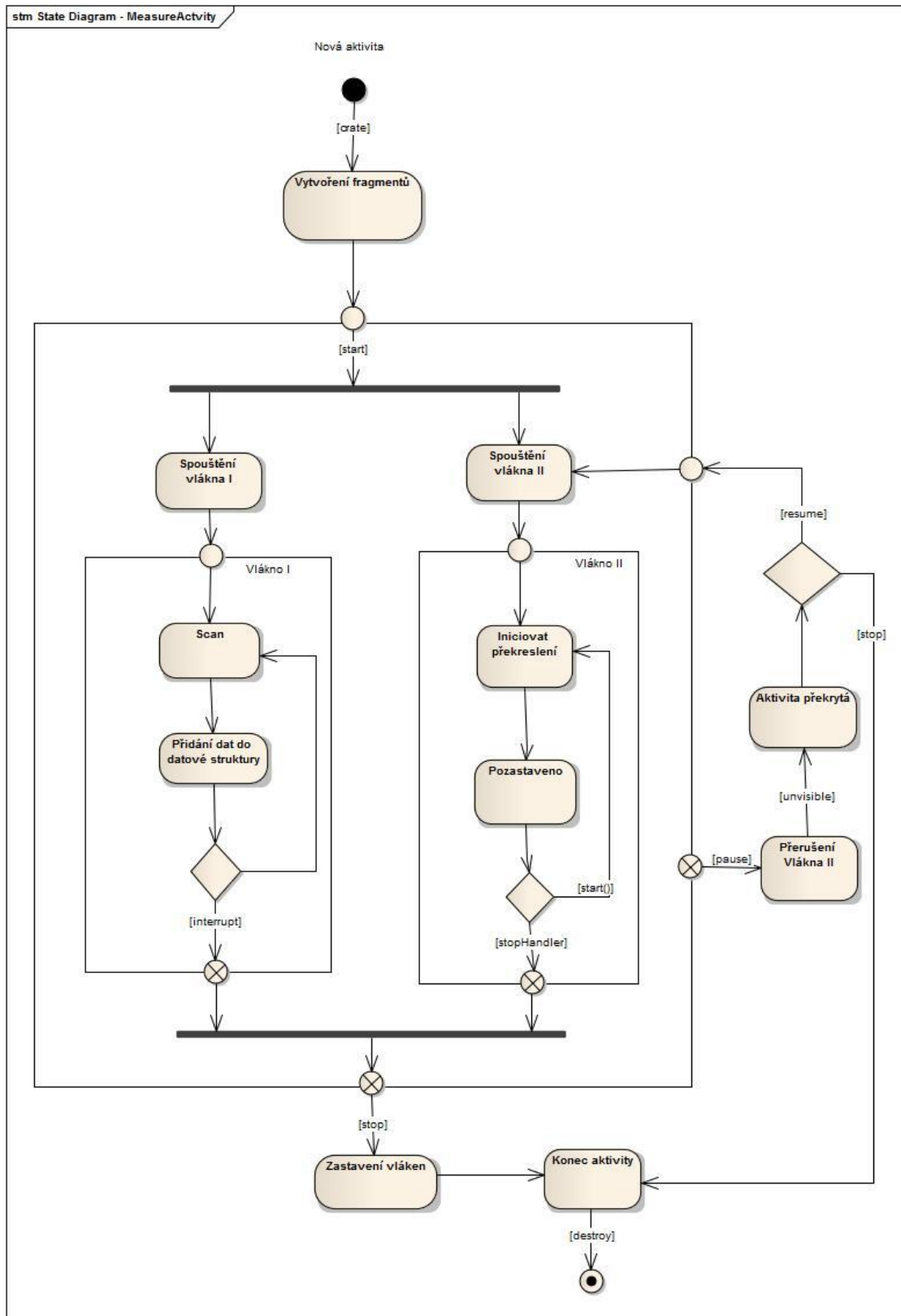
### 4.4.1 NetworkListActivity



Obrázek 4.3 : stavový diagram pro aktivitu NetworkListActivity

Obrázek 4.3 popisuje přechod mezi stavy v aktivitě `NetworkListActivity`, zobrazující seznam dostupných sítí. Po startu aktivity se registruje WiFi reciever, oskenuje se okolí a zobrazí se `ListView`. V případě, že reciever detekuje zprávu o změně stavu RSSI, provede se nový sken a upraví se `ListView`. Po potvrzení výběru uživatelem aktivita končí. Je-li aktivita na pozadí, proběhne odregistrování recieveru.

#### 4.4.2 MeasureActivity



Obrázek 4.4 : stavový diagram aktivity MeasureActivity

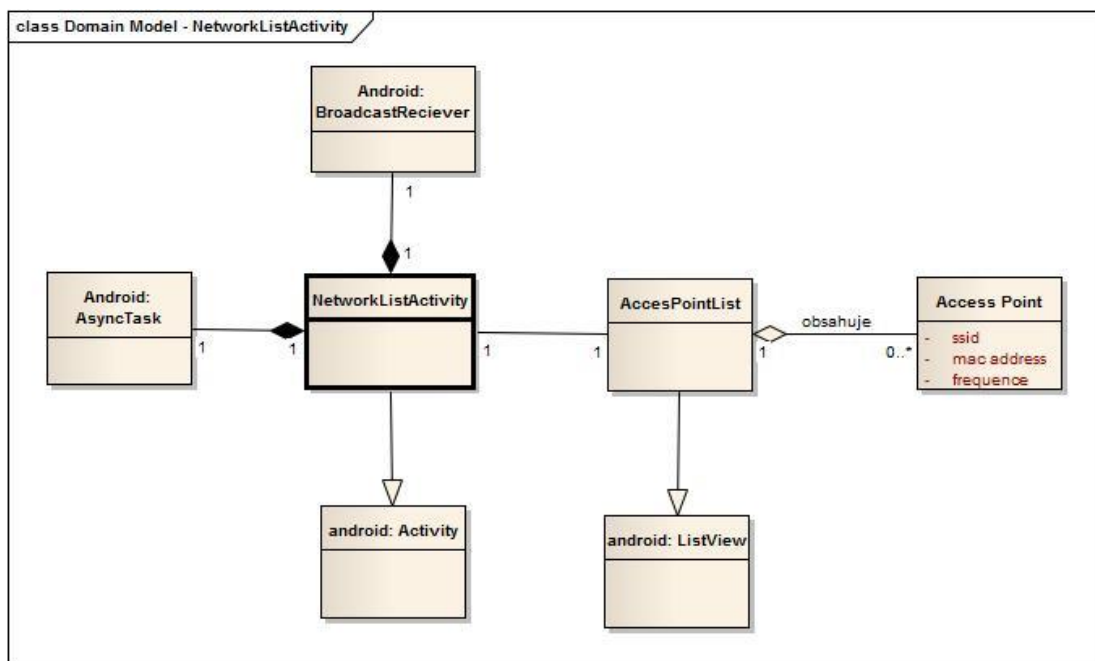
Souvislost stavů znázorněných na *obrázku 4.4* je následující: po spuštění aktivity se inicializují komponenty. Poté se spustí dvě na sobě nezávislá vlákna – jedno na načítání dat a druhé na překreslování view. První vlákno běží nepřetržitě až do zrušení aktivity. Druhé vlákno provede svou činnost, poté je na chvíli pozastaveno a opět provede svou činnost. Zatímco první vlákno se stará pouze o načtení nových dat a jejich uložení do datové struktury, druhé vlákno řídí překreslování daných view. Vykreslovací vlákno však může vykreslení jen iniciovat, protože platforma Android provádí všechny vykreslovací úkony v hlavním aplikačním vlákne s normální prioritou.

Zatímco načítací vlákno běží od začátku do skončení aktivity, vykreslovací vlákno se pozastaví vždy, když je aktivita překryta či běží na pozadí.

## 4.5 Doménový model

### 4.5.1 NetworkListActivity

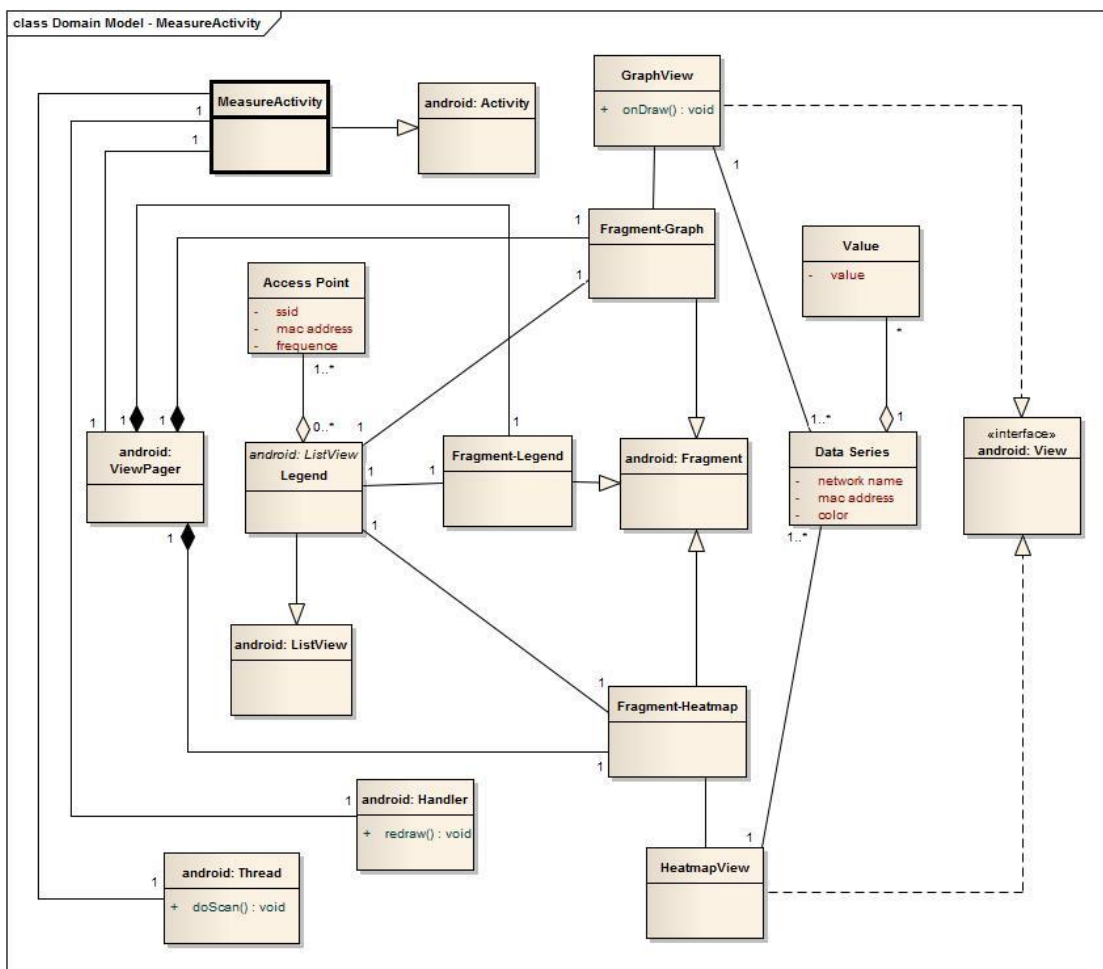
Doménový model NetworkListAktivity zachycuje vztahy mezi objekty, které se podílejí na zobrazení seznamu s dostupnými bezdrátovými sítěmi. Na *obrázku 4.5* je tučně zvýrazněna třída NetworkListActivity, kterého se model týká. Ze schématu je patrné, že třída NetworkListActivity obsahuje Android třídy BroadcastReceiever pro obdržení informací o změně stavu signálu, AsyncTask a děti od třídy Activity. Dále pak třída obsahuje seznam přístupových bodů, který je potomkem Android třídy ListView.



Obrázek 4.5 : doménový diagram pro třídu NetworkListActivity

### 4.5.2 MeasureActivity

Doménový popis aktivity měření a vizualizace dat. Objekt, který reprezentuje Android aktivitu je zvládněná třída MeasureActivity. Z grafu je patrné, že třída MeasureActivity obsahuje referenci na objekt třídy ViewPager z Android API, který nabízí správu fragmentů pomocí TabLayoutu. Každý fragment pak obsahuje vlastní implementaci View a ListView jako legendy. Každá implementace View obsahuje datové série s naměřenými hodnotami a legendu tvoří seznam přístupových bodů. Důležitým prvkem je také znázornění tříd Handleru a Thread, které jsou taktéž součástí třídy MeasureActivity.



Obrázek 3.6 : doménový diagram pro třídu MeasureActivity



## 5. Implementace

Tato kapitola popisuje způsob implementace důležitých částí aplikace, použité knihovny, a závěrem nabízí několik ukázek finální podoby projektu. Navazuje na softwarovou analýzu, podle které by měla implementace probíhat.

Zdrojový kód byl průběžně nahráván na server gitlab. Je možné zde nalézt kompletní verzi aplikace [17]

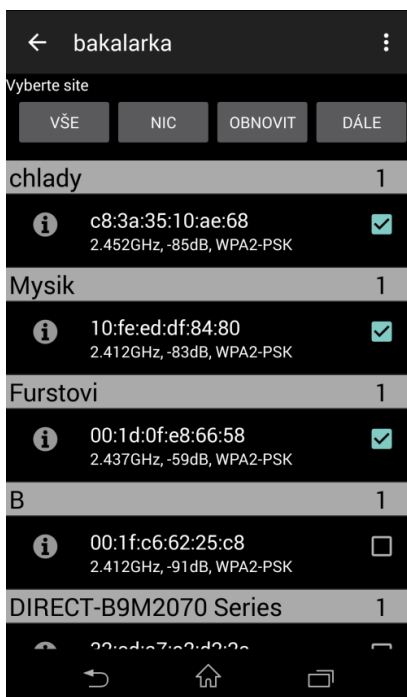
### 5.1 Průběh implementace

#### 5.1.1 NetworkList

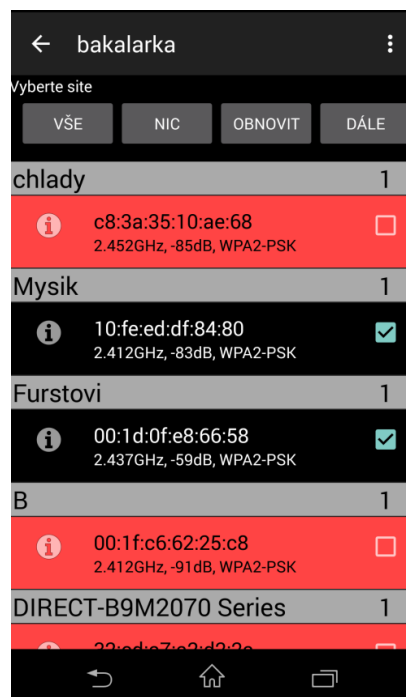
Jako první bylo třeba definovat třídu, respektive aktivitu, která bude provádět skenování WiFi sítí v okolí, a výsledky vizualizovat v listView. Každá položka seznamu reprezentuje instanci třídy. Rozlišujeme tři typy tříd obsažených v ListView, které implementují stejné rozhraní:

- Prvním typem je název sítě, neobsahuje nic než název sítě.
- Druhý typ reprezentuje přístupový bod. Obsahuje informace o přístupovém bodě, jako například MAC adresu, zabezpečení, frekvenci. Součástí tohoto itemu je také zaškrtačovací tlačítko reprezentující vybrání dané položky. Informace o tom, která AP byla označena a která ne, udržujeme v separátním poli (obrázek 5.1).
- Třetím typem je přístupový bod, který se ztratil z dosahu, je označen červeně a nelze jej vybrat.

Dále je registrován WiFi Reciever pro událost `RSSI_STATE_CHANGED`. Při změně se provede nový WiFi sken a nově získaná AP se porovnají s dosavadními, obsaženými v seznamu. Pokud některý z přístupových bodů chybí, jeho instance v ListView je nahrazena instancí třídy „ztracený přístupový bod“ (obrázek 5.2). V opačném případě dojde k obrácenému procesu. Kliknutím uživatele na speciální tlačítko se provede nový sken a ke kompletnímu překreslení view.



Obrázek 5.1 : Listview zobrazuje AP



Obrázek 5.2 : označení ztracených AP

### 5.1.2 GraphView

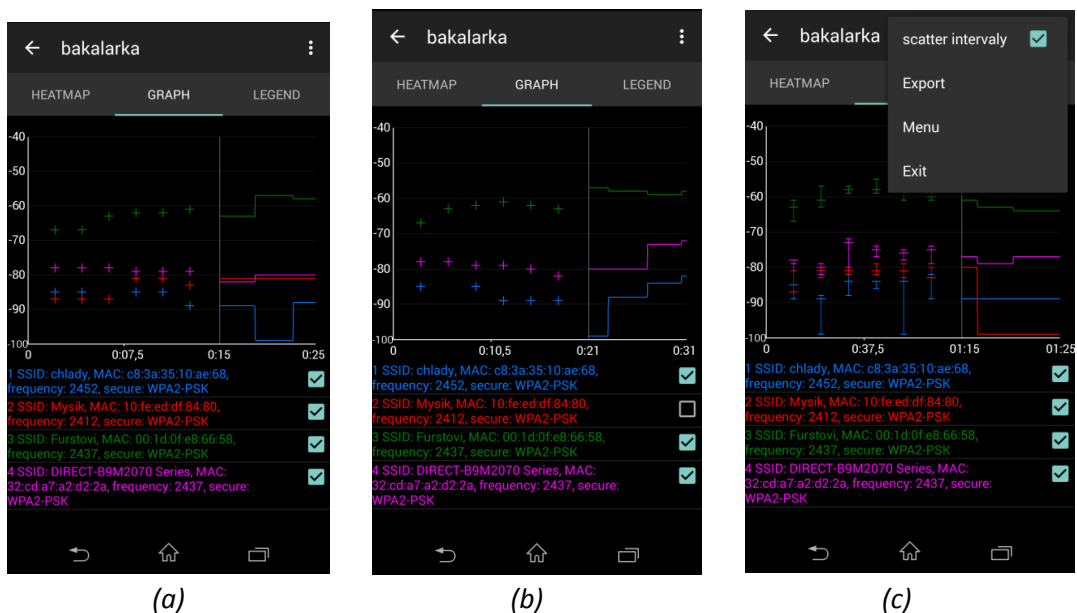
Implementace vlastní funkcionality zobrazení měřené síly signálu nabízí vizualizaci naměřených dat pomocí lineárního grafu. Přesněji řečeno skládá se z lineární části a z agregované části. Agregovaná část dat je pak reprezentována bodovým grafem s tolerančními intervaly v podobě percentilů. Hranice mezi aktuální (lineární) částí a agregovanou (bodovou) částí je určena hraniční čarou, která pomocí gest dvou prstů může měnit svou pozici (popsáno dále v kapitole 5.1.4). Gesto jedním prstem pak scrolluje v lineární části (viz. podkapitola 5.1.4). Třída udržuje instanci třídy HashMap, ve které je MAC adresa daného AP, a hodnotu reprezentuje seznam naměřených hodnot. Každý seznam hodnot reprezentuje křivku v grafu a pro snadné rozeznání má určenou svou barvu. Skenování a ukládání nových hodnot do HashMapy probíhá v samostatném vlákně, které běží, i v případě pozastavení aktivity.

Nejdříve se podívejme na část grafu, která pomocí linek vykresluje aktuální průběh signálu.

Je dána vzdálenost mezi jednotlivými body v grafu a interval mezi jednotlivými měřeními. Lze tedy určit, počet bodů v křivce a délku znázorňovaného časového úseku. Vykreslování probíhá od konce, tedy od naposledy naměřené hodnoty. V grafu je tedy nejaktuálnější hodnota vpravo a směrem doleva se vizualizují starší data. K vykreslení křivky se používá funkce Canvas.drawLine(). Osa X tedy reprezentuje čas a osa Y hodnotu signálu v decibelech. Pro lepší přehlednost začíná rozsah osy Y od -40 dBm.

Nalevo od hraniční čáry začíná vizualizace agregovaného zbytku dat, která se nevešla do pravého okna s aktuálním průběhem. Vzdálenost mezi scattery je dána a tudíž se dá výpočtem určit počet scatterů. Data, která mají být agregována, jsou rozdělena na stejný počet dílů, který je roven počtu scatterů. Pro každou tuto podmnožinu je vypočítána střední hodnota, toleranční interval definovaný percentily 0,05 a 0,95. Výsledné hodnoty jsou znázorněny jako scattery.

Překreslování grafu běží v samostatném vlákně, které se spouští každých 50 ms, tedy 20krát za sekundu. Tím je zaručena plynulost pohybu lineární části grafu. Část obsahující agregovaná data se překresluje každou sekundu a při každém překreslení dojde k přerozdělení dat a k přepočtu hodnotu scatterů. Je-li aplikace upozaděna, vykreslovací vlákno se zastaví.



Obrázek 5.3 : ukázka GraphView. Scattery bez intervalů (a), vypnutí vykreslování AP (b), zapnutí intervalů (c)

### 5.1.3 HeatmapView

HeatmapView reprezentuje data pomocí barev. Každé měřené AP je interpretováno pruhem, který se skládá z jednotlivých barevných rámců. Pruhy jsou skládány pod sebe a tvoří tak heatmapu (Obrázek 5.5 a 5.6). Barevný rámec má danou minimální šířku, takže na základě šířky grafu je možné stanovit maximální počet rámců v pruhu. Výška rámce a se mění na základě parametru počtu měřených AP a výšky grafu (Obrázek 5.7).

Je-li počet zobrazovaných dat menší nežli maximální počet rámců v grafu, každý rámeček reprezentuje jednu naměřenou hodnotu. Když však měřená data překročí počet maximálního počtu rámců v grafu, je nutné data průměrovat, tj. jeden rámeček bude zastupovat více naměřených hodnot, přesněji řečeno jejich průměr (obrázek 5.4). Průměrování probíhá odzadu, tzn. nejdříve průměrujeme starší data. Například bude-li maximální počet rámců 500 a my budeme mít naměřených 502 hodnot, všechny rámečky budou reprezentovat po jedné hodnotě, pouze první dva rámečky (nejstarší hodnoty) budou reprezentovat průměr ze dvou hodnot.

Načítání probíhá stejně jako u `MapView` v samostatném vlákně, periodicky po 50 ms. Taktéž překreslování mapy probíhá zvlášť a dojde k němu vždy, když je modul o počtu dat a maximálního počtu rámců roven 0 (v tomto případě bude každý rámeček reprezentovat stejný počet dat). Výjimkou je případ, kdy uživatel ručně přepne zobrazení heatmapy z jiného view. V takovém případě se heatmapa překreslí ihned a nastane výše zmíněný případ nerovnoměrného průměrování.

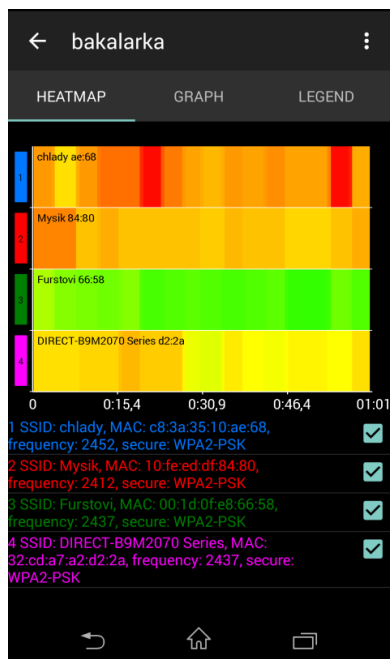
Obrázek 5.4 obsahuje ukázkou algoritmu pro průměrování hodnot v rámci.

```

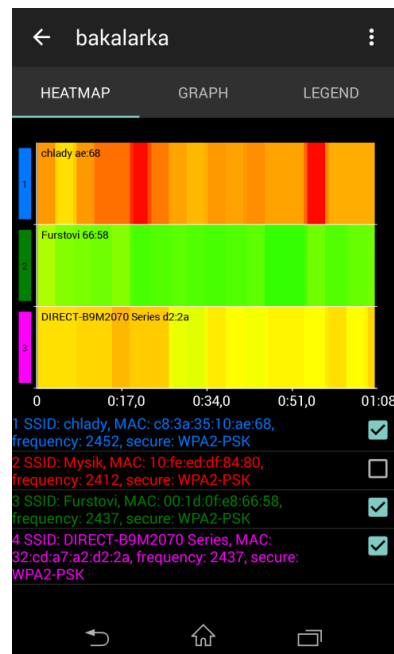
94 private int[] mergePoints(int maxFrameCount) {
95     int[] values = new int[maxFrameCount];
96     //how many data will merged to one:
97     int splitSize = (int) Math.floor((float) nmbOfPoints / (float) maxFrameCount);
98     //if splitSize is not whole number -> modulo!=0
99     int modulo = nmbOfPoints % maxFrameCount;
100    int index = 0;
101    for (int start = 0; start < nmbOfPoints; start += splitSize) {
102        int count = 0;
103        for (int i = 0; i < splitSize; i++) {
104            if ((start + i) >= nmbOfPoints) {
105                splitSize = i;
106                break;
107            }
108            int value = series.get(start + i).getValue();
109            value = Math.abs(value);
110            count += value;
111        }
112        //till the modulo isn't zero, we are counting one more value to each sub-array
113        if(modulo!=0){
114            int value = series.get(start + splitSize).getValue();
115            value = Math.abs(value);
116            count += value;
117            start++;
118            modulo--;
119            count = count / (splitSize+1);
120        }else
121            count = count / splitSize;
122        values[index] = count;
123        index++;
124    }
125    return values;
126

```

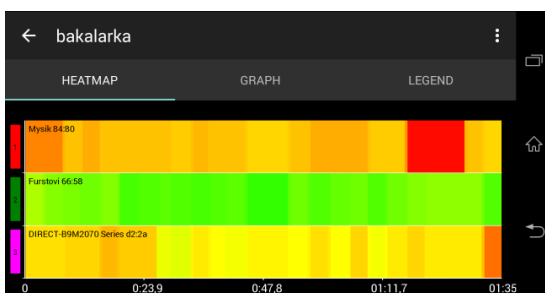
Obrázek 5.4 : zachycuje algoritmus průměrování. Klíčové veličiny jsou `maxFrameCount` – maximální počet rámců, `values` – pole výsledných hodnot, `splitSize` – počet dat v jednom rámečku, `nmbOfPoint` – počet naměřených hodnot



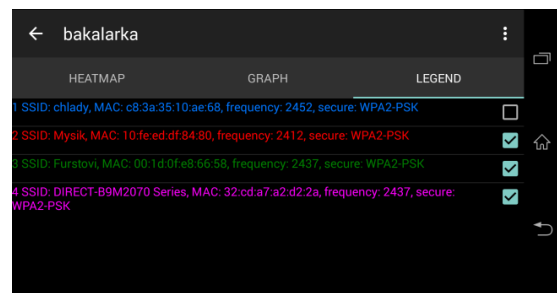
Obrázek 5.5: HeatmapView



Obrázek 5.6: HeatmapView – vypnuté AP



Obrázek 5.7: HeatMapView na šířku



Obrázek 5.8: legenda na šířku

#### 5.1.4 TouchEventHandler

Třída `TouchEventHandler` je součástí funkčnosti pro vizualizaci naměřených dat (kapitola 5.1.2) a je založena na implementaci metody `onTouchEvent`. Jejím úkolem je obstarat gesto jednoho a dvou prstů. Při pohybu po obrazovce jedním prstem v místě grafu dochází ke scrollování lineární části grafu. Lze tak tahem posouvat křivky a měnit poměr mezi aktuálními a agregovanými daty, která byla agregována do scatterů. Princip implementace je prostý. V grafu jsou udržovány indexy prvního a posledního zobrazeného bodu v grafu. Za normální situace je poslední zobrazený bod poslední naměřená hodnota, jeho index je roven `array.length`, a první zobrazená hodnota

v grafu je  $array.length - počet\_bodů\_tvořící\_křivku$ . V případě že tah proveden jedním prstem, algoritmus na základě délky pohybu určí, o kolik bodů se linka posune a do jaké míry se změní velikost agregované části grafu. Jako příklad si můžeme říci, že posuv proběhne o 20 bodů (jelikož mezi body je časový interval 50ms, 20 bodů znázorňuje 1 sekundu), graf nyní bude pomocí křivky zobrazovat hodnoty od indexu  $array.length - počet\_bodů\_tvořící\_křivku - 20$  do indexu  $array.length - 20$ . S posuvem samozřejmě dochází i k přepočítání scatterů a k úpravě hodnot na ose X (Obrázek 5.9)

Pohyb dvou prstů v podstatě rozšiřuje předchozí případ. Algoritmus na základě změny vzdálenost mezi prsty rozhodne o kolik a jakým směrem se provede posunutí hraniční čáry (Obrázek 5.10). Poté už se jen upraví proměnná zastupující index prvního zobrazeného bodu ve křivce grafu. Opět si lze pro příklad uvést případ, kdy posuv hraniční linky proběhne o hodnotu  $X$ . Algoritmus spočítá na základě vzdálenosti mezi jednotlivými body v grafu, kolik bodů se vejde do hodnoty  $X$ , a upraví index prvního zobrazeného grafu v křivce touto nově získanou hodnotu  $Y$ . Index první hodnoty pak bude  $array.length - počet\_bodů\_tvořící\_křivku - Y$ . Hodnota  $Y$  se na základě směru pohybu buďto přičítá či odečítá. Zvětší se nám tak okno s křivkou grafu a zmenší se okno zobrazující agregované hodnoty.

Obrázek 5.9 nabízí ukázkou algoritmu pro pohyb jednoho prstu. Na řádce 80 proběhne výpočet euklidovské vzdálenosti a na řádcích 82 a 85 se na základě této hodnoty upravují parametry grafu.

```

74     case MotionEvent.ACTION_MOVE:{
75         int touchedPoints = ev.getPointerCount();
76         if (touchedPoints == 1) {
77
78             float currX = ev.getX(0);
79             float currY = ev.getY(0);
80             int size = MyMath.euclidSize(currX, currY, mPrevFirstFingerX, mPrevFirstFingerY);
81             if(currX < mPrevFirstFingerX){
82                 graphParams.setScrollLineSeriesIndex(size/REDUCTION_VALUE);
83
84             }else if(currX > mPrevFirstFingerX){
85                 graphParams.setScrollLineSeriesIndex(-size/REDUCTION_VALUE);
86             }
87             mPrevFirstFingerX=currX;

```

Obrázek 5.9 : Gesto jedním prstem. Proměnné  $currX$ ,  $currY$  – aktuální souřadnice aktivního prstu,  $mPrevFirstFingerXY$  – předchozí souřadnice aktivního prstu,  $graphParams$  – třída obsahující veškeré parametry grafu

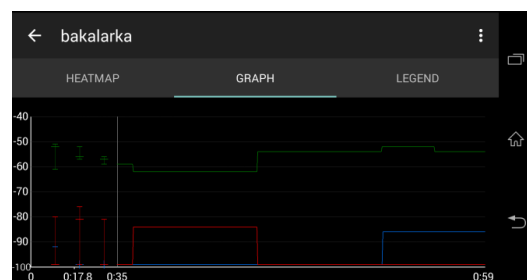
Algoritmus na obrázku 5.10 znázorňuje algoritmus pro gesto dvou prstů. Ve for cyklu začínajícím na řádce 96 se zachytí aktuální pozice prstů. Poté se spočítá současná a předchozí euklidovská vzdálenost prstů. Na řádce 108 dojde k výpočtu velikosti posuvu na základě těchto vzdáleností. Nakonec dojde k posuvu správným směrem tak, jak ukazují řádky 110 - 115. (Obrázek 5.11 a 5.12)

```

88     } else {
89         float currX1 = mPrevFirstFingerX;
90         float currX2 = mPrevSecondFingerX;
91         float currY1 = mPrevFirstFingerY;
92         float currY2 = mPrevSecondFingerY;
93         for (int index = 0; index < touchedPoints; index++) {
94
95             int id = ev.getPointerId(index);
96             if (id == firstActivePointerId) {
97                 currX1 = ev.getX(index);
98                 currY1 = ev.getY(index);
99             }
100             if (id == secondActivePointerId) {
101                 currX2 = ev.getX(index);
102                 currY2 = ev.getY(index);
103             }
104         }
105         int prevEuclid = MyMath.euclidSize(mPrevFirstFingerX, mPrevFirstFingerY,
106             mPrevSecondFingerX, mPrevSecondFingerY);
107         int currEuclid = MyMath.euclidSize(currX1, currY1, currX2, currY2);
108         int diff = Math.abs(prevEuclid-currEuclid)/REDUCTION_VALUE;
109
110         if (prevEuclid>currEuclid) {
111
112             graphParams.moveBorder(diff);
113         }
114         else{
115             graphParams.moveBorder(-diff);
116         }
117         mPrevFirstFingerX = currX1;
118         mPrevSecondFingerX = currX2;
119         mPrevFirstFingerY = currY1;
120         mPrevSecondFingerY = currY2;
121     }

```

Obrázek 5.10 : gesto dvěma prsty. Proměnné currX1, currX2, currY1, currY2 – aktuální souřadnice prvního a druhého aktivního prstu, mPrevFirstFinger, mPrevSecondFingerXY – předchozí souřadnice prvního a druhého aktivního prstu, prevEuclid, currEuclid – současná a minula euklidovská vzdálenost prstů



Obrázek 5.11 a 5.12: ukázka reakce na gesto dvou prstů

## 5.2 Použité knihovny

Během implementace byly použité následující knihovny s opensource licencí:

- **GraphView** v4.0 - knihovna pro vizualizaci dat ve formě grafů různých typů. Knihovna nakonec byla nahrazena vlastní grafovou funkcionalitou. [15]
- **OpenCSV** v3.6 - knihovna pro export dat v poli do souboru CSV [18]
- **Commons Math 3** - knihovna s matematickými funkcemi (percentily, střední hodnoty, odmocniny, mocniny...) [19]



## 6. Testování

Testování by mělo být nedílnou součástí každého vývoje softwaru. Úkolem testů je zajistit správnou funkci programu, zjistit, zdali vrací správné hodnoty a správně reaguje na vstupy. Můžeme také testovat kvalitu kódu, stabilitu softwaru nebo vytížení systému. Testy mohou být buď ruční nebo automatické. Podstatou je ověřit, zda zákazník dostane produkt funkcionality a kvality, kterou požadoval. Je nutné mít na paměti, že testy nikdy nepokryjí všechny možné scénáře a situace, ale měli bychom se snažit, ať je toto pokrytí co největší.

### 6.1 Testování Android aplikací

Při testování android aplikací, stejně tak jako při kterémkoliv jiném softwaru, můžeme rozlišit několik druhů testů. Mezi nejčastější patří jednotkové testy, kterými testujeme správnou funkčnost nejmenších možných jednotek kódu. Konkrétně v jazyce Java jde o JUnit testy testující jednotlivé metody. Jestliže jednotkové testy testují funkčnost základní jednotky, integrační testy ověřují funkčnost těchto jednotek v rámci většího celku. Jinými slovy správnou komunikaci mezi nimi. Nejvyšší vrstvu integračních testů pak tvoří systémové testy. Dalším typem jsou akceptační testy. Doposud zmíněné testy ověřovali funkčnost samotného kódu a tvořili je programátoři. Akceptační testy však ověřují správnou funkčnost běhu aplikace a korektní průběh jednotlivých úkonů. Většinou vychází ze scénářů případů užití a jsou tedy definovány zákazníkem. Vytvoří se akceptační test cases, na základě kterých probíhá testery nebo uživateli testování aplikace za chodu.

#### 6.1.1 Mockování

Mockování se využívá při jednotkových testech. Během testování jen části kódu se může stát, že testovaný blok (funkce, metoda, třída) je závislý na jiných objektech nebo datech. V takovém případě budeme tyto potřebné objekty či data mockovat. Mockovací framework v podstatě vytvoří simulaci potřebného objektu, který má stejné vlastnosti jako objekt „reálný“. Jednotkové testy pak mohou běžet i bez nutnosti běhu aplikace na zařízení. Funkčnost namockovaného objektu lze libovolně upravovat dle potřeb. Jako jedním ze zástupců mockovacích frameworků může zmínit například Robolectric [20].

## 6.2 Testy

### 6.2.1 Akceptační test case

Testování se snažilo pokrýt všechny situace a případy, které mohou během běžného užívání aplikace nastat. Průběh a výsledek každého testu je zachycen ve formě tabulek (6.1 – 6.6).

Tabulka 6.1 : scénář akceptačního testu 1

ID:	#1
Název:	Vstup do aplikace
Účel:	Ověření úspěšného přechodu z hlavní aktivity na aktivitu se seznamem dostupných sítí
Podmínky:	Spuštění aplikace – hlavní aktivity
Kroky:	1. Kliknutí na tlačítko „Chci měřit“
Očekávaný výsledek	1. Spuštění nové aktivity 2. Zobrazení seznamu dostupných bezdrátových sítí
Výsledek testu	OK
Poznámky	

Tabulka 6.2 : scénář akceptačního testu 2

ID:	#2
Název:	Výběr konkrétních přístupových bodů
Účel:	Zjistit zda aplikace bude skutečně měřit přístupové body které vybereme
Podmínky:	Aktivita se seznamem sítí
Kroky:	1. Klikneme v seznamu na položky které chceme 2. U vybraných položek se zaškrtně checkbox 3. Stiskneme tlačítko „Další“
Očekávaný výsledek	1. Spuštění nové aktivity 2. Zobrazí se graf 3. V grafu se budou zobrazovat data pouze námi vybraných přístupových bodů v předchozí aktivitě
Výsledek testu	OK
Poznámky	

Tabulka 6.3 : scénář akceptačního testu 3

ID:	#3
Název:	Multitouch event
Účel:	Otestovat správnou reakci na gesto dvou prstů
Podmínky:	Spuštěné měření, fragment s grafem
Kroky:	1. Položíme na graf dva prsty 2a. Prsty táhneme k sobě 2b. Prsty táhneme od sebe
Očekávaný výsledek	1. Hraniční čára v grafu mění svou polohu v rámci grafu 2. Mění se počet scatterů v levém okně 3. V pravém okně se mění časový úsek zobrazující linky a průběhu signálu
Výsledek testu	OK
Poznámky	

Tabulka 6.4 : scénář akceptačního testu 4

ID:	#4
Název:	Singletouch event
Účel:	Otestovat správnou reakci na gesto jednoho prstu
Podmínky:	Spuštěné měření, fragment s grafem
Kroky:	1. Položíme prst na graf 2a. Táhneme prstem zleva doprava 2b. Táhneme prstem zprava doleva
Očekávaný výsledek	1. V pravé části probíhá posuv datové linky na základě pohybu prstu 2. Na základě posuvu se mění i rozložení scatterů v levém okně 3. Na ose X se mění hodnota času
Výsledek testu	OK
Poznámky	

Tabulka 6.5 : scénář akceptačního testu 5

ID:	#5
Název:	Vypnutí vizualizace vybraných AP
Účel:	Zjistit správnou funkci při vypínání vizualizace jednotlivých AP během měření
Podmínky:	Spuštěné měření
Kroky:	<ol style="list-style-type: none"> <li>1. V jakémkoliv ze tří view klikneme na libovolnou položku v legendě</li> <li>2. Označíme/odznačíme danou položku</li> </ol>
Očekávaný výsledek	<ol style="list-style-type: none"> <li>1. Checkbox u vybraného itemu by měl změnit svůj stav ve všech třech fragmentech s legendou</li> <li>2. Na základě označení/odznačení by se daný AP měl zobrazit/zmizet z grafu a heatmapy</li> <li>3. Měření stále probíhá</li> <li>4. AP opět nezmění svůj stav dokud na něj znovu neklikneme</li> </ol>
Výsledek testu	OK
Poznámky	

Tabulka 6.6 : scénář akceptačního testu 6

ID:	#6
Název:	Export
Účel:	Vyzkoušet funkčnost exportu dat do souboru CSV
Podmínky:	Naměřená data
Kroky:	<ol style="list-style-type: none"> <li>1. Během měření a vizualizace dat klikneme v pravím horním rohu na menu</li> <li>2. Vybereme položku „Export“</li> <li>3. Klikneme na tlačítko „Export do CSV“</li> <li>4. Zadáme název souboru</li> <li>5. Potvrdíme</li> </ol>
Očekávaný výsledek	<ol style="list-style-type: none"> <li>1. Přeruší se měření a spustí se nová aktivita</li> <li>2. Zobrazí se dialog pro zadání názvu souboru</li> <li>3. V případě invalidního vstupu vyskočí hlášení</li> <li>4. Data se zapíší do csv souboru ve složce „bakalářka“, v kořenovém adresáři uložistě</li> </ol>
Výsledek testu	OK
Poznámky	

Tabulka 6.6 : scénář akceptačního testu 7

ID:	#7
Název:	Změna orientace
Účel:	Kontrola reakce na změnu orientace obrazovky
Podmínky:	Spuštěna aplikace
Kroky:	1. Uživatel v kterémkoliv okamžiku běhu aplikace změní orientaci obrazovky
Očekávaný výsledek	1. Přeruší se měření a spustí se nová aktivita 2. Zobrazí se dialog pro zadání názvu souboru 3. V případě invalidního vstupu vyskočí hlášení 4. Data se zapíší do csv souboru ve složce „bakalářka“, v kořenovém adresáři uložště
Výsledek testu	OK
Poznámky	

### 6.2.2 Unit

Jednotkové testování kódu [21] u aplikací vědeckého charakteru, jako je například tato, je složitější z důvodu závislosti naměřených dat. Tato data, společně s dalšími potřebnými moduly pro měření a správný běh aplikace, se proto musí simulovat neboli mockovat [20]. Vzhledem k rozsahu této práce nezbyl z časových důvodů prostor pro vytvoření toho druhu testů.

### 6.3 Shrnutí testů

Provedenými testy jsme nepokryli veškeré scénáře a situace, které by mohly během užívání aplikace nastat. Otestoval jsem však základní části, chování a funkčnost a lze konstatovat, že aplikace splňuje zadané požadavky.



## 7. Závěr

Cílem této práce bylo vytvořit mobilní aplikaci, která bude měřit a vizualizovat průběh signálu bezdrátových sítí. Aplikace nabízí lepší funkčnost v porovnání s dostupnými aplikacemi, v podobě agregace starších dat, nových možností zobrazení měřených dat, jejich perzistence a efektivnějšího využití výkonu měřícího mobilního zařízení.

V rámci této práce proběhla rešerše k problematice bezdrátových sítí, na jejím základě pak proběhlo určení požadavků a následně softwarová analýza. Hlavní část práce spočívala v implementační části projektu. Byly implementovány dva speciální grafy, které zobrazují průběh síly signálu v čase, a to v kompletní či agregované formě.

Načítání dat a řízení vykreslování probíhá samostatně ve dvou nezávislých vláknech. Hlavní vlákno je tak vytížené co nejméně. Aplikace také nabízí přehled všech dostupných sítí a přístupových bodů v dosahu a základní informace ke každému z nich. Dále byla implementována možnost perzistence naměřených dat exportováním do souboru ve formátu CSV. Data tak lze dále využít. Na závěr proběhlo testování běžného užívání aplikace

V budoucnu by se mohla rozšířit funkčnost aplikace o několik dalších vlastností. Jedním z nich je vizualizace využitých kanálů pro každý AP, tuto funkci mají některé konkurenční aplikace. Další užitečnou vlastností by rozhodně byla možnost vztahovat naměřené hodnoty k určitému místu a datu. Zpětným načtením dat a schématu poschodí by pak bylo možné vizualizovat signál pro určité místo a čas v budově. Díky použití fragmentů lze v budoucnu využít velikost displeje tabletů a zobrazit tak uživatelům všechna View s grafy najednou.





## Literatura

- [1] Onkar Pathak, Pratik Palaskar, Rajesh Palkar, Mayur Tawari. Wi-Fi Indoor Positioning System Based on RSSI Measurements from Wi-Fi Access Points - A Tri-lateration Approach. In: Academia.edu [online]. University of Cape Coast. [vid. 10.1.2016]. Dostupné z: [http://www.academia.edu/6895582/WiFi\\_Indoor\\_Positioning\\_System\\_Based\\_on\\_RSSI\\_Measurements\\_from\\_Wi-Fi\\_Access\\_Points\\_A\\_Tri-lateration\\_Approach](http://www.academia.edu/6895582/WiFi_Indoor_Positioning_System_Based_on_RSSI_Measurements_from_Wi-Fi_Access_Points_A_Tri-lateration_Approach)
- [2] Gregory Brown. i-programmer.info [online]. O'Reilly, 2009. [vid. 10.1.2016]. Dostupné z: <http://www.i-programmer.info/programming/hardware/2767-how-wifi-works.html>
- [3] CCM [online]. Communitic International. [vid. 10.1.2016]. Dostupné z: <http://ccm.net/faq/298-what-is-wi-fi-and-how-does-it-work>
- [4] How-To Geek [online]. How-To Geek, LLC. [vid. 10.1.2016]. Dostupné z: <http://www.howtogeek.com/204697/wi-fi-security-should-you-use-wpa2-aes-wpa2-tkip-or-both>
- [5] David Rohleder, Václav Lorenc. 802.1X - autentizace v počítačových sítích. In: webserver.ics.muni.cz [online]. Ústav výpočetní techniky, 2011. [vid. 10.1.2016]. Dostupné z: <http://webserver.ics.muni.cz/bulletin/articles/590.html>
- [6] Jan Dolejš. Proč je síla signálu telefonu uváděna v záporných hodnotách a co vyjadřuje? . In: svetandroida.cz [online]. Svetandroida.cz, 2013. [vid. 10.1.2016]. Dostupné z: <http://www.svetandroida.cz/sila-signalu-201503>
- [7] Lisa Phifer. Understanding WLAN signal strengt. In: techtarget.com [online]. TechTarget Inc, 2005. [vid. 10.1.2016]. Dostupné z: <http://searchmobilecomputing.techtarget.com/tip/Understanding-WLAN-signal-strength>
- [8] en.wikipedia [online]. Wikimedia Foundation. [vid. 10.1.2016]. Dostupné z: [https://wikimediafoundation.org/wiki/Privacy\\_policy](https://wikimediafoundation.org/wiki/Privacy_policy)
- [9] 4Gon Solution [online]. 4Gon. [vid. 10.1.2016]. Dostupné z: [http://www.4gon.co.uk/solutions/technical\\_factors\\_affecting\\_wireless\\_performance.php](http://www.4gon.co.uk/solutions/technical_factors_affecting_wireless_performance.php)
- [10] Kemisola Ogunjemilua, John N. Davies, Vic Grout, Rich Picking. An Investigation into Signal Strength of 802.11n WLAN. In: Glyndwr.ac.uk [online]. Glyndŵr University. Dostupné z: [http://www.glyndwr.ac.uk/computing/research/pubs/sein\\_odgp.pdf](http://www.glyndwr.ac.uk/computing/research/pubs/sein_odgp.pdf)

[11] radio-electronics.com [online]. Adrio Communications Ltd. [vid. 10.1.2016].  
Dostupné z: <http://www.radio-electronics.com/info/propagation/multipath/multipath-propagation-basics-tutorial.php>

[12] *Google Play* [online]. Google Inc. [vid. 10.1.2016]. Dostupné z:  
<https://play.google.com/store>

[13] *Google Play, WiFi Signal Strength* [software]. Dostupné z:  
<https://play.google.com/store/apps/details?id=com.cls.wificls>. Android 4.1 a vyšší

[14] *Google Play, Wifi Analyzer* [software]. Dostupné z:  
<https://play.google.com/store/apps/details?id=com.farproc.wifi.analyzer>. Android 2.3 a vyšší

[15] opensource, GRAPH VIEW [software]. Dostupné z: <http://www.android-graphview.org/> Android

[16] en.wikipedia [online]. Wikimedia Foundation. [vid. 10.1.2016]. Dostupné z:  
[https://en.wikipedia.org/wiki/Comparison\\_of\\_mobile\\_operating\\_systems](https://en.wikipedia.org/wiki/Comparison_of_mobile_operating_systems)

[17] *Gitlab* [online]. opensource. [vid. 10.1.2016]. Dostupné z:  
<https://gitlab.fel.cvut.cz/furstrom/bakalarka.git>

[18] *sourceforge.net* [online]. Slashdot Media. [vid. 10.1.2016]. Dostupné z:  
<http://opencsv.sourceforge.net/>

[19] *apache.org* [online]. Apache. [vid. 10.1.2016]. Dostupné z:  
<https://commons.apache.org/proper/commons-math/>

[20] *Robolectric* [online]. Maven. [vid. 10.1.2016]. Dostupné z: <http://robolectric.org/>

[21] *developer.android.com* [online]. Google Inc. [vid. 10.1.2016]. Dostupné z:  
[http://developer.android.com/tools/testing/testing\\_android.html](http://developer.android.com/tools/testing/testing_android.html)

## A. Obsah přiloženého CD

- **Měření\_a\_vizualizace\_signalu\_bezdratovych\_siti\_v\_interieru.pfd** – elektronická verze bakalářské práce ve formátu pdf
- **zdrojovy\_kod.rar** – soubor obsahující finální verzi zdrojového kódu
- **bakalarka.apk** – instalační soubor aplikace
- **uml.rar** - soubor s diagramy ve formátu jpg uvedených v této bakalářské práci
- **screenshot.rar** – soubor se screenshoty aplikace ve formátu jpg uvedených v této bakalářské práci