

# Jednoduchý WiFi lokátor

*ITU projekt, 2014*

Číslo zadání projektu: 55

Kapitán týmu: Ladislav Šulák (xsulak04)

Členové týmu: Ladislav Šulák (xsulak04), Ján Spišiak (xspisi03)

Datum: 20. 11. 2014

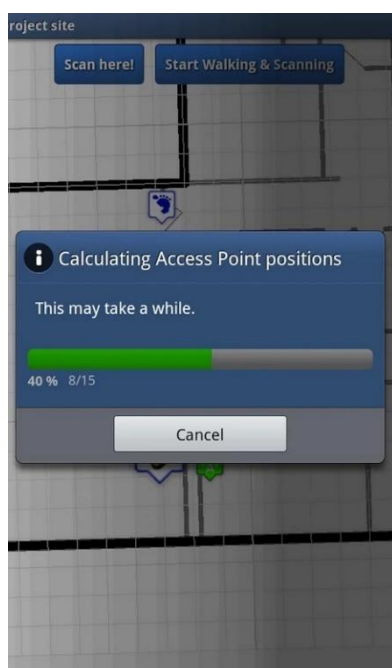
## Abstrakt

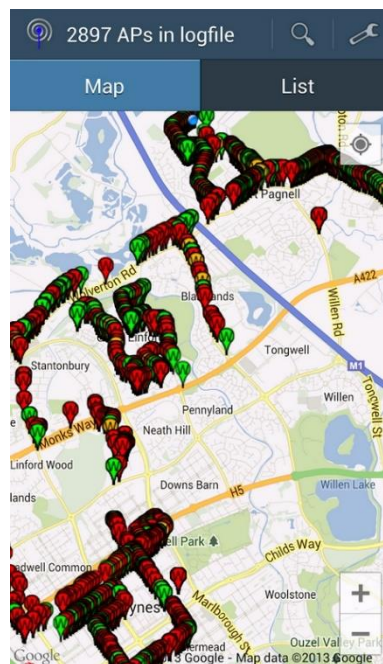
Cieľom projektu bolo vytvoriť užívateľské rozhranie ktoré má ako hlavnú činnosť určovanie približnej polohy WiFi zariadení na základe intenzity signálu AP. Aplikácia má zobrazovať aktuálnu mapu okolia. Pri návrhu a implementácii aplikácie je nutné vysporiadať sa s viacerými problémami:

- Zobrazovanie mapy okolia podľa aktuálnej polohy
- Výpočet približnej lokality AP podľa intenzity signálu. Podľa toho bude možné sa dopracovať ku kružniciam (resp. medzikružiam). Nastáva problém: v ktorom kvadrante sa nachádza zariadenie?
- Používanie na odlišných platformách (hlavne mobilné, no aj iné zariadenia) zaistenie správnej kompatibility (túto možnosť zadanie nešpecifikuje, no prenosnosť medzi rôznymi zariadeniami treba brať do úvahy vždy, napríklad odlišné rozlíšenia obrazoviek)
- Grafický dizajn ktorý zaujme a zjednoduší používanie aplikácie. Nie tak problém, ako nutnosť. Pri implementácii aplikácie sa bude využívať rozhranie QT v jazyku Python.

## Úvod

Aplikácii, ktoré určia polohu AP je naozaj veľmi málo. Príklad podobnej aplikácie je WiFi Compass [\[1\]](#), WiFiFoFum - WiFi Scanner [\[2\]](#), alebo WiFi Tracker [\[3\]](#).





Uvedené aplikácie majú niektoré z týchto nedostatkov:

- Chýba v nich aktuálna, presná a úplná mapa okolia (vnútro budov, popr. vonkajšie okolie). Teda ak by bola aplikácia použitá v novom prostredí, mapa by sa nezobrazila.
- Nepočítajú polohu AP, iba si po pripojení pamätajú kde sa približne nachádzal, a uložia zariadenia do súboru.
- Grafický dizajn mohlo byť spracovaný na lepšej úrovni.
- Uvedené aplikácie fungujú pod OS Android, teda pri inom OS by bolo nutné nainštalovať nejaký Android Emulátor. Aplikácií s daným zameraním pre iné OS než Android je minimum.

WiFi Compass má najmenej z vymenovaných nedostatkov, a principiálne je najviac podobný našej aplikácii. Obsahuje funkciu Scan Here, teda užívateľ zadá svoju aktuálnu pozíciu, premiestni sa, zadá vykonávanie tejto funkcie znova, a tak sa dopracuje k relatívne presným výsledkom. Naš návrh je podobný, avšak nepoužívame GPS (predpoklada sa aj to, že užívateľ počas používania aplikácie nebude mať prístup na internet), a teda užívateľ musí zadať miesto kde sa nachádza. Nie je teda možné určiť pozíciu v reálnom čase - postupne pri premiestňovaní zariadenia užívateľa.

Vačšina aplikácii nezobrazuje polohu AP, iba sa dokážu pripojiť, zmerať intenzitu spojenia, poprípade zobrazíť ďalšie atributy (typ, zabezpečenie,...). Naša aplikácia je užitočná v tom, že ak bude použitá na prenosné zariadenie (mobil, notebook) užívateľ bude môcť zlepšiť silu svojho signálu priblížením sa.

## Štúdia technológií

### Python

Backend aplikácie je python, multiplatformový interpretovaný jazyk, ktorý je vhodný aj na rýchle prototypovanie a prácu s dátami. V našej aplikácii používame python ako vrstvu medzi operačným systémom, ktorý nám poskytne dáta o bezdrátových sieťach, a grafickým rozhraním, ktoré tieto dáta vhodným spôsobom zobrazia užívateľovi. Existujú dve verzie, python2 a python3, rozdiel je hlavne v množstve modulov, avšak väčšina už podporuje aj python3.

### Python + Qt

Komunikácia medzi pythonom a Qt musí byť zabezpečená pomocou modulu. Existujú 3 hlavné alternatívy a to: PyQt, PySide a PyOtherSide. PyQt je vydané pod GPL a plne podporuje python3 a Qt5, PySide má licenciu LGPL ale podpora pre python3 a Qt5 nie je dokončená. PyOtherSide má opačnú filozofiu. Je to plugin do QML (deklaratívny jazyk od Qt) ktorý umožňuje prístup k python interpretu z prostredia QML. Je tým pádom pomerne nezaťažujúci na systémové zdroje.

### Qt

Qt ponúka viacero technológií na tvorbu užívateľského prostredia a to: Qt Widgets (desktopové rozhrania, štandardizovaný statický výzor), Qt Quick (QML, dynamický, podpora dotykových zariadení, javascript), Qt Webkit (web browser).

### Výber

Pri výbere sme zohľadnili možnosť podpory na mobilných a dotykových zariadeniach. Taktiež aktuálnosť technológií a tým aj nároky na údržbu. PySide nemá v tomto čase dokončenú podporu Qt5 a preto nie je vhodnou voľbou. Tým pádom môžeme použiť python3. Rozhodli sme sa používať PyOtherSide.

## Návrh aplikácie

### Metódy

Aplikácia najprv zisťuje silu Wi-Fi signálu sietí prostredníctvom operačného systému. Existuje viacero spôsobov, no najjednoduchším je použitie utility *iwlist* (*implementované v pythone*). Tá avšak potrebuje správckovské práva, pokiaľ nie je systém nakonfigurovaný inak. Toto môže byť problém na zablokovaných mobilných zariadeniach. Inak sa tieto informácie dajú zistiť cez rôzne užívateľsky nainštalované programy na správu sieťových pripojení.

V ďalšom kroku treba tieto informácie spracovať a vyhodnotiť lokáciu AP. Na to nám však nestačia dáta z jedného merania. To nám určí len približnú vzdialenosť, dve merania by mali určiť rovinu, a tri by mali stačiť na trianguláciu pozície zdroja signálu. Na presnosť tejto triangulácie samozrejme veľmi vplyva anatomia prostredia. Wi-Fi signál je mikrovlnné žiarenie ktoré interaguje s veľkým množstvom materiálov rôznymi spôsobmi: difrakciou, rozptylom a odrazom. K zariadeniu sa teda môže dostať signál z rôznych smerov, z rôznymi silami a aj s oneskorením.

Zároveň je tu problém so znalosťou polohy zariadenia ktoré robí meranie. V prípade mobilných zariadení existuje možnosť použiť GPS modul, prípadne iné senzory na detekovanie polohy, prípadne pohybu. Na prenosných počítačoch avšak takáto možnosť často chýba a preto je program odkázaný

na užívateľa aby zadal polohu zariadenia. Po načítaní mapy užívateľ zadá svoju polohu kliknutím na mapu. Mapu môže meniť, pridávať nové mapy – vo formáte .png.

Pri nahrávaní mapy okolia sa ponúkali viaceré možnosti - použiť existujúce mapy volne dostupné na internete – napríklad použiť mapy projektu OpenStreetMap[4], no z dôvodu veľkej veľkosti súborov by sa pracovalo len s mapou určitej lokality. My sme sa rozhodli pre jednoduché riešenie, ktoré by iste využili užívatelia – pridanie vlastných map v .png formáte do priečinku /map/nazov\_miesta/. Názov miesta bude udávať napríklad mesto, názov budovy , pričom je možnosť uložiť aj jednotlivé poschodia, ktoré budú v aplikácii aj sprístupnené.

Užívateľ si zároveň môže vybrať ktoré Wi-Fi siete bude chcieť zobrazíť na mape.

## Návrh užívateľského rozhraní

### Štruktúra

Aplikácia bude rozdelená na bloky, každý s inou funkcionalitou. V prvom rade musí užívateľ vybrať ktoré z Wi-Fi sietí chce triangulovať. Riešením je jednoduchý zoznam s možnosťou výberu viacerých položiek, ktorý zároveň môže poskytovať základné informácie o dostupných sieťach. V ďalšom bloku musí užívateľ zaznamenať polohy meraní, jedným spôsobom je použiť mapu na ktorej je možné pridávať body. Zdroj tejto mapy môže byť sám užívateľ, alebo je možné použiť vo vonkajších prostrediach aj satelitné mapy. Následne sa dá zobrazíť poloha AP na mape, alebo ak zariadenie poskytuje informácie o natočení zariadenia, je možné vytvoriť 3D kompas ukazujúci smer a vzdialenosť k cieľu.

### Proces

Štruktúra, by mala, a odzrkadľuje proces použitia programu. Jednotlivé bloky štruktúry sú teda identické s krokmi procesu, čo v našej aplikácii je jednoduchá postupnosť:

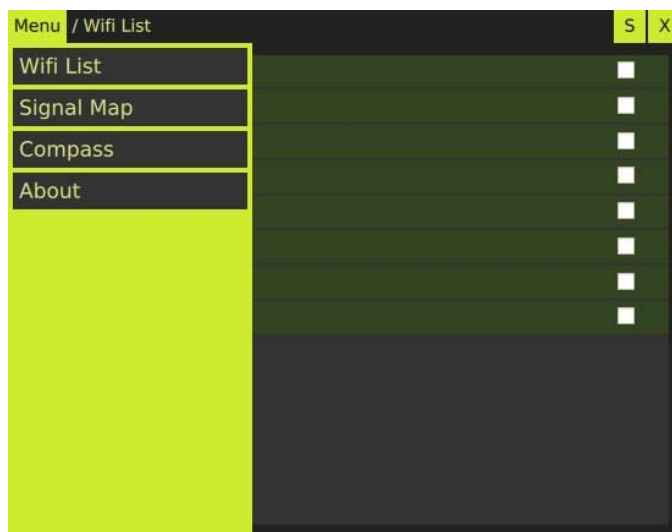
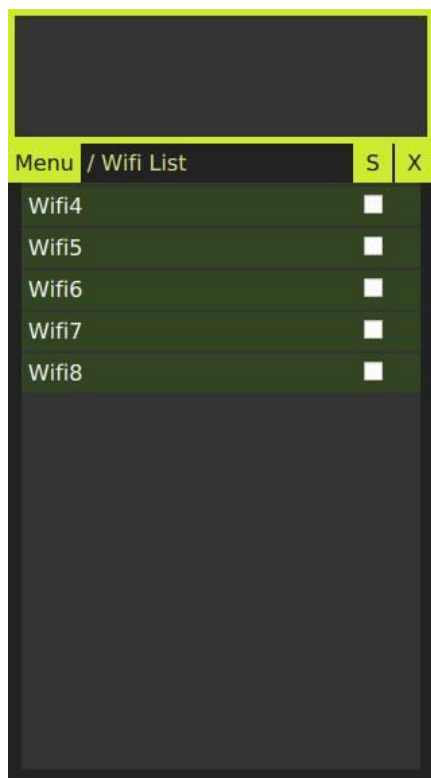
1. Výber sietí
2. Výber mapy
3. Určiť polohu meraní
4. Zobrazíť výsledok, teda poloha AP
5. Umožniť opakovanie procesu

### Design

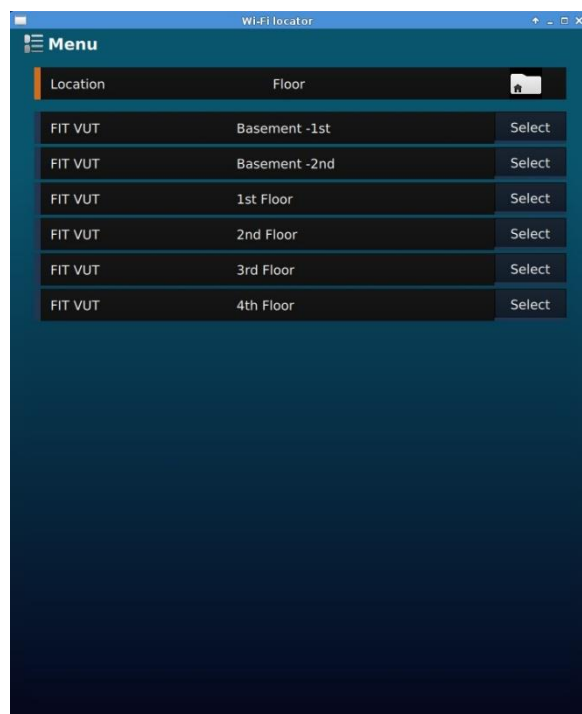
Návrh bol koncipovaný tak, aby bola aplikácia čo najprehľadnejšia bez zbytočných funkcií ktoré navyše vyžadujú komplikovanejšiu a zdĺhavejšiu postupnosť krokov pre užívateľa.

Aplikácia bola tiež navrhnutá tak, aby sa na rýchlo a jednoducho prezentovali dáta a požiadavky v danom kroku.

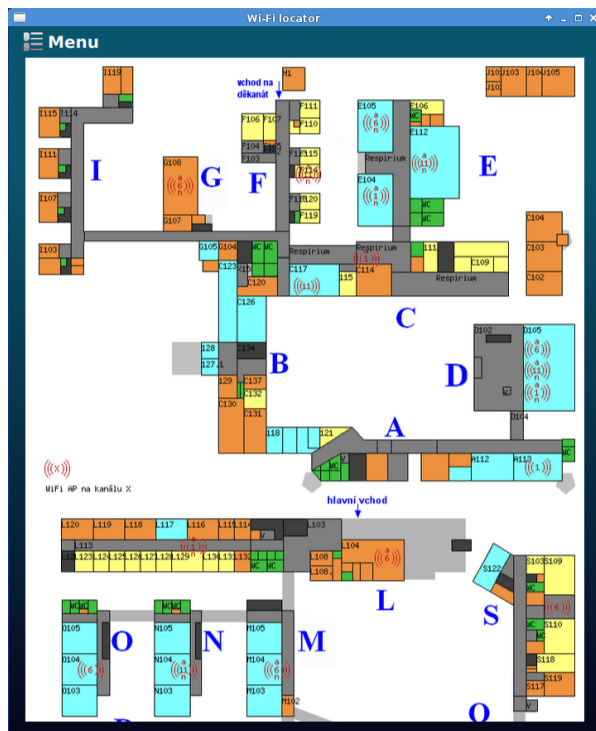
Počiatky implementácie dizajnu boli približne ako na nasledujúcich obrázkoch.



Finálna verzia má dizajn znázornený na nasledujúcich obrázkoch.



Menu			
ESSID	Quality	Auth	Select
hello	65.7%	WPA	Select
404 error	50.0%	WPA2	Select
\x00\x00\x00\x00	38.6%	WPA2	Select
Wifik	32.9%	WPA	Select
C0-51-4C-17-4B-62	44.3%	WPA2	Select
N24PU1	44.3%	No	Select
fitsucks	31.4%	WPA2	Select
Tenda_5FC680	32.9%	Unknown	Select



## Implementácia

Na implementáciu grafického užívateľského rozhrania a backend aplikácie boli využité technológie python3.4.2, PyOtherSide a Qt 5.3.2 pod operačným systémom Linux. Vymenované technológie je možné použiť aj v OS Windows, avšak implementácia back-end aplikácie, teda spracúvanie Wi-Fi signálu pod OS Windows nebolo uvažované. Program bol vyvíjaný pod operačnými systémami Archlinux a Fedora 20.

Aplikácia nie je plne multiplatformová – pre spracovanie Wi-Fi signálov sa používa iwlist (Linux wireless network tool) z OS Linux, teda aplikácia nie je kompatibilná s operačným systémom Windows. Nebolo testované na dotykových zariadeniach.

Aplikácia sa skladá z nasledujúcich **zdrojových súborov**:

*About.qml, chooseMap.qml, main.qml, MapPage.qml, Quit.qml, ScanPage.qml, Wifi.py*

## Ján Spišiak - implementácia

**Hlavné okno** aplikácie pozostáva z menu, ktoré je implementované v súbore *main.qml*, ktoré tvorí 5 buttonov. Obsahuje hlavný element *ApplicationWindow*, ktoré reprezentuje hlavné okno aplikácie. Obsahuje niekoľko elementov typu *Rectangle*, *Image*, *ListView*, *Repeater*, *MouseArea*, *Text*, *Gradient* a niekoľko ďalších. Python backend – script *wifi.py* na spracovanie výstupu z *iwlist scan()*, *prepojenie qml s pythonom*, *následné zobrazenie dát v List View s interakciou*.

Implementácia asynchrónného *Loaderu*, elementu na načítavanie stránok aplikácie.

Element *Rectangle* obsahuje prvky a deklarácie pre definovanie vzhľadu, napríklad pre uvedený panel s menu buttonmi, alebo obsahuje volanie iných častí aplikácie – stránok, definovaných v štruktúre *pagesList*, ktoré sa načítajú z iného súboru.

**Choose Wi-Fi** : button v hlavnom paneli, implementácia v súbore *ScanPage.qml*. Pre načítanie a spracovanie Wi-Fi signálu je implementovaných niekoľko funkcií v jazyku Python.

## Ladislav Šulák – implementácia

**Choose Map**: hlavný element *Rectangle* obsahuje prvky definujúce nie len vzhľad ale na niektorých miestach (v *MouseArea*) bolo potrebné použiť javascript, napríklad pre spracúvanie názvov súboru, alebo pre definovanie textu, ktorý bude následne použitý v tabuľke. Pomyšlná tabuľka obsahuje názvy priečinkov s názvom lokácie. V priečinkoch sa nachádzajú mapy vo formáte .png, ktoré sa následne zobrazia užívateľovi. Po kliknutí na meno mapy sa v aplikácii hneď zobrazí, to je zaistené implementáciou v súbore *MapPage.qml*. Mapa sa môže zobrazíť aj inak, pomocou buttonu **Map** v hlavnom menu. Meno súboru s mapou sa predáva pomocou premennej *mapSource*, ktorej hodnotu je možné získať v každom .qml súbore podľa *ID* elementu kde bola definovaná, a jej názvu. Ak mapa nebola ešte užívateľom definovaná, teda napríklad hneď po spustení aplikácie, implicitne nastavená mapa je mapa 1.poschodia FIT VUTBR. Premenná *mapSource* je definovaná v súbore *main.qml*.

**Map**: princíp popísaný vyššie. Zobrazuje mapu ktorú užívateľ zvolí, implicitne je zvolená mapa 1.poschodia FIT VUTBR. Implementované v súbore *MapPage.qml*. Je možné mapu približovať, pohybovať sa po nej, a určiť polohu užívateľa. Mapa a kurzor, kde sa používateľ nachádza sú definované v elementoch *Image*.

**About**: obsahuje element *Rectangle* ktorý ma niekoľko prvkov typu *Text*, ktoré obsahujú základné informácie o autoroch a aplikácii a krátky návod pre jednoduché použitie aplikácie. Implementované v súbore *About.qml*.

**Quit**: po kliknutí na tento button sa načíta súbor *Quit.qml*, ktorý obsahuje okrem elementu *Rectangle* komponentu *onCompleted*, ktorá po načítaní súboru vypne aplikáciu.

## Testovanie

Testovanie má za účel zistiť, či aplikácia je funkčná s ohľadom na zistenie AP v okolí a náročnosť aplikácie pre užívateľov – ako sa orientujú v aplikácii: intuitívnosť, pamätanie rozloženia ovládacích prvkov. Ak sa v čase testovania ukáže, že aplikácia je neprehľadná, zvážime zmenu vzhľadu a umiestnenia ovládacích prvkov, možno pridanie jednoduchšej nápovedy.

Testovanie grafického užívateľského rozhrania za účelom čo najjednoduchšieho používania aplikácie. Cieľ: užívateľ používa aplikáciu rád a často. K dosiahnutiu výsledku je potrebných čo najmenej krokov.

Testovanie bude realizované pomocou merania času od zapnutia aplikácie k získaniu výsledkov. Testovacie skupiny užívateľov:

1. technicky zdatní ľudia do veku 30 rokov
2. skupina ľudí do 30 rokov nezaoberajúcich sa technológiami
3. skupina ľudí nad 30 rokov nezaoberajúcich sa technológiami

V nasledujúcom kroku sa vykoná aritmetický priemer nad časmi v každej skupine, a porovnajú sa. Postup sa zopakuje 3x. Ak sa časy budú skracovať, znamená to, že naša aplikácia je user-friendly, je ľahké si zapamätať postupnosť operácii ktorá vedie k výsledku.

Ak časy medzi skupinami nebudú príliš odlišné, bude to znamenať, že aplikácia je dobre ovládateľná medzi rôznymi druhmi užívateľov.

Výsledky testovania priniesli to, že sa užívatelia vedía v aplikácii orientovať, teda v nej nie je prítomných veľa zdĺhavých a zbytočných funkcií alebo krokov. To je základný predpoklad na to, aby aplikáciu užívatelia často využívali, a nehladali alternatívy.

## Plán činnosti a rozdelenie medzi členmi tímu

Rozdelenie nebolo striktné.

Ladislav Šulák – pridávanie, zobrazovanie, spracovanie máp, zobrazovanie AP v aplikácii.  
Implementácia GUI. Návrh testovania, jeho realizácia.

Ján Spišiak – pracovanie s Wi-Fi signálom, implementácia grafického užívateľského rozhrania, zobrazovanie na mape. Testovanie.



## Reference

- <https://play.google.com/store/apps/details?id=at.fhstp.wificompass>
- <https://play.google.com/store/apps/details?id=com.dynamicallyloaded.wififorum>
- <https://play.google.com/store/apps/details?id=org.prowl.wifiscanner>
- <http://osm.kyblsoft.cz/archiv/>