

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra Informatiky a kvantitativních metod

Mapa WiFi sítí
Bakalářská práce

Autor: Roman Vlček
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Pavel Kříž, Ph.D.

Hradec Králové

duben 2016

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 20.4.2016

Roman Vlček

Poděkování:

Děkuji vedoucímu bakalářské práce Ing. Pavlu Křížovi, Ph.D. za metodické vedení práce a poskytnutí potřebných informací k jejímu vypracování. Dále děkuji hkfree.org, z.s. za poskytnutí testovacích dat pro ověření přesnosti lokalizace.

Anotace

Práce se zabývá tvorbou aplikace pracující s WiFi databázemi. Cílem je vytvoření webové aplikace, která bude získávat data z dostupných databází Wifileaks a Wigle a lokalizačních služeb Google, přičemž uživatel bude mít možnost o data požádat. Získané přístupové body budou zobrazeny na mapě s možností zobrazení jejich detailních informací.

Obsah práce je dělen do několika částí. V první části bude čtenář seznámen s účelem a způsobem tvorby WiFi databází a použitými službami. Druhou částí bude analýza aplikace a seznámení s použitými technologiemi. Třetí část se týká samotné implementace aplikace a v závěru práce budou data, získaná z použitých zdrojů, porovnána se známými daty a porovnána přesnost jednotlivých zdrojů dat.

Annotation

Title: Wireless Networks Map

This thesis deals with development of application working with WiFi databases. The objective is to create a web application that will retrieve the data from the available databases such as Wifileaks and Wigle, and from Google localization services. User will have option to create requests for data. Gained access points will be displayed on the map with option to show detailed information about it.

The content of this thesis is divided into several parts. In the first part the reader will be familiar with the purpose and ways of creating WiFi databases and with used services. Second part is about analysis of the application and description of used technologies. The third part concerns the application implementation and at the end there will be data obtained from sources compared with known data and comparison of resources accuracy.

Obsah

1	Úvod.....	1
2	Cíl práce.....	1
3	Přehled WiFi databází	2
3.1	Účel WiFi databází	2
3.1.1	Určování polohy	2
3.1.2	Nalezení zdroje rušení	2
3.1.3	Hledání volných sítí	3
3.1.4	Analýzy	3
3.2	Vytváření WiFi databází	3
3.2.1	Metody získávání WiFi sítí	4
3.3	Použité databáze.....	5
3.3.1	Wifileaks.....	5
3.3.2	Wigle.....	5
3.3.3	Google.....	5
4	Analýza	7
4.1	Požadovaná funkcionality.....	7
4.1.1	Získávání dat	7
4.1.2	Vizualizace dat	11
4.2	Použité technologie	13
4.2.1	Framework.....	13
4.2.2	PHP frameworky.....	13
4.2.3	Funkce a principy Nette Frameworku.....	14
4.2.4	API map	16
4.2.5	Google Maps API	17
4.2.6	Další použité technologie	19

4.3	Databázová vrstva aplikace.....	22
4.3.1	Databáze pro získávání dat.....	22
4.3.2	Databáze vizualizace dat.....	27
4.4	Aplikační vrstva aplikace	27
5	Návrh a implementace	30
5.1	Získávání dat	31
5.1.1	Určení dat – vytvoření požadavků	31
5.1.2	Zpracování požadavků.....	36
5.1.3	Získání přístupových bodů	40
5.2	Vizualizace dat.....	45
5.2.1	Generování překryvné vrstvy	45
5.2.2	Překryvná vrstva – optimalizace	52
5.2.3	Detail bodu	55
5.2.4	Zobrazení aktuálního nastavení mapy	56
5.2.5	Vyhledávání v Google mapách.....	57
5.3	Další funkce aplikace.....	57
5.3.1	Statistiky	57
5.3.2	Vytvořené požadavky.....	58
5.3.3	Export do CSV.....	58
5.3.4	Sdílení pomocí odkazu.....	59
5.4	Problémy při implementaci.....	60
6	Zhodnocení výsledků	62
6.1	Metoda porovnání zdrojů dat	62
6.1.1	Haversinův vzorec.....	62
6.2	Porovnání přesnosti zdrojů.....	63
6.2.1	Porovnání pro jeden bod	63

6.2.2	Výsledky porovnání všech bodů	64
6.2.3	Porovnání zdrojů dat.....	66
7	Závěry a doporučení	68
8	Seznam použité literatury.....	70
9	Přílohy	72

Seznam obrázků

Obrázek č. 1	WiFi hotspoty v aplikaci Mapy na Windows Phone	3
Obrázek č. 2	Kód pro načtení API Google Maps	17
Obrázek č. 3	Kód pro nastavení a zobrazení mapy.....	17
Obrázek č. 4	AJAX s využitím JQuery	20
Obrázek č. 5	Data ve formátu JSON.....	21
Obrázek č. 6	Příklad použití PHP CURL pro získání měření pro MAC adresu z Wigle	21
Obrázek č. 7	Databáze - požadavky od uživatelů	22
Obrázek č. 8	Databáze - rozpracované požadavky	24
Obrázek č. 9	Databáze - notifikace po získání dat.....	25
Obrázek č. 10	Databáze - tabulka s přístupovými body.....	26
Obrázek č. 11	Databáze - ostatní tabulky	27
Obrázek č. 12	Diagram funkcionality aplikace	30
Obrázek č. 13	Vytvoření požadavku zadáním rozsahu zeměpisných souřadnic	32
Obrázek č. 14	Rozdělení požadavku na několik menších.....	34
Obrázek č. 15	Rozsah barev přístupových bodů v překryvné vrstvě Wigle	37
Obrázek č. 16	Ukázka překryvné vrstvy Wigle.....	38
Obrázek č. 17	Zpracování požadavku - Wigle	39
Obrázek č. 18	Ukázka struktury TSV souboru	40
Obrázek č. 19	Ukázka návaznosti dlaždic.....	46
Obrázek č. 20	Výpočet konverzního poměru zeměpisné šířky a pixelu	48
Obrázek č. 21	Body označené podle parametrů (SSID eduroam).....	50
Obrázek č. 22	Zobrazení pouze jedné sítě.....	50

Obrázek č. 23 Mód vypočtená poloha bodu.....	51
Obrázek č. 24 Mapa s překryvnou vrstvou	52
Obrázek č. 25 Použití HTML geolokace v kombinaci s Google Maps	55
Obrázek č. 26 Okno s detailem bodu	56
Obrázek č. 27 Zobrazení aktuálního nastavení mapy	57
Obrázek č. 28 Skript pro přidání obdélníku do Google Mapy.....	58
Obrázek č. 29 Vytvoření CSV v PHP	59
Obrázek č. 30 Implementace Haversinova vzorce v PHP.....	63
Obrázek č. 31 Body v databázi pro určitou MAC adresu	64
Obrázek č. 32 Graf porovnání přesnosti zdrojů	66

Seznam tabulek

Tabulka č. 1 Porovnání zdrojů dat	10
Tabulka č. 2 Význam hodnot databázové tabulky požadavků	23
Tabulka č. 3 Význam hodnot stavu importu MAC adres.....	24
Tabulka č. 4 Význam hodnot v tabulce přístupových bodů.....	26
Tabulka č. 5 Utility - odpovědnost, funkcionalita.....	28
Tabulka č. 6 Prezentéry – odpovědnost, funkcionalita	28
Tabulka č. 7 Servisní třídy – odpovědnost, funkcionalita	29
Tabulka č. 8 Ukázka počátečního rozdělení plochy.....	38
Tabulka č. 9 Význam hodnot zabezpečení	40
Tabulka č. 10 Hodnoty vrácené z Wiple API pro rozsah zeměpisných souřadnic....	42
Tabulka č. 11 Význam hodnot - Wiple API detail MAC adresy	43
Tabulka č. 12 Odstíny barvy bodů podle zdroje dat.....	49
Tabulka č. 13 Čas expirace cache podle úrovně přiblížení mapy.....	53
Tabulka č. 14 Význam atributu wep získaného z Wiple	61
Tabulka č. 15 Porovnání zdrojů pro vybrané přístupové body	65
Tabulka č. 16 Porovnání zdrojů dat.....	66
Tabulka č. 17 Porovnání přesnosti zdrojů (uvedena chyba v metrech).....	67

Seznam pojmů

AJAX – Asynchronous JavaScript and XML, asynchronní zpracování požadavků na webu

Altitude – Nadmořská výška

API – Application Programming Interface, funkce programu nebo knihovny, poskytované vývojářům

BTS – Base Station System – systém základnových stanic mobilních sítí

Cache – mezipaměť aplikace sloužící k urychlení načítání dat

CSV – souborový formát pro výměnu dat, lze s ním pracovat například v Excelu

CURL – knihovna sloužící k výměně dat s použitím různých protokolů

DHCP – Dynamic Host Configuration Protocol, protokol pro automatickou konfiguraci sítě

DOM – Document Object Model, objektově orientovaná reprezentace HTML dokumentu

Excel – tabulkový procesor od společnosti Microsoft, součást kancelářského balíku Office

GPS - Global Position System, globální polohový systém

GUI – Graphical User Interface, grafické uživatelské rozhraní, část aplikace s kterou pracuje uživatel

Haversinův vzorec – metoda výpočtu vzdálenosti mezi dvěma body na kulové ploše

HTML – HyperText Markup Language, značkovací jazyk sloužící k tvorbě webových stránek

Interval beacon – časový interval, v kterém přístupový bod rozesílá informace o vysílané síti

jQuery – JavaScript knihovna usnadňující manipulaci s DOM, událostmi, a další

JSON – JavaScript Object Notation, datový formát určený pro přenos dat, bývá používán v kombinaci s jazykem JavaScript

Kanál sítě – rozsah frekvencí, na kterých přístupový bod vysílá signál

Latitude – zeměpisná šířka

Longitude – zeměpisná délka

MAC adresa – jedinečný identifikátor síťového zařízení, takzvaná fyzická adresa, je 48bitů dlouhý, zapisuje se jako šestice dvouciferných hexadecimálních čísel, například: 01:23:45:67:89:ab

MVC – model view controller, softwarová architektura, která rozděluje aplikaci na datový model, uživatelské rozhraní a řídicí třídy

MySQL – databázový systém, často nasazován v kombinaci s PHP

PHP – PHP: Hypertext Preprocessor, skriptovací jazyk určený především pro tvorbu dynamických webových stránek

PDO – PHP Data Objects, rozhraní jazyka PHP pro pohodlnější práci s databází, při správném použití chrání před útoky typu SQLInjection

PNG – Portable Network Graphics, grafický formát pro bezztrátovou kompresi rastrové grafiky, umožňuje průhlednost, je určený pro použití na webu

QoS – Quality of Service, funkce která v počítačových sítích slouží k řízení datových toků podle typu dat

Regulární výraz – řetězec popisující množinu řetězců pomocí zástupných znaků

SQLInjection – útok na databázovou vrstvu využívající vsunutí (injection) kódu přes neošetřený vstup a vykonání vlastního SQL dotazu

SSID – Service Set Identifier, identifikátor bezdrátové sítě, název sítě vysílaný přístupovým bodem

StreetView – Služba společnosti Google integrovaná v Google Maps, nabízí panoramatické pohledy do ulic

otevřená, WEP, WPA1, WPA2 – typ zabezpečení bezdrátové sítě

URL – Uniform Resource Locator, řetězec sloužící k specifikaci umístění zdrojů

WiFi – označení pro několik standardů IEEE 802.11 bezdrátové komunikace v počítačových sítích

XML – Extensible Markup Language, rozšiřitelný značkovací jazyk, slouží k výměně strukturovaných dat mezi aplikacemi

XSS – Cross-site scripting, metoda využití neošetřených vstupů k podstrčení vlastního kódu narušujícího chování aplikace

1 Úvod

Dnešní aplikace vyžadují stále větší personalizaci obsahu, výsledků vyhledávání a v neposlední řadě také reklam určených pro konkrétního uživatele. K tomuto účelu se, mimo jiné, využívá i geografická poloha, na které se zařízení nachází. Polohu lze zjistit několika možnými způsoby, které se liší svou přesností a možnostmi použití.

Nejpřesnější metoda zjištění polohy je pomocí GPS. Ta však nemusí být přesná na všech místech, a zejména v budovách ztrácí na efektivitě. Další možností je zjištění polohy pomocí přístupových bodů WiFi, k čemuž se využívá WiFi databázi. S jejich pomocí lze na základě přístupových bodů v okolí zařízení a síly jejich signálu určit přibližnou polohu. U této metody se přímo nabízí její nevýhoda, a to klesající možnosti v místech s nízkou hustotou obyvatel a tím pádem i nižší hustotou přístupových bodů. Poslední, ale relativně nepřesná metoda, je zjištění polohy pomocí bodů připojení k telefonnímu signálu, takzvaných BTS. Pro přesnější zjištění polohy se metody často kombinují [1].

2 Cíl práce

Cílem práce je vytvořit webovou aplikaci, využívající různé zdroje dat k naplnění vlastní databáze WiFi sítí. Aplikace bude rozdělena na 2 části - část, sloužící k získávání dat z vybraných zdrojů a část prezentující získaná data, která bude z velké části tvořena mapou. Aplikace bude umožňovat ovlivnit zobrazená data pomocí různých zobrazovacích módů.

Uživatel bude přímo pracovat pouze s vizuální částí aplikace, ale v uživatelském prostředí bude mít možnost vytvořit požadavek na získání dat pro ohraničenou oblast, kterou určí v mapě. Část získání dat bude pracovat automaticky pomocí opakovaně spouštěných skriptů.

Posledním cílem práce je porovnání získaných dat ze zdrojů, se známými daty, zejména pak porovnání přesnosti jednotlivých zdrojů.

3 Přehled WiFi databází

Existuje spousta WiFi databází, některé jsou určeny k výzkumným nebo statistickým účelům, jiné slouží uživatelům například k nalezení volného přístupového bodu k připojení k internetu.

3.1 Účel WiFi databází

WiFi databáze jsou velmi užitečný zdroj informací, ovšem je vysoce žádoucí tato data aktualizovat. Mají široké možnosti využití, využívají se například pro:

- Určování polohy (nebo vylepšení určení polohy)
- Nalezení zdroje rušení (analýza vytížení pásem)
- Nalezení volného přístupového bodu
- Analýza využití pásem, využití typů zabezpečení a další analýzy

3.1.1 Určování polohy

Informace z WiFi databází se využívají k lokalizaci osob nebo zařízení. Pomocí seznamu přístupových bodů nacházejících se v okolí, a síly jejich signálu, lze přibližně určit polohu, kde se dané zařízení nachází. Tyto informace jsou tím přesnější, čím více sítí se v dané oblasti nachází. V málo osídlených oblastech tato data mohou být velmi nepřesná nebo dokonce žádná. Proto je tato metoda vhodná pouze k zjištění přibližné polohy.

Pokud je potřeba zjistit přesnou polohu, je vhodnější použít GPS senzor. Často se tyto metody kombinují, kdy se údaje o poloze získané z polohy WiFi sítí využívají ke zrychlení funkčnosti GPS senzoru nebo jeho zpřesnění, pokud se nedaří získat z GPS informace o poloze s dostatečnou přesností. To se stává například v husté zástavbě, kde není přesnost GPS tak velká.

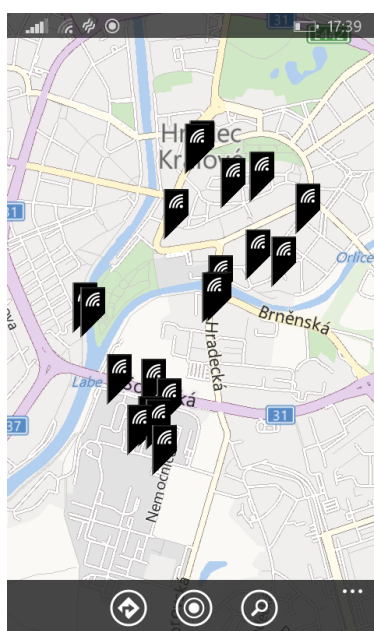
3.1.2 Nalezení zdroje rušení

Pomocí WiFi databází lze určit, které WiFi sítě se nachází v určité oblasti a díky dalším informacím z těchto databází je lze využít jako zdroj dat při hledání zdroje rušení. U toho se využívají zejména informace o používaném kanálu,

a pokud problém s rušením vzniká v této oblasti, lze pomocí informací z WiFi databází překonfigurovat rušený zdroj za účelem odstranění rušení.

3.1.3 Hledání volných sítí

WiFi databáze často obsahují i informace o zabezpečení sítí. Proto je lze využít k nalezení volného přístupového bodu k internetu. Některé mapové služby mají funkci k zobrazení volných sítí v okolí přímo integrovanou ve svých mobilních aplikacích. Tato funkce je dostupná například v mapách Bing v mobilním operačním systému Windows Phone ve verzi 8.1 [2]. Ukázka aplikace je na obrázku č. 1.



Obrázek č. 1 WiFi hotspots v aplikaci Mapy na Windows Phone

3.1.4 Analýzy

V poslední řadě lze informace z WiFi databází využít k různým analýzám. A to buď analýzy nastavení přístupových bodů - jako například používaný kanál, používaný typ zabezpečení, nejpoužívanější SSID a další, nebo analýzy pokrytí území nebo síly signálu.

3.2 Vytváření WiFi databází

K tomu, aby mohly být WiFi databáze využity, je nutné je nejprve naplnit daty. Informace o poloze WiFi sítí jsou získávány několika metodami a získává, ukládá a následně používá je několik společností. Tyto informace mohou být použity pro

zpřesnění, při použití GPS, nebo pro určení přibližné pozice osoby podle toho, které WiFi se v její dostupnosti nachází.

Vlastní databáze WiFi sítí a informací o nich používají a získávají, mimo jiné menší společnosti i velké IT společnosti jako jsou Google, Apple a Microsoft pomocí uživatelů, kteří používají jejich operační systémy.

3.2.1 Metody získávání WiFi sítí

Existují 3 základní způsoby získávání informací o WiFi sítích: ruční zadávání, získávání pomocí dopravního prostředku a získávání s pomocí široké veřejnosti (crowdsourcing).

3.2.1.1 Ruční zadávání

Ruční zadávání je poměrně neefektivní metoda založená na tvorbě databáze lidmi, kteří ručně naplní databázi. Její výhodou je vysoká přesnost informací, ale velkou nevýhodou je vysoká pracnost a nízká efektivita získávání dat.

3.2.1.2 Dopravní prostředek, nejčastěji auto, upravené k tomuto účelu

Styl získávání dat o WiFi sítích pomocí dopravního prostředku se podobá způsobu získávání fotografií ulic pro funkci street view na mapách Google. Místo fotografií se zachytávají nalezené WiFi sítě. Tuto metodu získávání dat používají i menší společnosti, například Wifileaks. Dříve metodu využíval i Google, kdy při vytváření snímků pro street view odchytil i WiFi sítě v okolí, nyní už tuto metodu Google nepoužívá [3]. Výhodou je zvýšení efektivity oproti ručnímu zadávání, nevýhodou je snížení přesnosti.

3.2.1.3 Získávání pomocí veřejnosti a jejich chytrých telefonů

Tato metoda je založena na získávání dat od uživatelů chytrých telefonů, které mají operační systém určité společnosti. To znamená, že tuto metodu používají zejména Google v operačním systému Android, Apple v IOS a Microsoft ve Windows resp. Windows Phone.

Funkce určená k lokalizaci chytrého telefonu pravidelně zjišťuje polohu zařízení s použitím GPS, informací od BTS a WiFi. Při odeslání těchto informací operační systém zároveň odešle seznam nalezených WiFi sítí. Tím mají společnosti

obě potřebné informace (polohu uživatele a seznam WiFi sítí) a sestavují z nich své databáze. K tomuto procesu není nutné zapínat žádnou další aplikaci, operační systém to provede sám. Společnosti ale veřejnost ujišťují, že získaná data jsou zcela anonymní [3].

Výhodou je vysoká efektivita a obrovské množství sítí, které lze získat. Nevýhodou je nutnost uživatele používat konkrétní operační systém, protože systémy nejsou navzájem propojeny.

3.3 Použité databáze

Existuje řada WiFi databází, některé z nich jsou přístupné přímo (například Wifileaks), jiné jsou přístupné s omezeními (Wigle) a některé jsou přístupné nepřímo pomocí lokalizačních služeb (Google).

3.3.1 Wifileaks

Wifileaks je česká služba, jejíž databáze se neustále rozšiřuje. Za touto službou stojí Jakub Čížek, známý například z českého IT portálu Živě.cz. Wifileaks má databázi čítající přes dva miliony WiFi sítí z celé České Republiky. Získávat data pomáhá komunita pomocí mobilní aplikace pro operační systémy Android a iOS [4].

3.3.2 Wigle

Wigle je celosvětová služba, která poskytuje mnohem více dat než Wifileaks, ale získávání dat je omezené. K získání dat musí být uživatel zaregistrován a přihlášen na webu. Zároveň musí zadat parametry, podle kterých chce vyhledávat. Wigle umožňuje získat hrubá data, což jsou vypočtené polohy sítí, nebo i jednotlivé naměřené hodnoty pro určitou MAC adresu. Bohužel získávání dat z Wigle je velmi omezené a z toho důvodu je proces získání dat z Wigle zdlouhavý.

3.3.3 Google

Google neposkytuje službu, která by přímo umožňovala získat data z jeho WiFi databází. Lze však využít jeho lokalizační služby k upřesňování polohy sítí, o kterých již informace v databázi aplikace jsou.

Z tohoto důvodu neumožňuje aplikace získat data přímo z Google na základě rozsahu zeměpisných souřadnic, ale pouze na základě již existujících přístupových bodů. Funkcionalita získání dat z Google na základě rozsahu zeměpisných souřadnic tedy funguje tak, že se nejprve získají data z Wagle a poté se tato data rozšíří o údaje z lokalizačních služeb Google.

4 Analýza

Aplikace bude založena na datech získaných ze zdrojů Wifileaks, Wigle a Google. Je tedy nutné, aby aplikace dokázala data z těchto zdrojů získat a zpracovat. Zpracovaná data poté budou využívána v prezentační části aplikace.

4.1 Požadovaná funkcionalita

Aplikace bude mít dvě základní funkce: získávání dat a vizualizace dat. Získávání dat je velmi důležitá součást aplikace, která bude z velké části probíhat automaticky na pozadí aplikace. Vizualizace dat je část aplikace, s kterou přímo pracuje uživatel. Mezi další funkcionality patří export vyfiltrovaných dat do souboru ve formátu CSV, zobrazení detailů pro každý bod v mapě, nastavení různých módů zobrazení, nebo například zobrazení statistických údajů. Ve statistických údajích bude celkový počet bodů v databázi, používané typy zabezpečení a rozdělení podle zdrojů.

4.1.1 Získávání dat

Do získávání dat bude uživatel zasahovat pouze možností určit, která data má aplikace získat. Zpracování požadavků a následující získání dat již proběhne automaticky, na pozadí aplikace, bez zásahů uživatele.

Data budou získávána ze zdrojů Wifileaks, Wigle a Google. Zdroje se od sebe liší způsobem práce s nimi, daty, která poskytují a omezeními.

4.1.1.1 Wifileaks

Získání dat z Wifileaks proběhne jednorázově, uživatel nemá možnost požádat o data ze zdroje Wifileaks. Služba neposkytuje API, proto je možné získat pouze všechna data najednou ve formě TSV souboru umístěného na subdoméně webových stránek služby Wifileaks. V TSV souboru jsou záznamy oddělené řádkováním, na každém řádku je jeden záznam, konkrétní hodnoty jsou oddělovány tabulátorem, který bývá označován jako `\t`.

Poskytovaná data z Wifileaks jsou MAC adresa, SSID sítě, typ zabezpečení vyjádřený číselnou hodnotou v intervalu od 0 do 4, zeměpisná šířka, zeměpisná délka, nadmořská výška a datum poslední aktualizace.

Každá MAC adresa se může v datech vyskytovat vícekrát, například pokud jedna MAC adresa odpovídá více SSID. To se stává v případě, kdy byl signál ze stejného zařízení naměřen různě v čase a mezi tím se změnilo SSID vysílané zařízením.

Poloha u jednotlivých bodů bude s velkou pravděpodobností vypočítávána. Způsob výpočtu není znám. Z 83 006 981 měření je v databázi k 17. březnu 2016 celkem 2 092 208 bodů [4].

Protože data budou získána jednorázově, případně opakovaně, ale ne s častou frekvencí, omezení u tohoto zdroje nejsou důležitá.

4.1.1.2 Wigle

Zdroj Wigle je oproti Wifileaks mnohem složitější. Poskytuje API, díky kterému je možné získávat požadovaná data určená parametry. Uživatel bude mít možnost vytvořit požadavek na získání dat z Wigle.

Wigle API umožňuje získat hromadně seznam přístupových bodů, u kterých jsou zeměpisné souřadnice vypočtené ze všech měření tohoto bodu, ale umožňuje i získat všechna tato měření pro určitou MAC adresu. Aplikace bude používat obě tyto funkce. Nejprve získá hrubá data, díky kterým získá seznam MAC adres na daném území, a poté pro všechny tyto MAC adresy získá i konkrétní měření.

Informace, které Wigle poskytuje k jednotlivým bodům, jsou poloha, vypočtená jako vážený průměr poloh všech měření, kde vahami jsou síly signálu umocněné na druhou [5]. Další poskytované informace jsou použitý kanál, nastavení funkce QoS, typ zabezpečení, MAC adresa, SSID sítě, uživatel, který bod naměřil, datum a čas prvního a posledního měření, datum a čas poslední úpravy, typ sítě, použití DHCP, informace o tom jestli je síť volná nebo placená a beacon interval. U jednotlivých měření pro danou MAC adresu je vypočtená poloha nahrazená naměřenou polohou a jsou přidány další informace, konkrétně síla signálu, nadmořská výška, přesnost měření a informace o rušení. Získaných dat je mnohem více než ze zdroje Wifileaks, ale ne všechna data jsou podstatná pro použití v této aplikaci, proto jsou použity a uloženy pouze některé hodnoty. Typ zabezpečení je určen několika hodnotami, proto je nutné sjednotit typ zabezpečení na jeden formát, odpovídající všem zdrojům.

Wigle sice poskytuje více dat než Wifileaks, ale bohužel má velmi přísná omezení. Pro jakoukoliv komunikaci s Wigle API je nutné mít vytvořen uživatelský účet a být přihlášen. Další omezení je možnost získat jedním požadavkem informace pouze o 100 přístupových bodech s vypočtenou polohou. Konkrétní naměřené hodnoty je možné získat všechny, ale pouze na základě MAC adresy, takže jedním dotazem pouze pro jeden bod. Poslední omezení je počet požadavků, kterých může za jeden den být pouze 50.

Kvůli těmto omezením je nutné požadavky pro získání dat, vytvořené pro zdroj Wigle, předzpracovat a minimalizovat jejich počet. Proto budou požadavky od uživatelů rozděleny na několik menších, na základě analýzy překryvné vrstvy, kterou Wigle generuje do své mapy a také od sebe budou požadavky navzájem odečítány, aby nebylo v jednu dobu aktivních několik požadavků pro Wigle na stejné území. Pokud tedy již existující požadavky pro získání dat z Wigle budou zasahovat do nově vytvářeného požadavku, tak bude nový požadavek rozdělen tak, aby se tyto části nepřidávaly do fronty znovu.

4.1.1.3 Google

Pro získání dat z Google budou využity lokalizační služby Google, které slouží k určení polohy uživatele na základě přístupových bodů nacházejících se v jeho okolí. Uživatel bude mít možnost vytvořit požadavek pro zdroj Google, ovšem funkcionality bude rozdělena na 2 části: nejprve se vytvoří stejný požadavek pro zdroj Wigle, a až po dokončení získávání všech dat z Wigle týkajících se tohoto požadavku, započne získávání dat z Google. Z toho důvodu je nutné implementovat funkci blokování požadavků, která bude vysvětlena v části zabývající se implementací aplikace.

Lokalizační služby Google vyžadují předání alespoň dvou přístupových bodů. Podle jejich umístění a síly signálu poté vrátí odhadovanou polohu. Získání dat z Google tedy proběhne na základě již existujícího bodu v databázi, k němuž se nalezne druhý bod. Bodům se vygeneruje fiktivní síla signálu takovým způsobem, že bod, jehož informace potřebujeme získat, bude simulován jako bod s velkou silou signálu a druhý bod je simulován jako bod s velmi slabým signálem. Tím se docílí co nejmenšího zkreslení vrácené polohy z Google.

Z lokalizačních služeb Google je vrácena pouze poloha a její přesnost. Z toho důvodu je bod získaný z Google uložen s ostatními informacemi zděděnými z již existujícího bodu získaného ze zdroje Wigle, o kterém byla poloha zjišťována.

4.1.1.4 Shrnutí zdrojů dat

Zdroje se od sebe liší v omezeních, přesnosti dat a poskytovaných datech. Porovnání těchto údajů z používaných zdrojů je znázorněno v tabulce č. 1.

Zdroj dat	Wifileaks	Wigle	Google
Získaná data	MAC adresa SSID Typ zabezpečení Zeměpisná šířka Zeměpisná délka Nadmořská výška Datum poslední aktualizace	MAC adresa SSID Typ zabezpečení Zeměpisná šířka a délka Nadmořská výška Přesnost měření Použitý kanál Datum prvního měření Placená/volná síť Beacon Interval	Zeměpisná šířka Zeměpisná délka Přesnost informací
Lze získat konkrétní měření	NE	ANO	NE
Formát dat	TSV soubor	JSON	JSON
Omezení	Nelze požádat pouze o jeden bod	Maximálně 50 API požadavků denně Na jeden požadavek určený oblastí vrátí maximálně 100 sítí Nutnost získávat nejdříve vypočtené polohy a až poté po jedné získávat detaily Nutné vytvoření uživatelského účtu	Nutnost předat v dotazu minimálně 2 MAC adresy (vychází z použití lokalizačních služeb)
Pracnost získání dat	Nízká	Vysoká	Střední

Tabulka č. 1 Porovnání zdrojů dat

Jak lze vidět z tabulky, nejjednodušší je získat informace z Wifileaks. Zároveň ale jako jediný zdroj předává Wifileaks data v jiném formátu než JSON a nelze požádat pouze o jeden bod ani získat všechna konkrétní měření. Omezení u zdroje Wifileaks nejsou podstatná, protože data budou získána jednou.

Získání dat z Google je složitější, využívá se totiž lokalizačních služeb Google. Princip využití lokalizačních služeb byl popsán v části popisu zdroje Google. Data z lokalizačních služeb jsou získána ve formátu JSON a počet požadavků na server pravděpodobně není omezen.

Wigle jako jediný zdroj umožňuje získání i konkrétních měření pro každou MAC adresu. Pracnost získání dat je ale vysoká. Data z Wigle jsou získána ve formátu JSON a o jednom bodu lze získat nejvíce informací právě z Wigle.

4.1.2 Vizualizace dat

S částí, která se týká vizualizace dat, pracuje přímo uživatel aplikace. Grafické rozhraní bude tvořeno převážně mapou a panelem s ovládacími prvky.

Mapa bude umožňovat několik způsobů zobrazení. Konkrétně zobrazení všech bodů, zobrazení bodů podle nastaveného filtru, označení bodu podle parametrů, zobrazení pouze jedné sítě a zobrazení vypočtené polohy bodu.

Panel ovládacích prvků bude tvořen vyhledávacím formulářem, tlačítky sloužícími k iniciování požadavku na získání dat a položkami menu. Uživatel bude mít možnost zobrazit si bližší informace o každém bodu v mapě. Toho dosáhne kliknutím na vybraný bod, na nějž aplikace zareaguje zobrazením okna s detailními informacemi.

4.1.2.1 Módy zobrazení

1. **Zobrazení všech bodů** – Mód zobrazení všech bodů je výchozí, v tomto módu budou zobrazeny všechny body odlišené barevnými odstíny podle zdroje dat, z kterého pochází.
2. **Zobrazení vyfiltrovaných bodů** – Zobrazení vyfiltrovaných bodů se nebude nijak lišit od módu zobrazení všech bodů. Jediný rozdíl je v tom, které body se budou zobrazovat. Filtrovat bude možné podle SSID, MAC adresy, používaného kanálu, používaného typu zabezpečení a podle zdroje, z kterého byly body

získány. Filtry bude možné kombinovat, například: všechny sítě ze zdroje Wigle, se zabezpečením WPA2, používající kanál 9.

3. **Označení bodů podle parametrů** – V zobrazení typu označení bodů podle parametrů budou zobrazeny všechny body stejně jako v módu zobrazení všech bodů. Přičemž body, odpovídající vybranému typu označení, budou zobrazeny jinou barvou než všechny ostatní body. Označit body bude možné podle SSID, MAC adresy nebo kanálu.
4. **Zobrazení pouze jedné sítě** – V módu zobrazení pouze jedné sítě se budou zobrazovat pouze body s určeným SSID. Body od sebe nebudou nijak barevně odlišené, ale budou větší. Toto zobrazení bude simulovat pokrytí oblasti pro danou konkrétní síť.
5. **Zobrazení vypočtené polohy** – V zobrazení vypočtené polohy se budou body zobrazovat podobným způsobem jako v módu zobrazení pouze jedné sítě. Tentokrát však budou body určeny MAC adresou, a navíc se k nim, z poloh měření bodů, které mají stejnou MAC adresu a leží v okolí, vypočte poloha, na kterou se odlišnou barvou zobrazí bod. Pomocí tohoto zobrazení bude možné přesněji určit polohu konkrétního přístupového bodu.

V uživatelském rozhraní nesmí chybět tlačítko na zrušení všech nastavených filtrů, čímž dojde k odebrání všech podmínek a nastavení módu zobrazení všech sítí. Posledním tlačítkem bude možné exportovat vyfiltrované body do souboru CSV.

4.2 Použité technologie

Aplikace bude naprogramována v jazyce PHP s použitím jednoho z frameworků a jako databáze bude použito MySQL. Součástí uživatelského rozhraní bude zejména mapa, z čehož vyplývá, že další potřebná technologie je některé z dostupných API map.

4.2.1 Framework

Framework je část softwaru, jejíž úkolem je usnadnit práci vývojářům a umožnit jim více se soustředit na konkrétní funkčnost softwaru místo programování obecných funkcí jako je například přihlašování uživatelů nebo základní ošetřování vstupů z uživatelského rozhraní.

Velkou výhodou frameworků je implementace MVC architektury, kterou nemusí programátoři programovat sami, což výrazně zkrátí dobu potřebnou pro vývoj.

Nevýhodou frameworků bývá jejich velká robustnost a složitost, čímž dochází ke zpomalení aplikace. Další částečná nevýhoda je nutnost programátora naučit se framework používat a nastudovat si jeho dokumentaci. Doba nutná k nastudování frameworku se v budoucnu vrátí, pokud bude stejný framework používán u více projektů.

Vzhledem k výpočetní náročnosti některých frameworků se jejich použití nehodí pro každou aplikaci. Zejména u aplikací, kde se klade nejvyšší důraz na co nejvyšší odezvu, nemusí být použití frameworku vhodné.

4.2.2 PHP frameworky

Frameworků určených pro programovací jazyk PHP je celá řada, z nichž většina je licencována jako open source. Příklady PHP frameworků jsou: Symfony, CakePHP, Kohana, Yii, Laravel, Zend nebo český Nette Framework.

4.2.2.1 Symfony

Framework Symfony je open-source framework vycházející z MVC vzoru. Jeho filozofií je podpora profesionality, osvědčených postupů a standardizace psaní aplikací. Symfony je produktem firmy Sensio Labs sídlící v Paříži [6].

4.2.2.2 CakePHP

Open-source framework CakePHP vznikl v roce 2005. Inspirací pro vývojáře byl framework Ruby on Rails. CakePHP je založený na architektonickém vzoru MVC. Zaměřuje se zejména na přehlednost a krátký zápis kódu [7].

4.2.2.3 Zend Framework

Zend framework se řadí mezi nejznámější PHP frameworky. Je licencovaný jako open-source a vyvíjí se od roku 2005. Licencován je pod New BSD licenci. Hlavní sponzor projektu je společnost Zend Technologies, ale dohromady se na vývoji a jeho podpoře podílelo více společností, mimo jiné například i Google a Microsoft [8].

Zend framework je založený na použití modulů, což umožňuje programátorovi využívat pouze ty části Zend frameworku, které potřebuje. Mezi jednotlivými moduly však mohou být částečné závislosti. Zend je navržen tak, aby si ho programátoři mohli sami rozšiřovat podle potřeby [7].

4.2.2.4 Nette Framework

Nette Framework je český PHP framework určený pro tvorbu webových aplikací v jazyce PHP 5. Jeho autorem je David Grudl, který k Nette frameworku pořádá školení. Framework Nette je open-source framework s MVC architekturou dále rozvíjený organizací Nette Foundation. V České Republice má Nette framework širokou podporu programátorů a je velmi oblíbený.

Nette se skládá z modulů a využívá událostmi řízené programování, má vlastní šablonovací systém nazývaný Latte, soustředí se na ošetření proti útokům typu XSS a SQL injection a obsahuje silné nástroje pro ladění chyb [9].

Z důvodů poskytovaných funkcí, srozumitelnosti dokumentace a velké podpoře českých vývojářů byl vybrán k tvorbě této aplikace framework Nette, který bude blíže popsán.

4.2.3 Funkce a principy Nette Frameworku

Nette umožňuje programátorům použít pouze některé jeho moduly a nenutí je používat framework jako celek. To znamená, že pokud programátor chce použít pouze modul pro formuláře, může využít modul Nette/Forms samostatně [10].

Nette funguje na principu architektury MVC (MVP), to znamená, že se aplikace dělí na 3 základní součásti [11].

1. **Modely (model)** – Model obsahuje doménové třídy pro jednotlivé entity, ale také aplikační funkce pro výpočty nebo pro práci s databází. Model aplikace by měl být znovupoužitelný a nezávislý na uživatelském rozhraní.
2. **Pohled, šablona (view)** – Šablona uživatelského rozhraní, určující vzhled aplikace. Využívají se šablonovací jazyky, u Nette se jedná o jazyk Latte.
3. **Řízení (controller nebo presenter)** – Controller je část aplikace komunikující s uživatelem pomocí view. Uživatel si pomocí uživatelského rozhraní vyžádá nějakou operaci a předá parametry, které jsou pro vykonání potřebné. Presenter rozhodne o akci, vybere model, pomocí kterého akci provede, získá data od modelu a předá je zpět šabloně.

Další důležitou součástí frameworku Nette je tzv. Routování. Routování je vlastně proces mapování URL adres na konkrétní presenter, který událost obsluží.

4.2.3.1 Životní cyklus Nette aplikace

Životní cyklus aplikace je velmi podobný u všech aplikací implementujících MVC architekturu.

1. Nejdříve uživatel aplikace vytvoří požadavek, který je určený URL adresou.
2. Požadavek přijde k routeru, který zná mapování URL adres na presentery, takže vybere presenter přiřazený tomuto požadavku a předá mu řízení.
3. Presenter rozhodne, která akce se má provést, získá potřebný model a zavolá na něm požadovanou událost.
4. Model provede akci, například získá data z databáze a vrátí je presenteru.
5. Presenter určí šablonu a předá jí data.
6. Šablona zpracuje data (sestaví z nich výsledný HTML kód), a vrátí ho presenteru.
7. Presenter přijme HTML kód a vrátí ho uživateli.

4.2.4 API map

Uživatelské rozhraní bude tvořeno i mapou, v které jsou polohy bodů zobrazeny, z toho důvodu je potřebné využít i některé API map.

4.2.4.1 Funkce API map

Všechna API map mají podobné funkce. Základní funkcí je zobrazení mapy, které lze nastavit takovým způsobem, aby se například přiblížilo konkrétní město. Další užitečnou funkcí je možnost zobrazení satelitních nebo leteckých snímků.

Do mapy lze přidat značky sloužící k označení zajímavých míst nebo zobrazit naplánovanou trasu z místa A na místo B. Další velmi užitečné využití map je vizualizace dat, což je předmětem této práce.

4.2.4.2 Existující API map

K použití map existují propracovaná API a obsáhlé dokumentace o jejich použití. Po celém světě známé je Google Maps API. Mapy Google mají obrovské množství možností, mapy a satelitní snímky celého světa, API pro použití na webu, pro mobilní operační systémy Android a iOS a velmi známou funkčnost zvanou StreetView, která umožňuje prohlížení ulic z lidského pohledu.

V České Republice velmi známé Mapy.cz, vlastněné společností Seznam.cz, mají také své vlastní API. To má sice menší množství možností než API od Google, ale i tak jsou funkce dostačující. Od 27. 5. 2015 obsahují Mapy.cz mapy celého světa, letecké snímky jsou však kvalitní pouze v České Republice, kde je v některých městech možnost přepnout i na pohled z ptačí perspektivy. Nově Seznam zavádí 3D mapy a rozšiřuje pokrytí svou obdobou funkce street view, zvanou Panorama.

Poslední vybrané mapy jsou mapy Bing. Bing má mapy celého světa, letecké podklady jsou však v České Republice velmi špatné. Výhodou je kvalitní pohled z ptačí perspektivy, který je dostupný pouze ve vybraných městech po celém světě (zejména v USA). Mapy Bing mají API pro Windows Phone, webové stránky a desktopové aplikace pro operační systém Windows.

Vzhledem k účelu této aplikace a počtu možností a použitelnosti map bylo pro tvorbu této aplikace vybráno API Google Maps, jehož některé funkce budou blíže popsány dále v textu.

4.2.5 Google Maps API

Mapy Google, patřící společnosti Google, lze pro nekomerční použití využívat zdarma. Nabízí mapy ulic celého světa, kvalitní satelitní snímky a funkci street view. K jejich API je velmi propracovaná dokumentace dostupná na internetu.

Použití Google map je velmi jednoduché. Pokud chceme používat pokročilejší funkce, jako například sledování statistik používání, můžeme na webu získat takzvané API KEY. Celé API se poté načte jako jeden skript, kterému se dané API_KEY předá. Do určitého počtu zobrazení je použití API_KEY nepovinné. Ukázka skriptu pro načtení API Google Maps je na obrázku č. 2.

```
1. <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=API_KEY"></script>
```

Obrázek č. 2 Kód pro načtení API Google Maps

Poté stačí na stránce vytvořit blok, nastylovat ho podle potřeby a na stránku přidat skript, který lze nalézt v dokumentaci. Jak je vidět na obrázku č. 3 postup je jednoduchý.

1. Vytvoření nastavení mapy – zde se nastaví například míra přiblížení a bod, na který bude mapa vycentrována.
2. Pomocí funkce Google Maps API je vytvořena mapa, jako první parametr je předán dříve vytvořený blok, který bude obsahovat mapu, a jako druhý parametr je předáno nastavení mapy.
3. Nastavení akce, která se má stát při události. V tomto případě se při načtení okna spustí funkce initialize, která obsahuje nastavení a vytvoření mapy.

```
1. <script type="text/javascript">
2.     function initialize() {
3.         var mapOptions = {
4.             center: { lat: -34.397, lng: 150.644 },
5.             zoom: 8
6.         };
7.         var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
8.     }
9.
10.     google.maps.event.addDomListener(window, 'load', initialize);
11. </script>
```

Obrázek č. 3 Kód pro nastavení a zobrazení mapy

Tyto 3 jednoduché kroky stačí k zobrazení mapy na webu, její vycentrování a přiblížení podle předaného nastavení.

4.2.5.1 Google Maps API – Markers

Markers neboli značky jsou základními prvky, které lze do mapy přidat. Slouží k označení bodu na konkrétní pozici určené zeměpisnými souřadnicemi. Google Maps API umožňuje značkám přidat animace a změnit jejich vzhled. Značky se často kombinují s funkcí Info Windows takovým způsobem, že po kliknutí na značku se otevře informační okno s podrobnějšími informacemi o daném bodu [12].

Značky jsou užitečná funkcionality, ale bohužel jsou velmi náročné na výkon a proto se nehodí k použití v této aplikaci. Značek by na mapě bylo více než milion. Z toho důvodu budou body zobrazeny pomocí překryvné vrstvy, která je vysvětlena dále v textu.

4.2.5.2 Google Maps API – Info Windows

Informační okna slouží k zobrazení podrobnějších informací. Objevují se v kombinaci se značkami, kde značka určuje pozici na mapě a info okno obsahuje ostatní podrobnější informace. Obsah info okna může být strukturován pomocí HTML entit [13].

4.2.5.3 Google Maps API – Události

Google Maps API podporuje odchyťování událostí a umožňuje programátorovi na ně reagovat. Toho lze využít právě při kombinaci značky a info okna. Značce je přidán event listener¹ na událost kliknutí a jako reakce bude otevření info okna.

Další možností využití je načítání podrobnějších informací do mapy až při určitém přiblížení. Z mapy lze pomocí funkce `getBounds()` získat viditelné okraje a díky tomu je možné například odstranit body, které nejsou vidět při daném přiblížení. Často se načítají i body ležící mimo viditelnou oblast za účelem eliminace viditelnosti načítání dalších bodů, které může nějakou dobu trvat.

Základní události, na které lze reagovat, jsou například události při posouvání mapy myší, pohybu kurzorem přes mapu, změně viditelných okrajů mapy a další. Události jsou popsány v dokumentaci Google Maps [14].

¹ Funkce, která je vykonána při určité události

4.2.5.4 Google Maps API – překryvná vrstva

Mapy Google umožňují překreslit mapu vlastní vrstvou, což umožňuje zobrazit uživateli nějaká data navíc, například poukázat na body zájmu. Tato funkce umožňuje zobrazit cokoliv, co je do překryvných dlaždic vygenerováno. Funkcionalita je založená na principu tvorby částečně průhledných dlaždic, na kterých se v požadovaných místech nakreslí požadovaný objekt.

O umístění dlaždic na správné místo v mapě i o vytvoření požadavků na server se stará samotné Google Maps API. Nutné je pouze nastavit, že se má vrstva vykreslovat, a implementovat metodu pro získání konkrétní dlaždice [15].

4.2.6 Další použité technologie

Při implementaci aplikace budou použity i další technologie, konkrétně databáze MySQL. Pracovat se bude také pomocí AJAXových požadavků s využitím javascriptové knihovny JQuery. Dále také v procesu získávání dat, ale i při implementaci uživatelského rozhraní bude používán formát JSON.

4.2.6.1 MySQL databáze

MySQL je světově nejoblíbenější open-source databáze vlastněná společností Oracle [16]. Je licencovaná pod GPL licenci a podporovaná komunitou vývojářů. MySQL databázi používá mimo jiné například sociální síť Facebook [17].

4.2.6.2 JQuery a AJAX

„JQuery je knihovna s otevřeným zdrojovým kódem určená pro jazyk JavaScript, která zjednodušuje interakci mezi dokumentem HTML, přesněji řečeno objektovým modelem dokumentu (model DOM), a jazykem JavaScript“ [18].

JQuery je neustále vyvíjeno, existuje k němu mnoho rozšíření a vzhledem k počtu funkcí je JQuery používáno na velkém množství webových stránek.

Mimo práci s HTML dokumentem umožňuje JQuery například zpracování událostí, animace v uživatelském rozhraní, nebo například zjednodušuje práci s technologií AJAX.

Pro větší uživatelský zážitek a snížení požadavků na server se využívá technologie AJAX. AJAX je zkratka z „asynchronní JavaScript + XML“. AJAX umožňuje

provést na webu asynchronní akci. Při takové akci není načítána celá stránka znovu, ale pouze některé její části. AJAX nemusí sloužit pouze k načítání obsahu do částí webu, ale i jako možnost vykonání činnosti. Synchronní funkcionalitu založenou na odkázání uživatele na skript, který pouze provede svou činnost a poté přesměruje uživatele zpět na původní stránku, lze nahradit právě pomocí AJAXu a uživatel vůbec nemusí zaznamenat, že se daná činnost provedla.

AJAX je založen na využití objektu XMLHttpRequest (XHR). „Objekt XMLHttpRequest byl původně implementován v Internet Exploreru 5 jakožto komponenta ActiveX“ [19]. Z toho vyplývá, že technologie AJAX není žádnou novinkou. Často je využíván právě v kombinaci s knihovnou JQuery, která velmi usnadňuje implementaci technologie AJAX na webu. Ukázka použití JQuery v kombinaci s technologií AJAX je na obrázku č. 4.

```
1. $.ajax("download/time2down",{
2.     data: {
3.         lat1: userEditableRectangle.getBounds().getSouthWest().lat(),
4.         lat2: userEditableRectangle.getBounds().getNorthEast().lat(),
5.         lon1: userEditableRectangle.getBounds().getSouthWest().lng(),
6.         lon2: userEditableRectangle.getBounds().getNorthEast().lng(),
7.         sourceDownloadFrom: $("#createDownloadRequest").attr("data-source"),
8.         filter: getUrlVars(),
9.     }
10. }).done(function(data) {
11.     eval(data);
12. });
```

Obrázek č. 4 AJAX s využitím JQuery

4.2.6.3 Formát JSON

JSON je formát určený pro výměnu dat. „JSON je navržen tak, aby bylo jeho použití člověkem a parsing strojem snadné“ [19]. Formát JSON je oproti XML jednodušší, výsledný soubor obsahuje méně režijního kódu, a proto se JSON používá stále častěji.

Formát JSON lze snadno používat v jazyce JavaScript. „Dvě běžné úlohy při práci s formátem JSON jsou zakódování objektu do textového řetězce a dekodování textového řetězce do literálu objektu“ [18].

Formát JSON lze použít i v PHP, kde k dekodování formátu JSON slouží funkce `json_decode` a k zakódování funkce `json_encode`. „Kromě samotných dat je vhodné

poslat hlavičku informující prohlížeč o typu přenášených dat“ [20]. Ukázka dat ve formátu JSON se nachází na obrázku č. 5.

```
1. {
2.   "accuracy": 33690,
3.   "location": {
4.     "lat": 50.1967098,
5.     "lng": 15.8068965
6.   },
7.   "status": "OK"
8. }
```

Obrázek č. 5 Data ve formátu JSON

4.2.6.4 PHP CURL

K získávání dat je nutné komunikovat s jinými aplikacemi a využívat jejich API. K tomu lze v PHP použít CURL rozšíření. Extenze CURL se umí postarat i o přenos cookies [20]. To je nutné například při vytváření požadavků na Wigle API, k jejichž provedení je nutné, aby uživatel byl přihlášen. Přihlášení je uloženo do souboru cookie, který je využíván u dalších požadavků. Příklad použití PHP CURL je na obrázku č. 6.

```
1. $ch = curl_init();
2. curl_setopt($ch, CURLOPT_URL, "https://wigle.net/api/v1/jsonLocation");
3. curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query(array("netid" => "41:9E:01:52:65:CC")));
4. curl_setopt($ch, CURLOPT_HEADER, true);
5. curl_setopt($ch, CURLOPT_COOKIESESSION, true);
6. curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
7. curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
8. curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
9. curl_setopt($ch, CURLOPT_COOKIEJAR, "cookie.txt");
10. curl_setopt($ch, CURLOPT_COOKIEFILE, "cookie.txt");
11. $result = curl_exec($ch);
12. curl_close($ch);
```

Obrázek č. 6 Příklad použití PHP CURL pro získání měření pro MAC adresu z Wigle

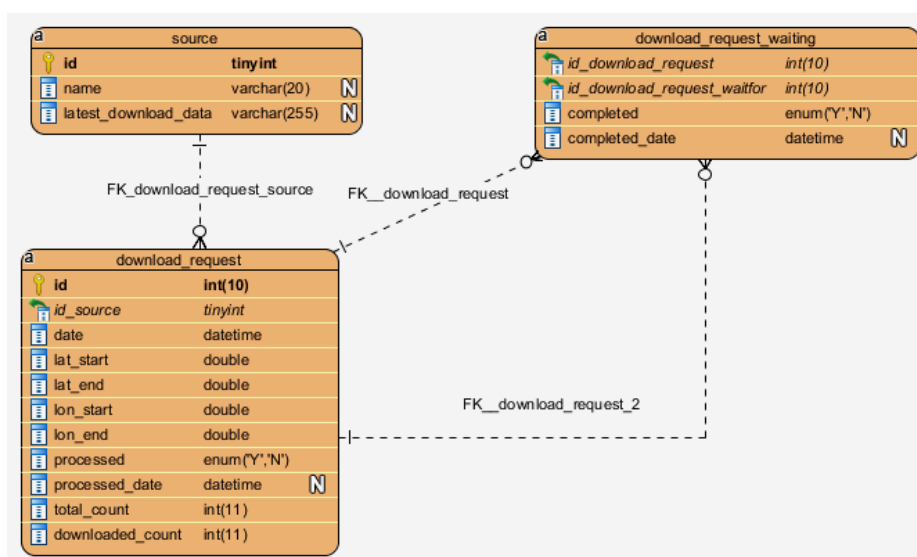
4.3 Databázová vrstva aplikace

Aplikace se skládá ze dvou základních částí - získávání dat a vizualizace. Z toho vyplývá i fakt, že databáze bude obdobně strukturována. Databáze bude rozdělena na 2 části - jedna bude část týkající se získávání dat, to znamená struktury potřebné pro ukládání požadavků uživatelů, k ukládání předzpracovaných dat a další. Druhou částí jsou tabulky využívané k uložení konkrétních bodů, které se budou zobrazovat do mapy, a dalších tabulek sloužících k vedlejšími funkcionalitám, které aplikace bude poskytovat.

4.3.1 Databáze pro získávání dat

V první řadě je nutné ukládat požadavky od uživatelů. Tyto požadavky na sebe také mohou čekat a blokovat se, tím pádem je nutné umožnit požadavky blokovat mezi sebou. U požadavku se odlišuje také zdroj dat, z kterého jsou data požadována.

Vznikají 3 tabulky: tabulka zdrojů dat, požadavků a propojovací tabulka požadavek-požadavek, která slouží pro ukládání blokování požadavků mezi sebou. Tabulka zdrojů bude obsahovat pouze identifikátor, název zdroje a dále také sloupec, který bude sloužit k uložení názvu souboru, který byl jako poslední zpracován ze zdroje Wifileaks. Tabulka blokování požadavků bude obsahovat ID blokováného požadavku a ID blokujícího požadavku, dále identifikátor označující stav blokování a datum a čas kdy byl blokující požadavek dokončen. Struktury těchto tabulek jsou zobrazeny na obrázku č. 7.



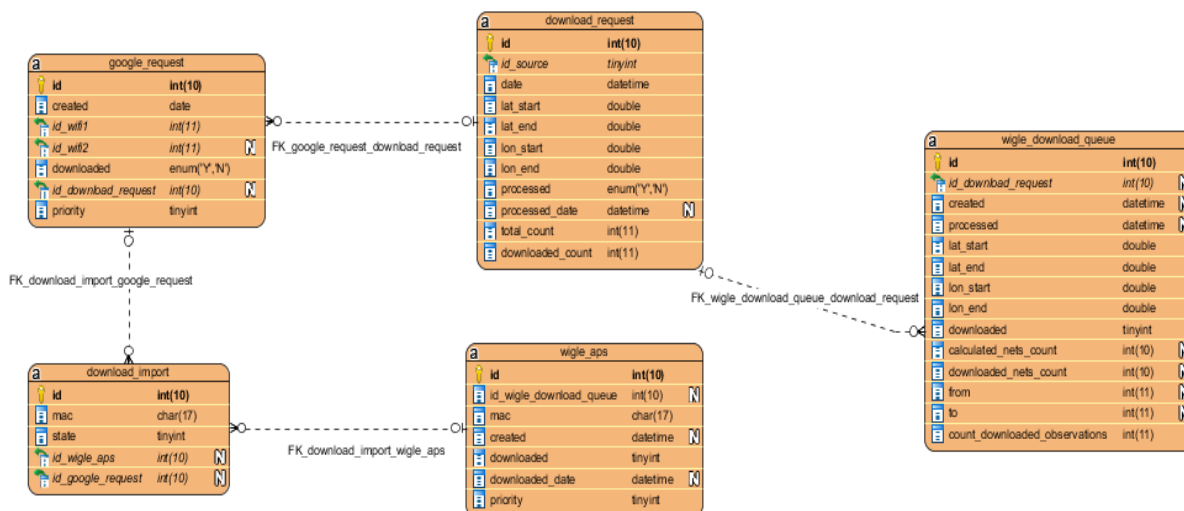
Obrázek č. 7 Databáze - požadavky od uživatelů

V tabulce konkrétních požadavků jsou hodnoty, kterými je požadavek určen. To znamená rozsah souřadnic, zdroj dat, datum vytvoření požadavku a další hodnoty, s kterými se bude pracovat při zpracovávání požadavku. Významy sloupců jsou vysvětlené v tabulce č. 2.

Název sloupce	Význam hodnot
processed	Stav zpracování požadavku.
processed_date	Datum a čas zpracování požadavku.
total_count	Hodnota vyplněná až po zpracování požadavku. Znamená počet záznamů vytvořený ve frontě požadavků konkrétně pro zdroj Wigle.
downloaded_count	Počet dokončených záznamů ve frontě požadavků pro zdroj Wigle, vytvořených z daného požadavku. Pokud tato hodnota je stejná jako hodnota ve sloupci total_count znamená to, že byl celý požadavek již zpracován a data jsou získána v databázi.

Tabulka č. 2 Význam hodnot databázové tabulky požadavků

Při zpracování požadavku vznikají další data, která jsou rozdělena do několika dalších tabulek. Konkrétně do tabulky s frontou požadavků pro zdroj Wigle (tabulka wigle_download_queue), po jejichž zpracování vzniká tabulka MAC adres, jejichž měřená budou získána ze zdroje Wigle (tabulka wigle_aps). Pokud jsou data požadována i ze zdroje Google, vznikají další data, která slouží jako požadavky pro zdroj Google (tabulka google_request). Poslední způsob tvorby požadavků je import, takže je nutné uložit importované MAC adresy (tabulka download_import). Struktura těchto tabulek je ukázána na obrázku č. 8.



Obrázek č. 8 Databáze - rozpracované požadavky

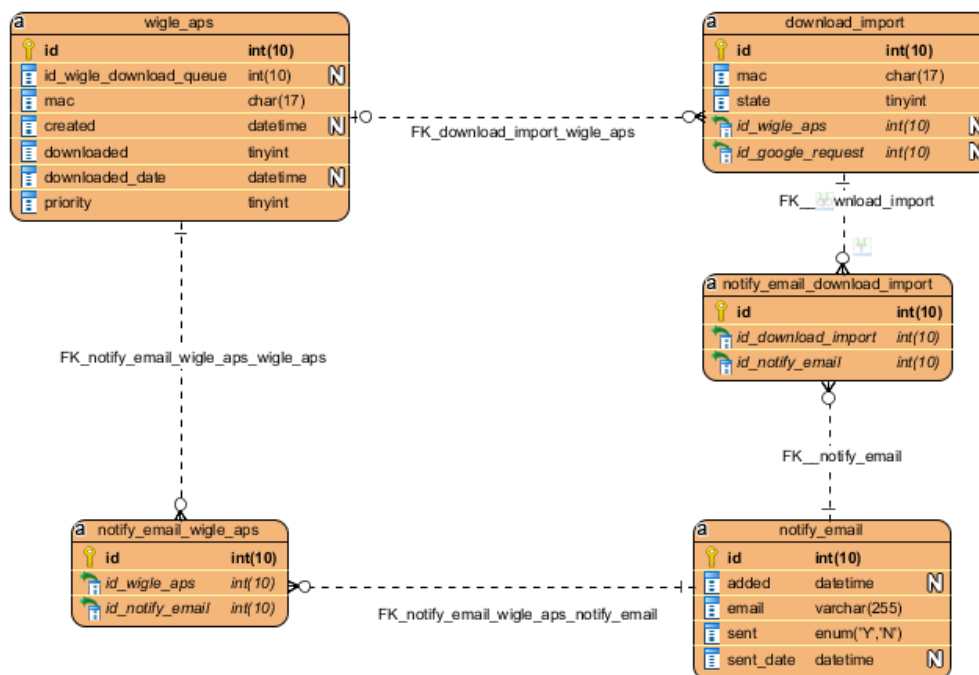
V tabulce wigle_download_queue je rozsah souřadnic, informace o stavu zpracování, požadavek z kterého byl záznam vytvořen a další hodnoty. Hodnoty sloupců from a to jsou hodnoty nutné kvůli omezením Wigle API, které umožňuje získat pouze 100 sítí najednou. Hodnota ve sloupci calculated_nets_count je pouze orientační údaj, značí kolik bylo vypočteno bodů na dlaždici získané z Wigle. Hodnota v sloupci downloaded_nets_count je reálný počet sítí, který z požadavku byl získán. Sloupec count_downloaded_observations je počet již zpracovaných MAC adres, z nichž byly získány i všechna měření. Pokud je hodnota sloupce count_downloaded_observations stejná jako hodnota v sloupci downloaded_nets_count, znamená to, že tento požadavek již byl zpracován.

Tabulka download_import je společná pro zdroj Wigle i Google. Jsou do ní importovány požadované MAC adresy vytvořené požadavkem typu import MAC adres. V tabulce je důležitá hodnota sloupce state, která nabývá několika hodnot, jejichž význam je popsán v tabulce č. 3.

Hodnota	Význam hodnoty
1	Přidáno do fronty Wigle MAC adres (tabulka wigle_aps)
2	Měření byla získána z Wigle
3	MAC adresa byla přidána do fronty pro získání z Google
4	Údaje byly získány z Google

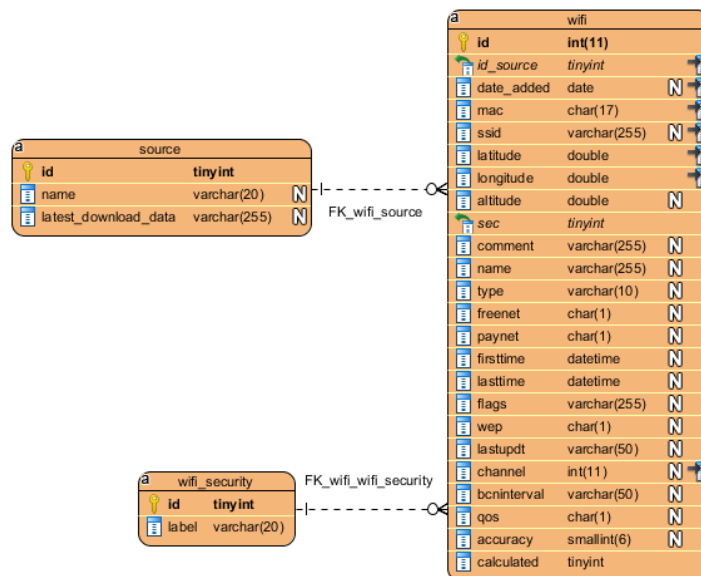
Tabulka č. 3 Význam hodnot stavu importu MAC adres

Další údaje, které se týkají získání dat a vytváření požadavků jsou požadavky na notifikaci po získání dat. Pro ty je vytvořena další datová struktura obsahující email, informaci zda již byl uživatel informován a vazby s požadavky, po jejichž zpracování má být uživatel informován. Struktury jsou popsány na obrázku č. 9.



Obrázek č. 9 Databáze - notifikace po získání dat

Nakonec jsou získané body uloženy do tabulky wifi, která obsahuje všechny informace o přístupových bodech. Tabulka wifi je společná pro všechny zdroje. Z části získávání dat jsou do ní přidávány záznamy, které jsou využívány ve vizualizační části aplikace. Tato tabulka je nejdůležitější, proto bude popsána podrobněji. Struktura tabulky se nachází na obrázku č. 10. Tabulka je propojena se zdrojem dat a tabulkou zabezpečení, což jsou pouze číselníkové tabulky, ve kterých je výčet hodnot, kterých hodnota sloupce nabývá a popis, který se bude zobrazovat v GUI.



Obrázek č. 10 Databáze - tabulka s přístupovými body

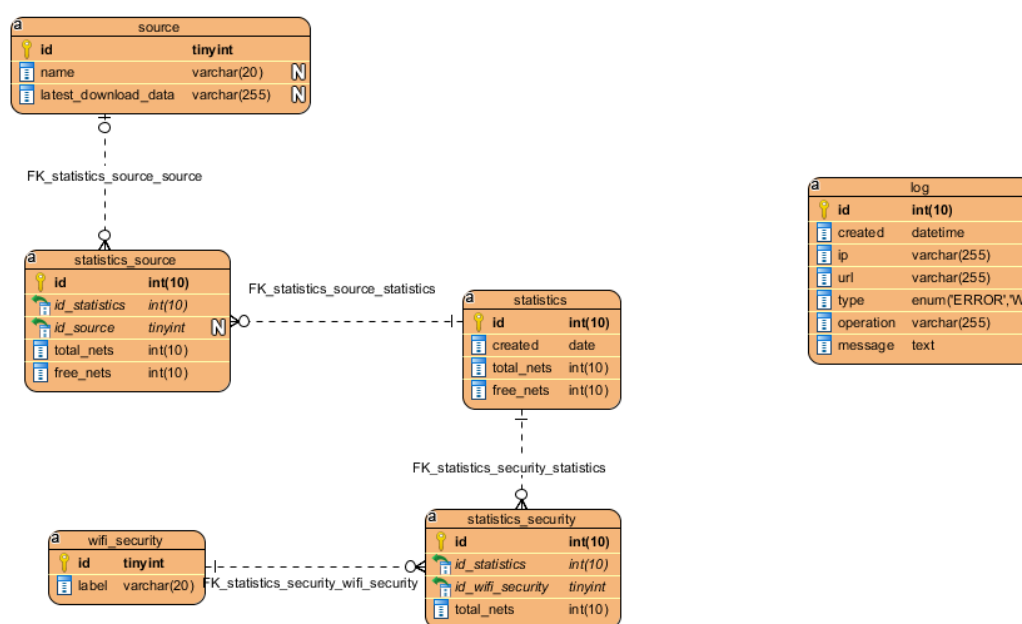
Význam některých hodnot jednotlivých sloupců je popsán v tabulce č. 4. Pro nepopsané hodnoty je dosti vypovídající již jejich název.

Název sloupce	Význam hodnot
sec	Vazba s tabulkou zabezpečení sítě (tabulka wifi_security)
comment, name, type, flags	Nepoužívané hodnoty získané z Wigle, ukládané pro možnost dalšího rozšíření
freenet, paynet, wep	Zabezpečení podle Wigle, v aplikaci je zpracováno a sjednoceno v hodnotě sec
firsttime, lasttime, lastupdt	Časové hodnoty Wigle, první měření, poslední měření a poslední aktualizace
channel	Použitý kanál
bcninterval	Interval beacon
accuracy	Přesnost polohy (pokud byla ze zdroje zjištěna)
calculated	Uložení vypočtené polohy z Wigle před získáním všech měření.

Tabulka č. 4 Význam hodnot v tabulce přístupových bodů

4.3.2 Databáze vizualizace dat

Vizuální část aplikace využívá data z dříve popsané tabulky přístupových bodů. Další součástí GUI jsou také statistiky a seznam vytvořených požadavků. Aplikace si také některé údaje (zejména průběh získávání dat a případné chyby) zaznamenává. Pro tyto funkce byly vytvořeny struktury zobrazené na obrázku č. 11.



Obrázek č. 11 Databáze - ostatní tabulky

Některé funkcionality týkající se získávání dat a zpracování požadavků, zejména pak změny hodnot znamenajících počet zpracovaných MAC adres z požadavku a další, jsou prováděny pomocí databázových triggerů. Celá struktura databáze, včetně triggerů, je přiložena v příloze 1.

4.4 Aplikační vrstva aplikace

Již z MVC architektury vyplývá rozdělení do částí model, kontrolér (v Nette Frameworku presenter) a pohled. Pohledy slouží pouze jako prezentace dat, pro popsání funkce jsou důležitější prezentéry a modely. Vrstva model se dále dělí do 3 částí – entity, reprezentující základní používané objekty, servisní třídy, v nichž je většina aplikační funkcionality, a utility (někdy nazývané helpers), které slouží jako pomocné nástroje. V tabulkách č. 5, 6 a 7 jsou popsány odpovědnosti a funkcionality jednotlivých prezentérů, utilit a servisních tříd.

Název třídy	Odpovědnost a funkcionality
ArrayUtil	Práce s poli
Color	Modelová třída a zároveň utilita, která umožňuje vygenerovat náhodnou barvu.
Coords	Třída reprezentující rozsah souřadnic. Obsahuje funkce pro automatické určování počáteční a koncové latitude i longitude, zvyšování rozsahu na k-násobek, výpočet středu a velikosti plochy a výpočet vzdálenosti v metrech.
MyUtils	Další pomocné metody jako práce s MAC adresou, vytváření klíče pro cache, převod obrázku do textu nebo změny nastavení serveru

Tabulka č. 5 Utility - odpovědnost, funkcionality

Některé výše popsané třídy jsou kombinací dat a funkcionality, jiné pouze funkcionality. Modelové třídy reprezentující samotné entity není nutno popisovat, protože jejich odpovědnost je velmi nízká. V tabulce č. 6 je popis prezentérů.

Všechny prezentéry dědí od třídy BasePresenter, která poskytuje základní funkce a nastavení používané ostatními prezentéry. Funkcí prezentérů je logicky oddělit části aplikace a delegovat úkoly dalším modelovým třídám. Většina funkce prezentérů je tedy založena na zpracování dat, předaných požadavku z uživatelského rozhraní, využití servisní třídy k nějaké činnosti a použití zobrazení k navrácení výsledku uživateli.

Název prezentéru	Odpovědnost a funkcionality
ApiPresenter	Export dat do CSV, import požadavků pro získání dat
DownloadPresenter	Získávání dat, zpracování požadavků
HomepagePresenter	Domovská stránka s mapou
RequestsPresenter	Stránka vytvořených požadavků
StatisticsPresenter	Stránka statistik, generování nových statistik
WifiPresenter	Překryvná vrstva mapy, zpracování kliknutí do mapy a zobrazení detailu

Tabulka č. 6 Prezentéry - odpovědnost, funkcionality

Největší počet tříd obsahuje servisní vrstva aplikace. Servisní třídy slouží k práci s daty, získávání a ukládání dat do databáze a zpracování dat. Odpovědností tříd servisní vrstvy jsou výpočty, zpracování dat a transformace dat. Většina servisních tříd potřebuje ke své práci spojení s databází, které je zajištěno přes třídu BaseService. Servisní třídy jsou do prezentérů injektovány, to zajišťuje Nette Framework. Některé servisní třídy jsou popsány v tabulce č. 7.

Název servisní třídy	Odpovědnost, funkcionalita
Download	Ukládání přístupových bodů do databáze. Třídy pro získávání dat ze zdrojů od této třídy dědí a využívají její funkce.
DownloadRequest, WigleDownloadQueue	Vytváření a zpracování požadavků pro získání dat. Analýza překryvné vrstvy Wigle.
DownloadImportService	Správa požadavků vytvářených pomocí importu MAC adres.
GoogleDownload, WigleDownload, WifileaksDownload	Získávání dat z konkrétních zdrojů. Tyto třídy jsou potomky třídy Download.
NotifyEmailService	Správa notifikací a požadavků o notifikaci od uživatele.
WifiManager, OptimizedWifiManager	Práce s konkrétními body. Třída OptimizedWifiManager vynechává databázovou vrstvu Nette a data vrací jako asociativní pole místo objektů.
OverlayRenderer	Generování dlaždic překryvné vrstvy.
Logger	Logování událostí a chyb.
StatisticsManager	Generování statistik a práce se statistikami.

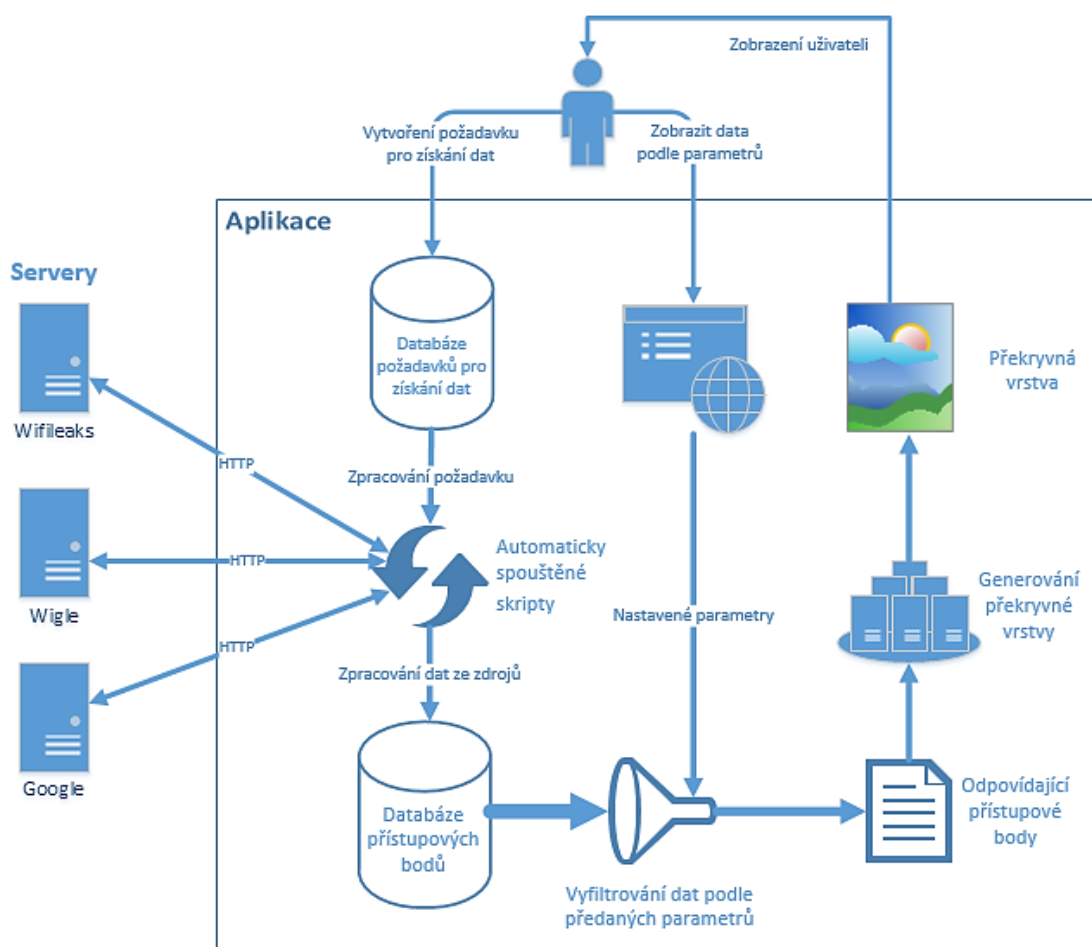
Tabulka č. 7 Servisní třídy – odpovědnost, funkcionalita

V aplikaci jsou i další servisní třídy, například SourceManager pro práci se zdroji dat v databázi a další. Ty slouží pouze k tvorbě objektů z databázových dat, a proto nejsou tak podstatné a není nutné je podrobněji popisovat.

Konkrétnější funkcionalita některých částí aplikační i datové vrstvy aplikace je popsána v části Návrh a Implementace.

5 Návrh a implementace

Jak již vyplývá z analýzy tak se aplikace bude dělit na dvě základní části a to část určenou pro získávání dat a část pro jejich vizualizaci. Stejně tak bude rozdělená i databáze. Uživatel v aplikaci bude mít možnost vytvářet požadavky pro získání dat a ve vizuální části aplikace bude moci pomocí nastavených filtrů ovlivnit způsob zobrazení a určit, která data budou zobrazena. Tato struktura aplikace je ukázána na diagramu na obrázku č. 12.



Obrázek č. 12 Diagram funkcionality aplikace

Na diagramu lze vidět, že uživatel má možnost vytvořit požadavek pro získání dat. Ten bude uložen do části databáze, která slouží k ukládání požadavků. Následně bude tento požadavek pomocí automaticky spouštěných skriptů zpracován, a tyto skripty zajistí získání a zpracování dat ze zdrojů. Nakonec jsou získaná a zpracovaná data uložena do databáze přístupových bodů.

Další funkcí, kterou lze vidět na diagramu je ovlivnění zobrazení. Uživatel aplikace vytvoří požadavek na zobrazení dat, přičemž v aplikaci nastaví parametry zobrazení a filtr dat. Aplikace předané parametry zpracuje a s jejich pomocí vyfiltruje požadovaná data z databáze přístupových bodů. Z toho vznikne seznam požadovaných dat, který je předán části starající se o generování překryvné vrstvy. A po vygenerování je tato překryvná vrstva zobrazena uživateli.

5.1 Získávání dat

Nejdůležitější součástí aplikace je samotné získávání dat z různých zdrojů. To probíhá automaticky pomocí periodicky spouštěných skriptů. Proces od zadání požadavku až po získání konkrétních informací je velmi komplikovaný a zdlouhavý a liší se podle zdroje dat. Obecně se začíná určením požadovaných dat, poté se tyto požadavky zpracují a následně se začnou získávat data.

5.1.1 Určení dat – vytvoření požadavků

Aby bylo možné nějaká data získat, je nejprve nutné určit, která data jsou požadována. Požadovaná data lze určit několika způsoby, jež se liší jak samotným vytvořením požadavku, tak i jeho následným zpracováním. Tyto způsoby jsou: zadání rozsahem zeměpisných souřadnic, zadání na základě již existujícího přístupového bodu nebo pomocí importu MAC adres přístupových bodů, o kterých je požadováno zjištění dalších informací.

K získání dat z Wifileaks se žádné požadavky nevytváří, k uložení dat dojde jednorázově pomocí zpracování souboru, který lze získat na webu Wifileaks. Z toho důvodu jsou požadavky vytvářeny pro zdroj Wigle a Google.

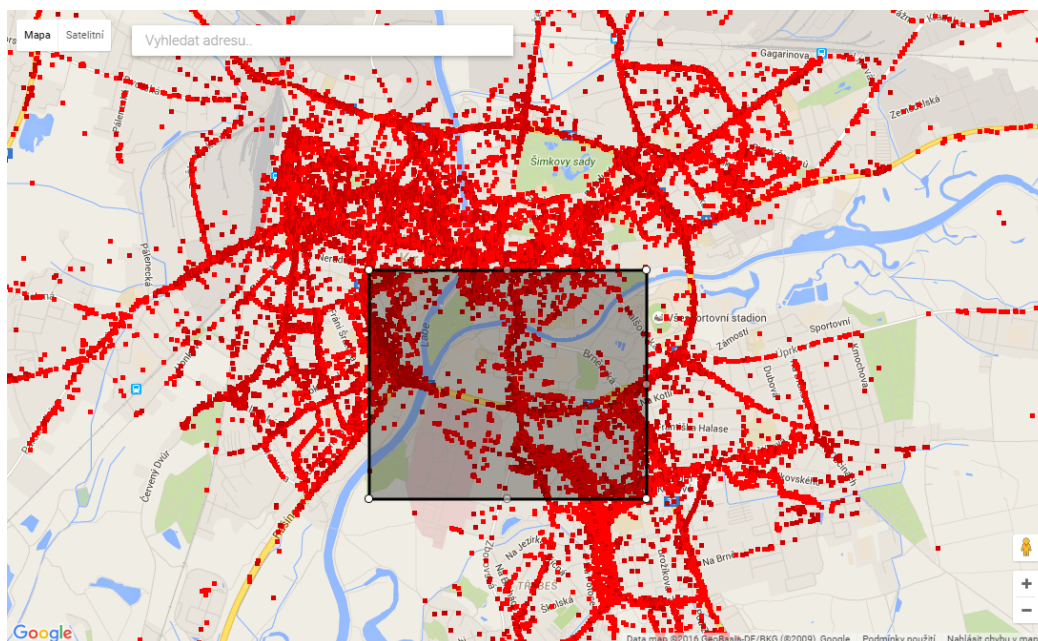
5.1.1.1 Importem konkrétních MAC adres

Při vytvoření požadavků importem konkrétních MAC adres aplikace načte soubor s MAC adresami, kde na každém řádku je jedna MAC adresa, která se po zpracování uloží do databáze. Požadavky vytvořené tímto způsobem mají vyšší prioritu. Data jsou získána nejprve z Wigle, do jehož fronty se požadavky přidají automaticky, protože není třeba je dále zpracovávat. A následně se informace získají i z Google. K tomu je v databázi speciální tabulka, do které jsou přidávány

importované MAC adresy, a podle příznaku, který znamená aktuální stav každého řádku, jsou s nimi prováděny akce.

5.1.1.2 Zadání rozsahem zeměpisných souřadnic

Metoda určení dat rozsahem zeměpisných souřadnic je nejobecnější. Stačí zadat požadovaný zdroj dat a obdélník určený počáteční a koncovou zeměpisnou šířkou a počáteční a koncovou zeměpisnou délkou. V aplikaci je pro usnadnění možnost tyto souřadnice určit graficky v mapě pomocí obdélníku, kterým lze pohybovat a měnit jeho rozměry. Na obrázku č. 13 je ukázka vytváření požadavku pomocí rozsahu zeměpisných souřadnic.



Obrázek č. 13 Vytvoření požadavku zadáním rozsahu zeměpisných souřadnic

Požadavky vytvořené zadáním rozsahu zeměpisných souřadnic se od sebe liší v následném zpracování požadavku podle zdroje dat. Pro získání dat z Google nejprve musí být v databázi přístupové body z Wigle, proto se při vytváření požadavku na data z Google vytvoří požadavky dva – a to jak na Wigle, tak na Google. Proto je nutné požadavek na Google blokovat dokud není zpracován požadavek na Wigle (viz Blokování požadavků). Z důvodů omezení Wigle API jsou požadavky už při vytváření předzpracovány (popsáno dále v textu).

Blokování požadavků

Blokování požadavků je funkcionalita, která umožňuje požadavky navzájem blokovat. Jeden požadavek může být blokován i více požadavky najednou. Takový požadavek bude blokován do té doby, než budou zpracovány všechny požadavky, na které blokováný požadavek čeká. Konkrétně se tedy požadavek na získání dat z Google nezačne zpracovávat, dokud se pro stejnou plochu nezpracuje požadavek na získání dat z Wagle.

Předzpracování požadavku zadaného rozsahem zeměpisných souřadnic

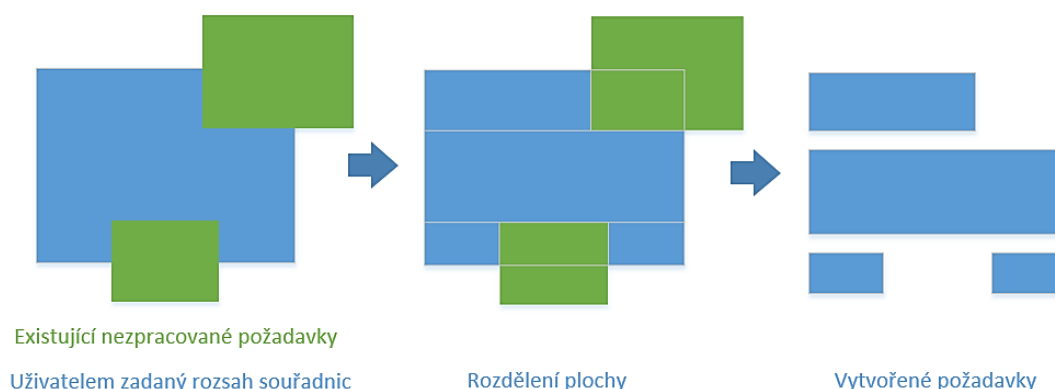
Již při vytváření projde požadavek částečným předzpracováním skládajícím se z několika kroků.

Nejprve dojde k zaokrouhlení zadaných souřadnic na 2 desetinná místa. Poté se aplikace pokusí nalézt v databázi již existující požadavek přesně odpovídající nově vytvářenému. Pokud takový požadavek existuje, ale ještě nebyl zpracován nebo byl zpracován před méně než třiceti dny, vytváření požadavku se zruší. Pokud proces vytváření požadavku v této fázi neskončí, postupuje se do třetí fáze.

Ve třetí fázi může dojít k rozdělení požadavku na několik menších podle potřeby. Při zahájení třetí fáze se získají všechny již existující a zároveň nezpracované požadavky, které zasahují do plochy vytvářeného požadavku. Poté dojde k mapování zeměpisných souřadnic na indexy v poli, čímž dojde k odstranění problému s různými rozsahy souřadnic. V první fázi mapování se mapuje vytvářený požadavek a poté všechny další požadavky, které byly nalezeny v daném rozsahu souřadnic. Nakonec dojde k seřazení mapovaného pole podle velikosti. Toto se provede pro zeměpisnou šířku (X) i zeměpisnou délku (Y).

Na základě těchto polí dojde k vytvoření dvourozměrného pole, které se naplní hodnotami 0 a 1 podle toho jestli v dané oblasti již je existující požadavek. Nula znamená, že v dané oblasti není žádný existující požadavek, jednička znamená, že existuje. V tomto poli se následně vyhledají spojitě bloky nul, pomocí pole mapování zeměpisných souřadnic na indexy v poli se indexy těchto bloků zpětně převedou na zeměpisné souřadnice a všechny tyto nalezené bloky se přidávají jako požadavek.

Ukázka rozdělení vytvářené plochy na několik menších je znázorněna na obrázku č. 14.



Obrázek č. 14 Rozdělení požadavku na několik menších

5.1.1.3 Zadání rozsahem zeměpisných souřadnic – s aplikovaným filtrem

Zadání rozsahem souřadnic lze použít i při aplikovaném filtru. V tomto případě se uživateli zobrazí informace, že bude vytvořen požadavek pouze na vyfiltrované body. Tyto požadavky se vytvoří s vyšší prioritou, takže jsou zpracovány přednostně a také lze odhadnout čas, který bude trvat získání dat. Proto se tato metoda vytvoření požadavku funkčně dělí na 2 části – část pro výpočet času a část, která přidá požadavky s vyšší prioritou do databáze. U tohoto módu má uživatel možnost nechat se informovat emailem poté, co budou všechna požadovaná data získána v databázi.

Tato metoda bude použita automaticky v případě, že uživatel bude mít nastavený filtr na data a zároveň iniciuje vytvoření požadavku pro získání dat z Wigle nebo Google.

Výpočet času nutného k získání dat

Při vytvoření a při každém následném pohybu obdélníku, nímž uživatel zadává rozsah souřadnic, bude asynchronně volána funkce, jejíž účelem je výpočet času, nutného k získání dat do databáze. Čas bude zobrazován ve formě „čas nutný k získání vyfiltrovaných dat + čas, nutný k zpracování ještě nezpracovaných požadavků se stejnou nebo vyšší prioritou“. Funkci jsou předány aktuální označené zeměpisné souřadnice, zdroj, z kterého budou data požadována, a aktuálně nastavený filtr.

Funkce zpracuje obdržené hodnoty, nalezne všechny aktuálně vyfiltrované body, které leží v zadaném rozsahu souřadnic, a spočítá počet unikátních MAC adres.

Poté následuje výpočet času v minutách, který je určen podle požadovaného zdroje dat. Pokud uživatel požaduje data z Wigle, tak je čas vypočten jako počet vyfiltrovaných unikátních MAC adres krát čas, v který je periodicky volán skript pro získání konkrétních bodů z Wigle, aktuálně je to 30 minut, k tomu je přičten stejnou metodou vypočtený počet minut nutný k získání již aktuálních požadavků.

Pokud uživatel požaduje data z Google, je čas vypočten jako součet času nutného k získání těchto dat z Wigle, protože před každým požadavkem na Google se nejprve data získají z Wigle, a času, který bude získání trvat z Google. Čas, který bude získání trvat z Google je určen periodou, v které je spuštěn skript na získání dat z Google.

Po výpočtu doby nutné k získání dat v minutách je tento čas převeden na počet hodin a počet minut. Výsledný řetězec vypadá například takto: „Získání dat bude trvat 5 hodin a 30minut + 9 hodin a 35 minut“. Vypočten a zobrazen je pro usnadnění i součet těchto časů.

Jelikož nedává smysl, aby uživatel vytvářel požadavek, jehož zpracování bude trvat několik měsíců, byl maximální čas nutný k získání dat stanoven na 1 týden (168 hodin). Pokud vypočtený čas tuto hodnotu přesáhne, je tlačítko, kterým uživatel doopravdy požadavky vytvoří, skryto. Skrytí nastává i v případě, když uživatel zadá rozsah souřadnic, v kterém nemá vyfiltrovaný ani jeden bod.

Vytvoření požadavků s nastaveným filtrem

Poté, co uživatel vybere oblast s vyfiltrovanými požadavky a čas nepřesahuje maximální povolený čas, může uživatel navíc zadat svůj email, na který si přeje být informován, až budou data získána. Po kliknutí na tlačítko „Požádat o stažení“ je vyvolána akce, které jsou předány stejné parametry, jako akci pro výpočet času nutného k získání dat. Tato akce je stejná pro vytváření požadavku rozsahem souřadnic bez nastaveného filtru i pro vytvoření požadavku rozsahem souřadnic s nastaveným filtrem. Akce se podle obdržených dat rozhodne, a pokud žádný filtr nastavený není, dojde k vytvoření požadavku základním způsobem (bez nastaveného filtru).

V opačném případě se provede podobná činnost jako v případě výpočtu času: nejprve se podle rozsahu souřadnic a nastaveného filtru získají všechny

odpovídající unikátní MAC adresy a pokud uživatel vyplnil i email pro notifikaci, tak je v databázi vytvořen záznam o notifikaci, ke kterému dále budou přiřazeny data, která souvisí s vytvářeným požadavkem. Poté je rozhodnuto podle zdroje, z kterého jsou data požadována - pokud je to Wifigle tak jsou MAC adresy přidány do tabulky požadavků na konkrétní měření z Wifigle, ale s vyšší prioritou. Pokud jsou požadována data z Google, tak jsou MAC adresy přidány stejně, jako kdyby byly importovány ze souboru. Pokud uživatel vyplnil email, tak jsou nakonec všechny vytvořené záznamy přiřazeny k záznamu o požadované notifikaci.

Informování uživatele o získání dat

Další funkce, která bude spouštěna opakovaně, je funkce kontrolující notifikace. Pro každou neodeslanou notifikaci získá díky propojení počet ještě nezpracovaných MAC adres. Pokud je tento počet nulový znamená to, že všechny požadavky týkající se této notifikace byly zpracovány a uživateli je odeslán informační email.

5.1.1.4 Na základě existujícího přístupového bodu

Určení požadavku na základě již existujícího bodu je velmi podobné metodě zadání importem MAC adres. Rozdíl je v tom, že MAC adresa se určí z již existujícího přístupového bodu v místní databázi. Z toho důvodu je tato metoda použita pouze pro vytvoření požadavku pro zdroj Google.

Google neumožňuje získat informace na základě jediné MAC adresy, vzhledem k využití funkce pro určení polohy vyžaduje Google alespoň 2 přístupové body v okolí. Požadavek na Google je tedy určen dvěma již existujícími přístupovými body, kdy první je požadovaný bod a druhý je bod, který je nejbližší k požadovanému. Algoritmus vychází z článku Automated WiFi-based Localization and Visualization of Wireless Network [21].

5.1.2 Zpracování požadavků

Po vytvoření požadavků, je nutné požadavky zpracovat. To probíhá zcela automaticky, pomocí opakovaného automatického spouštění úloh starajících se o zpracování. Kvůli tomu, že požadavky lze vytvořit několika odlišnými způsoby, jsou

požadavky rozmístěny do více databázových tabulek podle typu požadavku. Přičemž každý typ požadavku musí být zpracováván odlišným způsobem.

K zpracování dochází pouze u požadavků, které se týkají zdrojů Wigle a Google a byly vytvořeny na základě rozsahu zeměpisných souřadnic. Všechny ostatní způsoby vytvoření požadavků jsou již dostatečně konkrétní.

5.1.2.1 Zpracování požadavků – Wigle

Zpracování požadavků pro získání dat ze zdroje Wigle je zdlouhavý proces. Vykonává se automaticky jednou za hodinu.

Nejprve se z databáze požadavků získá jeden požadavek, který není blokován a ještě nebyl zpracován. Ten projde zpracováním, po kterém vznikne několik menších požadavků, které jsou určeny hustotou sítí v dané ploše. Toto zpracování je založeno na analýze překryvné vrstvy, kterou Wigle generuje do své mapy.

Analýza překryvné vrstvy

Nejprve dojde k zjištění barev, kterými Wigle označuje ve své překryvné vrstvě jednotlivé body. Barvy jsou určeny obrázkem umístěným na webu Wigle, který projde analýzou, a všechny barvy jsou uloženy do pole. Toto pole se poté využívá při analýze překryvné vrstvy. Rozsah analyzovaných barev se nachází na obrázku č. 15.



Obrázek č. 15 Rozsah barev přístupových bodů v překryvné vrstvě Wigle

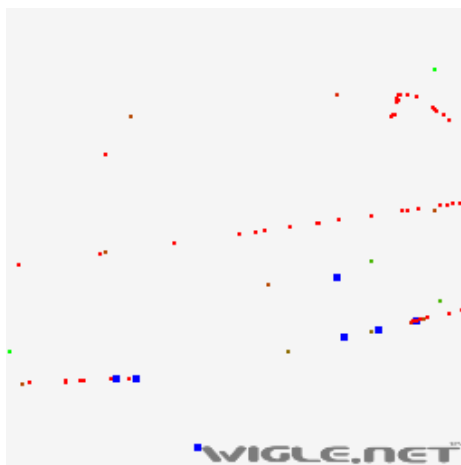
Poté dojde k základnímu rozdělení rozsahu souřadnic požadavku na plochy o velikosti odpovídající $0,05^\circ$ zeměpisné šířky i délky, pokud je to možné. V případě, že to možné není, nebo v případě že plocha není přesně dělitelná tímto číslem, zůstane celá plocha nebo pouze zbývající část nerozdělená.

Plocha s rozsahem zeměpisné šířky 50,15 – 50,23 a zeměpisné délky s rozsahem 15,65 – 15,77 tedy bude rozdělena na 6 ploch, viz tabulka č. 8.

Plocha	Rozsah zeměpisné šířky	Rozsah zeměpisné délky
1	50,15 – 50,20	15,65 – 15,70
2	50,15 – 50,20	15,70 – 15,75
3	50,15 – 50,20	15,75 – 15,77
4	50,20 – 50,23	15,65 – 15,70
5	50,20 – 50,23	15,70 – 15,75
6	50,20 – 50,23	15,75 – 15,77

Tabulka č. 8 Ukázka počátečního rozdělení plochy

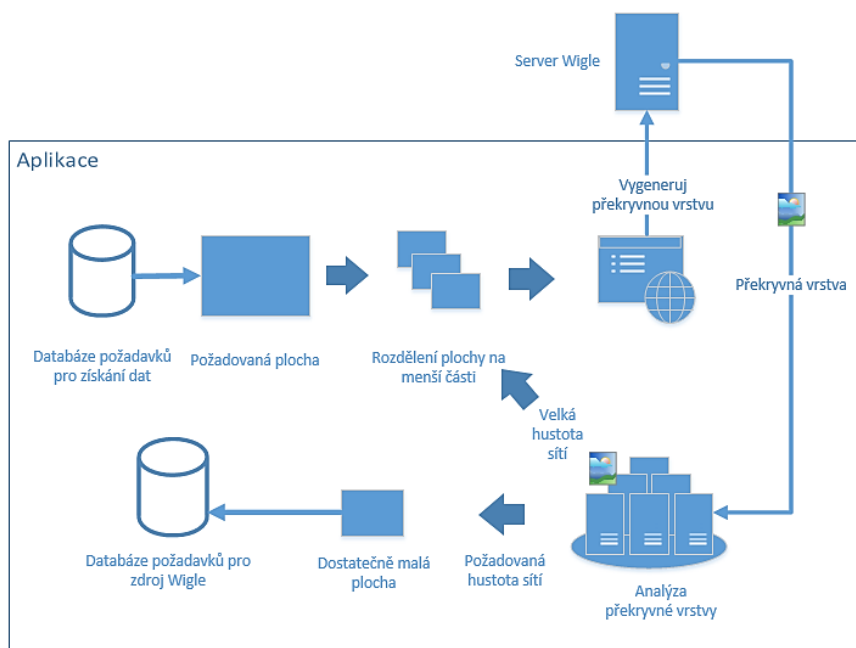
Po tomto rozdělení projde toto pole menších ploch analýzou, při které mohou některé plochy být dále rozděleny. Pro každou plochu se získá z Wigle překryvná vrstva (viz obrázek č. 16). Na ní se spočítá počet pixelů, které mají některou z barev, kterými jsou označeny přístupové body, a pokud celkový počet pixelů s některou z určených barev je více než určená hodnota, tak je plocha rozdělena na 4 menší plochy a celý proces analýzy překryvné vrstvy se pro každou tuto plochu spustí znovu. Takto proces pokračuje rekurzivně až do doby, než se zpracují všechny plochy z počátečního rozdělení i plochy vygenerované v průběhu analýzy.



Obrázek č. 16 Ukázka překryvné vrstvy Wigle

Po dokončení analýzy překryvné vrstvy jsou data, vzniklá analýzou, zpracována a vzniká pole s menšími plochami určenými hustotou přístupových bodů v oblasti. U každé plochy se pamatuje vypočtený počet přístupových bodů. Tato hodnota je užitečná pouze jako statistický údaj, který lze použít k vylepšení algoritmu pro analýzu překryvné vrstvy. Všechny vygenerované plochy jsou

následně uloženy do databáze, a požadavek, na jehož základě byly plochy vygenerovány, je označen jako zpracován a rozšířen o hodnotu, na kolik menších ploch byl rozdělen. Této hodnoty se využívá k určení, zda jsou již všechna data získána. Požadavky, které jsou blokovány tím, že čekají na tento požadavek, nemohou být zpracovány do té doby, než budou v databázi i data přístupových bodů. Celý postup zpracování požadavku pro získání dat ze zdroje Wigle je znázorněn v diagramu na obrázku č. 17.



Obrázek č. 17 Zpracování požadavku - Wigle

5.1.2.2 Zpracování požadavků – Google

Zpracování požadavků pro zdroj Google je snazší. Jeho implementace je podobná jako vytváření požadavku pro zdroj Google z jiného existujícího přístupového bodu. Vytvoření požadavků z přístupových bodů na zadané ploše ale proběhne automaticky. Tyto požadavky jsou zpracovávány automaticky každých 30 minut.

Opět se začíná nalezením neblokovaného a nezpracovaného požadavku, s rozdílem, že se hledá požadavek určený pro zdroj Google. Poté se naleznou všechny existující přístupové body v zadaném rozsahu zeměpisných souřadnic a pro každý bod je vygenerován požadavek, stejně jako v případě určení požadavku z existujícího bodu.

Nakonec se požadavek označí jako zpracovaný a tím je zpracování požadavku pro zdroj Google dokončeno.

5.1.3 Získání přístupových bodů

Po vytvoření požadavků a jejich předzpracování je na řadě získávání dat z vybraných zdrojů. Data z Wifileaks jsou získávána najednou, takže zdroje Wifileaks se vytváření požadavků ani jejich zpracování netýkalo. Získání informací o přístupových bodech je posledním krokem, po kterém již budou v databázi konkrétní použitelná data.

5.1.3.1 Wifileaks

Data z Wifileaks jsou uložena v TSV souboru, který lze buď mít lokálně uložený v adresáři temp, nebo ho aplikace dokáže získat z webu Wifileaks.

```
00:00:00:00:00:00 HP444DA4 4 49.2033903377 16.629932187956 268 2012-01-13 15:12:52
00:00:00:00:00:00 HPEF8AD3 4 49.195362802448 16.611118352567 238 2012-01-10 15:40:53
00:00:00:00:00:00 MTG 4 49.309743394517 16.395127262934 267 2015-01-09 06:41:52
00:00:00:00:00:00 HP50D28B 4 50.062774720427 14.442662881776 265 2012-01-12 20:09:18
00:00:00:00:00:00 HP-nomodel.17368D 4 50.669047302339 13.998148219669 232 2012-01-12 15:21:40
00:00:00:00:00:00 HP-nomodel.D1C89E 4 49.744069313945 18.614249382624 332 2012-01-12 18:10:55
```

Obrázek č. 18 Ukázka struktury TSV souboru

Soubor má jednoduchou strukturu (viz obrázek č. 18). Na začátku je komentář popisující význam hodnot a pod ním je na každém novém řádku jeden záznam se všemi daty. Oddělovačem jednotlivých hodnot je tabulátor. Hodnoty v každém záznamu jsou: MAC adresa, SSID sítě, typ zabezpečení, zeměpisné souřadnice a datum aktualizace. Pokud daný řádek je komentář, je označen hvězdičkou jako prvním znakem řádku. Typ zabezpečení je udán hodnotou z intervalu od 0 do 4. Význam jednotlivých hodnot je popsán v tabulce č. 9.

Hodnota	Typ zabezpečení
0	Otevřená síť (žádné zabezpečení)
1	WEP
2	WPA1
3	WPA2
4	Jiné zabezpečení

Tabulka č. 9 Význam hodnot zabezpečení

Zpracování lokálního souboru probíhá jednoduše. Soubor se otevře pro čtení a postupně se zpracovávají jednotlivé řádky. Pokud je řádek komentář nebo nemá žádný obsah, nezpracovává se. U ostatních řádků dojde ke zpracování a následně jsou data ukládána do databáze. Z důvodu optimalizace probíhá ukládání dat hromadně po tisíci záznamech. Nakonec je soubor uzavřen.

Zpracování souboru uloženého na webu Wifileaks je o něco složitější. Nejprve aplikace otevře odpovídající webovou adresu a pomocí regulárního výrazu nalezne všechny soubory s odpovídajícím názvem, například wileaks_150709.tsv.

Číselná hodnota znamená datum, v tomto případě tedy 9. 7. 2015. Ze všech nalezených souborů se za pomoci této číselné hodnoty vybere nejnovější soubor. Jeho název je porovnán s názvem naposledy zpracovaného souboru, který je uložen v databázi. A pokud je název stejný znamená to, že již jsou aktuální data v databázi a soubor se tedy nebude zpracovávat. V opačném případě je soubor otevřen a zpracuje se stejným způsobem, jako kdyby byl soubor uložen lokálně. Nakonec se aktualizuje hodnota naposledy zpracovaného souboru u zdroje Wifileaks na název právě zpracovaného souboru a tím je proces u konce.

5.1.3.2 Wigle

Po zpracování požadavku pro zdroj Wigle je v databázi spousta malých ploch, pro které jsou požadovány data z Wigle, a které mají informaci o tom, z jakého uživatelského požadavku byly vygenerovány. Další proces získání dat z Wigle se dělí na dvě části - získání MAC adres všech sítí podle rozsahu zeměpisných souřadnic a poté získání konkrétních dat pro každou MAC adresu.

Kvůli omezením, která Wigle má, je skript pro získávání dat z Wigle spuštěn každých 30 minut.

Získání MAC adres všech přístupových bodů v ploše ze zdroje Wigle

Před jakoukoli další komunikací s Wigle API je nutné provést přihlášení. Přihlášení je uloženo do souboru cookie, který se ve všech dalších požadavcích na Wigle API využívá. Komunikace s Wigle API probíhá pomocí PHP CURL dotazů na server s předáním potřebných hodnot, podle kterých API vrátí požadované údaje.

Po autorizaci uživatele se z databáze vezme náhodný nezpracovaný rozsah souřadnic, který byl zpracováním požadavku vygenerován. Žádost na Wigle API je poté sestavena pomocí několika povinných a případně nepovinných údajů. Povinné jsou atributy longrange1, longrange2, latrange1 a latrange2, což jsou rozsahy zeměpisné délky a šířky. Další dva nepovinné údaje jsou označovány jako first a last, což je první a poslední vrácený záznam. Ty je potřeba zadat pokud je v daném rozsahu souřadnic více než 100 přístupových bodů. Wigle API totiž neumožňuje jedním dotazem vrátit více než 100 záznamů. Vrácený údaj je JSON s hodnotami popsány v tabulce č. 10.

Hodnota	Význam
resultCount	Počet vrácených záznamů
success	Úspěšnost dotazu
first	Číslo prvního vráceného záznamu
last	Číslo posledního vráceného záznamu
results	Konkrétní záznamy

Tabulka č. 10 Hodnoty vrácené z Wigle API pro rozsah zeměpisných souřadnic

Pokud byl dotaz úspěšný, všechny vrácené záznamy jsou uloženy do databáze. Tyto záznamy mají polohu vypočtenou z konkrétních měření. Cílem je získat všechna tato měření, a proto aplikace MAC adresy všech získaných bodů uloží do stejné tabulky, jako požadavky, které byly pro zdroj Wigle vytvořeny importem MAC adres (tabulka wigle_aps).

Vygenerovaná plocha, která byla zpracována, se rozšíří o údaj počtu MAC adres získaných na základě této plochy. Dále je příznak označující počet MAC adres, z nichž již byly získány z Wigle i všechna měření nastaven na 0 a plocha je označena jako zpracovaná.

Pokud vrácená hodnota atributu resultCount je 100, což je maximální počet vrácených záznamů, je vytvořena další plocha založená na základě právě zpracované plochy, a je rozšířena o parametry first a last. Tím je zajištěno, že budou pro tento rozsah souřadnic získány všechny záznamy. Nakonec musí být upraven původní požadavek, na jehož základě tato plocha vznikla, takovým způsobem, že je navýšena hodnota označující celkový počet ploch, na kolik byl požadavek rozdělen.

Poloha u všech těchto záznamů je pouze vypočtená poloha ze všech měření daného přístupového bodu a proto je u záznamu nastaven příznak, který značí, že daný bod má vypočtenou polohu. Cílem je získat přesné údaje z těchto měření a k tomu slouží funkce získání konkrétních dat.

Získání konkrétních dat pro každou MAC adresu

Získání konkrétních dat pro každou MAC adresu je automaticky spouštěná funkcionální zajišťující získání všech měření, které Wigle poskytuje pro danou MAC adresu. Takto zpracovány jsou všechny MAC adresy, které byly získány na základě požadavků, nebo importovány. Importované MAC adresy mají vyšší prioritu, a proto mají při zpracování touto funkcí přednost.

Pro komunikaci s Wigle API je opět nutné se přihlásit. Poté se podle priority vybere jedna MAC adresa z databáze. Tato MAC adresa je předána Wigle API v parametru netid. Wigle API vrátí JSON se všemi měřeními týkajícími se dané MAC adresy. Vrácená data jsou popsána v tabulce č. 11.

Parametr	Význam
Success	Požadavek byl úspěšný
Wifi	Jedná se o WiFi síť
GSM	Jedná se o GSM přístupový bod
Result	Nalezené body s touto MAC adresou
CDMA	CDMA mobilní síť

Tabulka č. 11 Význam hodnot - Wigle API detail MAC adresy

Pro tuto aplikaci jsou zajímavé pouze WiFi sítě, proto hodnoty parametrů GSM a CDMA nejsou důležité. Vrácená hodnota parametru result obsahuje informace o přístupovém bodu, jako například SSID sítě, používaný kanál, zabezpečení, vypočtenou polohu a vnořené pole locationData, které obsahuje konkrétní měření, přesné zeměpisné souřadnice, sílu signálu a přesnost měření. Data pro každé měření se uloží do databáze. Následně se zpracovaná MAC adresa ve frontě MAC adres pro Wigle označí jako zpracovaná.

Pokud byla MAC adresa vytvořena na základě importu, tak je stav u dané MAC adresy v tabulce importu změněn na „staženo z Wigle“ a je z ní vytvořen požadavek

pro zdroj Google. Poté se opět změní stav u importované MAC adresy v tabulce importu na „přidáno do Google fronty“.

Pokud byla MAC adresa vygenerována na základě požadavku, zvýší se počet zpracovaných MAC adres, které byly vygenerovány na základě tohoto požadavku, o jedna. Pokud hodnota v tomto sloupci je stejná jako počet vygenerovaných MAC adres z vybrané plochy pro zdroj Wigle znamená to, že tato menší plocha již byla zpracována. V takovém případě je pomocí databázového triggeru upraven i původní požadavek, z kterého tato plocha vznikla, a to tak, že hodnota ve sloupci označujícím počet zpracovaných ploch se zvýší o jedna. Pokud upravená hodnota je stejná jako hodnota ve sloupci, který znamená počet ploch, na který byl uživatelský požadavek rozdělen, znamená to, že již celý tento požadavek pro zdroj Wigle byl zpracován. A proto všechny požadavky, které na tento čekaly, jsou odblokovány, pokud již nemusí čekat na žádný jiný.

Po získání konkrétních měření je celý proces získání dat z Wigle dokončen.

5.1.3.3 Google

Získání dat z Google probíhá až jako poslední, protože je závislé na datech, která již v databázi jsou. To je proto, že data z Google nejsou získána přímo, ale pomocí lokalizačních služeb Google.

Dotaz pro lokalizační služby Google musí obsahovat data minimálně dvou přístupových bodů. Konkrétně to je MAC adresa, SSID sítě a síla signálu. Síla signálu je využita k tomu, aby byla získána data o správném přístupovém bodu. Dotaz je vygenerován z dvou již existujících bodů v databázi: cílového bodu, jehož poloha je požadována a druhého bodu, který je nejbližší k požadovanému. Cílový bod je simulován jako bod, který je nejbližší zjišťované poloze, proto mu je vygenerována vysoká síla signálu (75 – 85dBm) a druhému bodu je vygenerována nízká síla signálu, aby příliš neovlivnil získaná data.

Takto sestavený dotaz se odešle na servery Google, z kterých je vrácen JSON s údaji. Vrácené údaje jsou: přesnost v metrech, status požadavku, jestli byl úspěšný nebo neúspěšný a poloha. Pokud v databázi Google údaje nejsou, nebo nejsou dostatečně přesné, je vrácena poloha s velmi nízkou přesností. Taková data jsou

ignorována a nebudou se ukládat. Pokud je přesnost vrácených dat vysoká, jsou data uložena a požadavek na Google se označí jako zpracovaný.

Počet požadavků na Google není významně omezen. Informace se získávají pro 10 bodů zároveň a funkce pro získání dat z lokalizačních služeb Google se provádí každých 5 minut.

5.2 Vizualizace dat

Vizualizace dat může být v mapách Google reprezentována několika způsoby. Základní způsob je pomocí značek (markers). Tento způsob ale není vhodný k zobrazování velkého množství dat a s narůstajícím počtem značek vysoce narůstá náročnost webu na hardwarové prostředky. Proto bude využita druhá možnost, a to generování obrázků překryvné vrstvy, které bude probíhat na serveru.

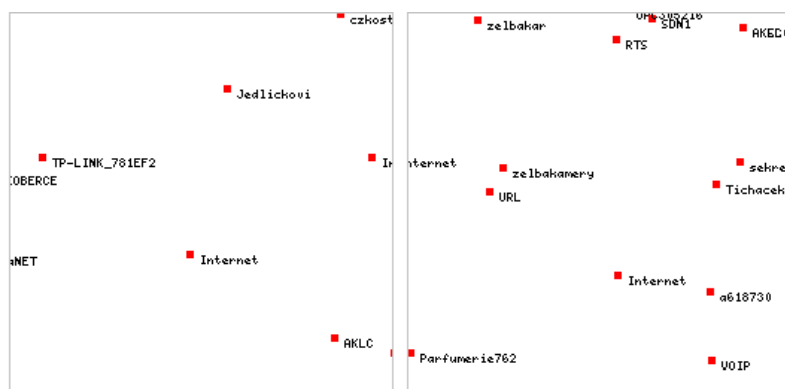
5.2.1 Generování překryvné vrstvy

Generování obrázků probíhá na serveru, na požádání. Google mapy mají integrovanou funkci, pomocí které si Google Maps API samo určí dlaždice, které potřebuje vygenerovat a následně je i automaticky umístí na správné místo v mapě. Tím pádem je nutné se postarat pouze o samotné vygenerování dlaždice. Její správné umístění i o žádost na vygenerování zajistí Google Maps API [15].

Každá dlaždice bude mít velikost 256 x 256 pixelů. Server obdrží vždy počáteční a koncové souřadnice odpovídající souřadnicím, na které bude dlaždice umístěna, hodnotu přiblížení mapy a podle nastaveného módu zobrazení obdrží další potřebné údaje. Samotné vygenerování se skládá ze dvou částí – nejprve je dle parametrů nutné získat správné body, ležící v této ploše, a poté z nich vygenerovat danou dlaždici.

Nejprve server zpracuje nastavené parametry a také dojde k zvětšení rozsahu souřadnic ve všech směrech o 12,5 %. Toto zvětšení je nutné proto, aby byly skutečně zobrazeny všechny body. Pokud by například nějaký bod odpovídal přímo okraji požadované dlaždice, tak by ve výsledné dlaždici byl oříznutý nebo téměř nebyl vidět. Proto dojde k vygenerování větší dlaždice, která je nakonec oříznutá do původní velikosti. Tím se docílí toho, že část zobrazení tohoto bodu bude na jedné

dlaždici, a zbytek na druhé, a ve výsledku to na překryvné vrstvě nepůjde poznat (viz Obrázek č. 19).



Obrázek č. 19 Ukázka návaznosti dlaždic

Z nastavených parametrů, zeměpisných souřadnic a míry přiblížení dojde k vygenerování klíče pro cache. Klíč se skládá z několika hodnot, první je nastavený mód, poté dvojtečkou oddělené souřadnice, a nakonec přiblížení, poté následuje čtyřtečka, a za ní všechny další parametry ve formátu „~[název parametru]=[hodnota parametru]“. Příklad vygenerovaného klíče v cache je: „MODE_SEARCH:50.21206:50.21909:15.809326:15.820312:15::~~ssidmac=eduroam~channel=3“. Pod vygenerovaným klíčem se server pokusí nalézt záznam v cache. Pokud ho nalezne, tak vrátí záznam z cache, odpovídající požadované dlaždici. V opačném případě skript pokračuje ve své činnosti.

Pokud skript neskončil, následuje získání dat, odpovídajících nastaveným parametrům, z databáze. Získání dat je specifické pro každý mód zobrazení. Získávaná data jsou: zeměpisná šířka, zeměpisná délka, SSID sítě, MAC adresa a zdroj, z kterého bod pochází.

V módu zobrazení všech sítí, módu pouze jedna síť a módu zobrazení vyfiltrovaných bodů stačí pouze získat data podle parametrů a poté pokračovat v tvorbě dlaždice.

U módu zvýraznění bodů podle parametrů nejprve dojde k zjištění, jestli jsou k dispozici všechny potřebné parametry, pokud ne, tak vygenerovaná dlaždice bude stejná jako u módu zobrazení všech bodů. Pokud k dispozici všechny potřebné parametry jsou, získají se nejprve všechny body, které budou zvýrazněné, a poté i všechny ostatní body odpovídající generované dlaždici.

Při použití módu zobrazení vypočtené polohy bodu je proces nejsložitější. Nejprve jsou získány všechny body odpovídající dané MAC adrese. Navíc je podle bodu, z kterého byl výpočet iniciován, vygenerován rozsah souřadnic, v kterém jsou nalezeny body s odpovídající MAC adresou. A následně je z těchto bodů vypočtena poloha pro nově vygenerovaný bod.

5.2.1.1 Tvorba obrázků v PHP

K tvorbě obrázků je v Nette Frameworku k dispozici API, které usnadňuje práci s obrázky. Z důvodu optimalizace rychlosti vykreslení dlaždice nebylo použito API Nette Frameworku, ale přímo funkcí které k práci s obrázky nabízí PHP.

Funkcí pro práci s obrázky je celá řada, popsány budou pouze funkce použité při implementaci aplikace.

K vytvoření obrázku slouží funkce `imagecreate`, které jsou předány hodnoty šířky a délky v pixelech. „Po vytvoření obrázku a před přidáním tvaru nebo textu musíte zadat barvy, které se budou používat, nejlépe přidělením proměnným“ [22]. K tomu existuje funkce `imagecolorallocate`, které je nutné předat referenci na obrázek a hodnoty barev v RGB spektru. Je vhodné výsledek funkce uložit do proměnné, která bude použita při kreslení objektu dané barvy. Pro alokaci průhledné barvy je využívána funkce `imagecolortransparent`.

Po vytvoření obrázku a alokaci barev je možné začít do obrázku kreslit objekty. V této aplikaci se do obrázku budou kreslit 3 druhy objektů – vybarvený čtverec, vybarvený kruh a text. Nakreslení vybarveného čtverce proběhne vykonáním funkce `imagefilledrectangle`, které je nutno jako atributy předat referenci na obrázek, počáteční a koncové souřadnice na osách X i Y a barvu. Vybarvený kruh je nakreslen funkcí `imagefilledellipse`, které jsou předány podobné parametry. Změna je v udání pouze souřadnic X a Y, na kterých se nachází střed kružnice a poté šířka a výška elipsy, která v případě vykreslování kruhu je totožná. Posledním vykreslovaným objektem je text. Text je vykreslen funkcí `imagestring`, které je nutno předat referenci na obrázek, typ fontu, souřadnice levého horního rohu kde bude text začínat, samotný text a barvu, kterou bude text vykreslen.

Nakonec bude obrázek ještě ořezán. Ořezání je zajištěno pomocí vytvoření nového obrázku o menších rozměrech, alokaci stejných barev jako v případě většího

obrázku, a poté bude výřez většího obrázku zkopírován do nového menšího obrázku funkcí imagecopy. Funkci imagecopy jsou předány reference na nový i původní obrázek, souřadnice v novém obrázku, na které bude výřez kopírován, souřadnice původního obrázku a výška a šířka původního obrázku. Souřadnice do původního obrázku jsou v tomto případě vypočtené jako (šířka velkého obrázku – šířka malého obrázku)/2, stejný výpočet je i pro výšku obrázku.

Po nakreslení všech objektů do obrázku se provede návrat obrázku ve formátu PNG funkcí imagepng a nakonec se provede funkce imagedestroy, jejíž účelem je uvolnění použitých zdrojů. Oběma těmito funkcím je nutné předat referenci na obrázek.

5.2.1.2 Samotné vygenerování dlaždice

Po vybrání správných dat, která se budou zobrazovat, je nutné vygenerovat požadovanou dlaždici. Generování se liší podle módu zobrazení, ale základ je stejný.

Nejprve je vytvořen obrázek v rozměrech 320 krát 320 pixelů, to je stejný poměr o jaký se zvětšoval rozsah souřadnic (12,5 % ve všech směrech). Obrázku jsou přiřazeny potřebné barvy a pozadí je nastaveno na průhlednou barvu.

V druhém kroku je nutné vypočítat konverzní poměr pro převod zeměpisných souřadnic do pixelového rastru. Tento poměr je vypočten jako podíl absolutní hodnoty rozdílu koncové a počáteční zeměpisné šířky a rozměrů obrázku. Vzhledem k malé velikosti dlaždic je možné výpočet provést tímto způsobem.

$$\frac{50.28933925329177 - 50.233151832472245}{320} = 0.000175585690061015625$$

Obrázek č. 20 Výpočet konverzního poměru zeměpisné šířky a pixelu

Tímto výpočtem získáme, jak velké zeměpisné ploše odpovídá 1 pixel obrázku (viz obrázek č. 20). Stejný výpočet je použit i pro zeměpisnou délku. Výpočtem jsou získány údaje, potřebné k mapování polohy bodů, určené zeměpisnými souřadnicemi, do obrázku.

Následuje procházení všech bodů, které se mají na dlaždici zobrazit. Pro každý tento bod je vypočtena poloha v pixelové mřížce odpovídající zeměpisným souřadnicím bodu. Jedna souřadnice je vypočtena jako velikost obrázku mínus podíl




mezi rozdílem zeměpisné šířky bodu a hodnotou, která odpovídá okraji požadované dlaždice, a hodnotou které odpovídá jeden pixel.

$$Y = \text{výška dlaždice} - \frac{(\text{zeměpisná šířka bodu} - \text{počáteční zeměpisná šířka dlaždice})}{\text{zeměpisná šířka odpovídající jednomu pixelu}}$$

Odečtení je nutné z důvodů posunutí bodu [0,0] do levého spodního rohu z původního levého horního rohu. Mapování souřadnic X není nutné otáčet a proto se X vypočte pouze jako podíl rozdílu zeměpisné délky bodu a počáteční zeměpisné délky a zeměpisné délky, která odpovídá jednomu pixelu. Tento výpočet a následné umístění konkrétního bodu na dlaždici není přesné, velikost chyby závisí na reálné velikosti plochy, které dlaždice odpovídá.

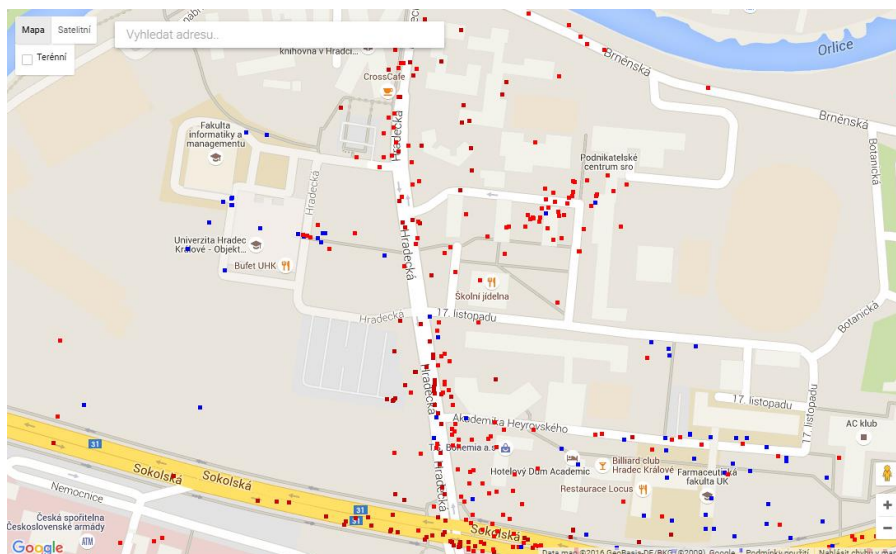
Právě ve způsobu vykreslování jednotlivých bodů je rozdíl mezi jednotlivými módy zobrazení. U některých módů se liší pouze barva, u jiných i obrazec, který představuje jedno měření bodu.

Módy zobrazení všech bodů a zobrazení vyfiltrovaných bodů jsou z hlediska vykreslování nejjednodušší. Pro každý bod jsou vypočteny souřadnice v dlaždici a podle zdroje dat daného bodu se barvou, určenou pro tento zdroj (viz tabulka č. 12), vykreslí čtverec. Všechny takovéto body mají červenou barvu, jednotlivé zdroje se od sebe liší konkrétním odstínem.

Zdroj dat	Wifileaks	Wigle	Google
Barva			
Kód barvy	#ff0000	#d00000	#c00000

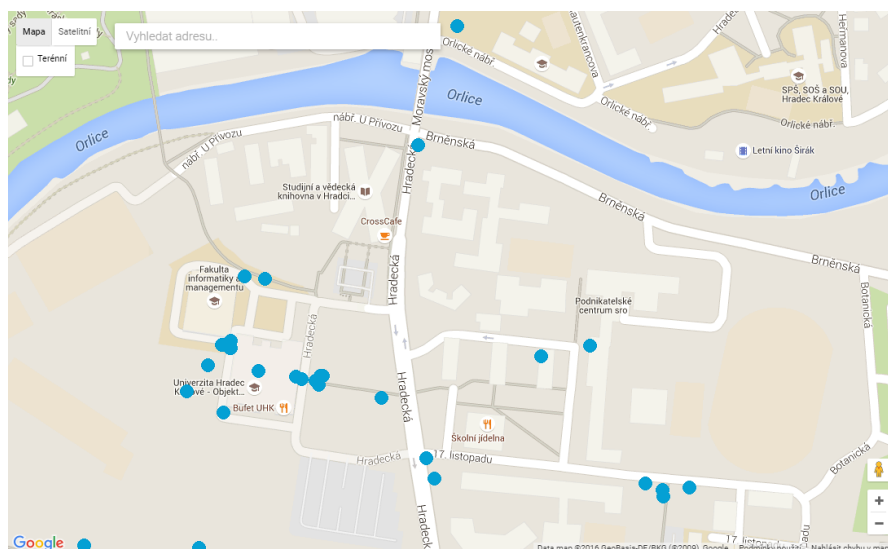
Tabulka č. 12 Odstíny barvy bodů podle zdroje dat

Při použití módu zvýraznění bodů v mapě podle parametrů, je navíc pro každý bod provedena kontrola, zdali má být bod označen. Pokud ano, tak se bod vykreslí modrou barvou, jinak je vykreslen barvou odpovídající zdroji dat tohoto bodu. Body v tomto módu jsou opět znázorněny barevným čtvercem o velikosti 4 krát 4 pixely. Ukázka překryvné vrstvy v módu zvýraznění bodů podle parametrů je na obrázku č. 21.



Obrázek č. 21 Body označené podle parametrů (SSID eduroam)

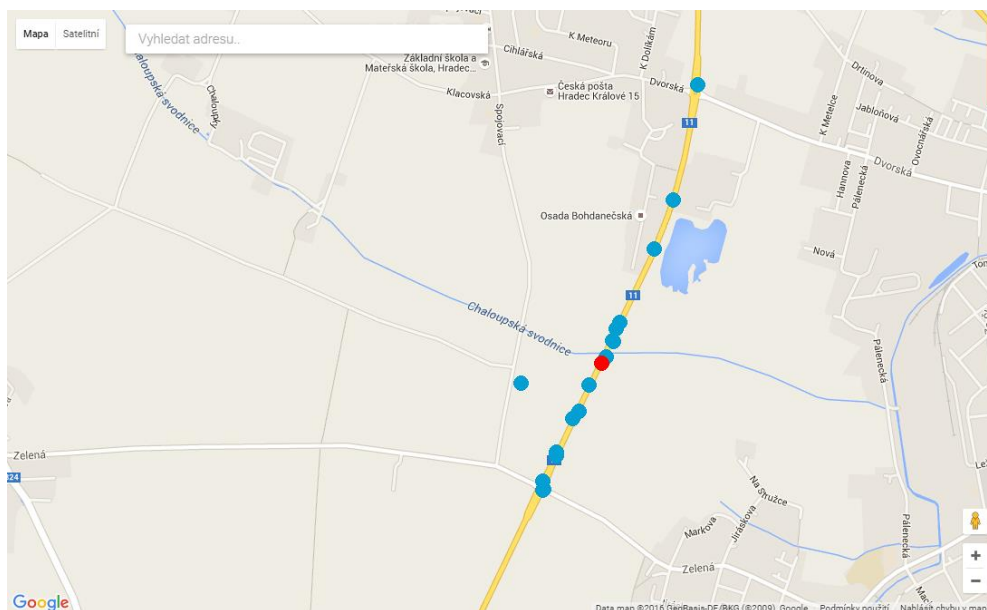
V módu zobrazení pouze jedné sítě není důležitý zdroj dat. Cílem je pouze znázornit pokrytí území, kde se nachází sítě se stejným SSID. Z toho důvodu jsou všechny body vykresleny jako kruh o průměru 16 pixelů stejnou modrou barvou (viz obrázek č. 22).



Obrázek č. 22 Zobrazení pouze jedné sítě

Posledním módem je zobrazení vypočtené polohy bodu. Vykresleny jsou pouze body, ze kterých byla poloha vypočtena, a to stejnou modrou barvou i ve stejných rozměrech jako při použití módu jedné sítě. Nakonec je vykreslen bod umístěný na souřadnicích vypočtených jako průměr z odpovídajících bodů – to

znamená bodů, které leží v okolí iniciačního bodu a mají stejnou MAC adresu. Viz obrázek č. 23.

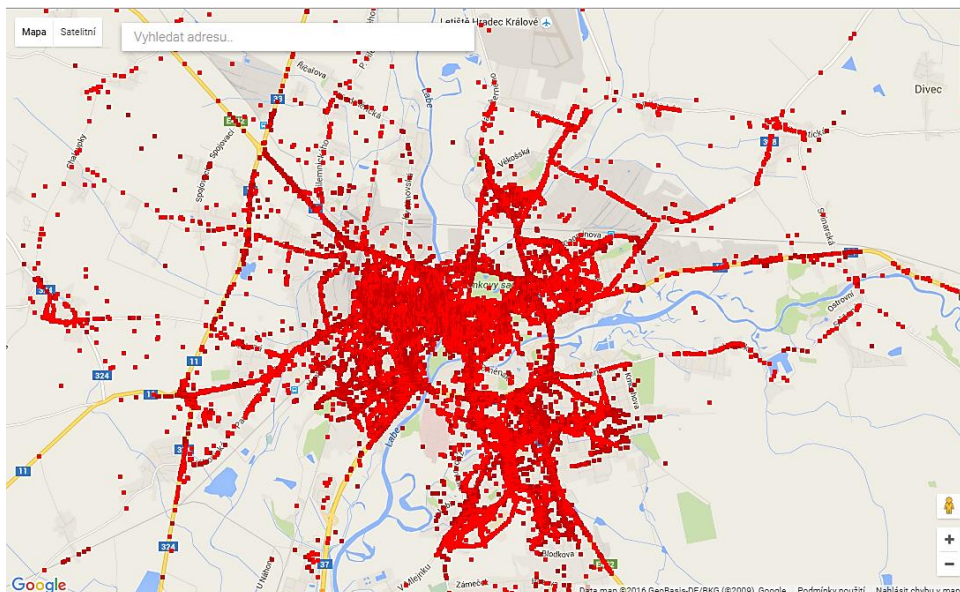


Obrázek č. 23 Mód vypočtená poloha bodu

U všech bodů se při dosažení dostatečného přiblížení, zobrazí i odpovídající SSID sítě, do které patří tento bod. Pokud SSID sítě u daného bodu není k dispozici, místo něj se zobrazí MAC adresa. Název sítě je automaticky zkracován na délku 20 znaků.

Po vykreslení všech bodů, které na dlaždici mají být, je dokončeno vykreslení dlaždice. Dlaždice má ale stále větší rozměry, než jsou potřeba, jak bylo psáno v předchozích odstavcích. Před vrácením je tedy nutno nejprve dlaždici ořezat do správných rozměrů. Ořezání proběhne tak, že je vytvořena dlaždice o požadovaných rozměrech (256 krát 256 pixelů) a do ní je namapován výřez o těchto rozměrech z vygenerované větší dlaždice.

Po vytvoření dlaždice o požadovaných rozměrech je dlaždice uložena do cache, pokud je cache aktivovaná a pro daný mód se mají dlaždice ukládat do cache. Následně už je vygenerovaný obrázek ve formátu PNG vrácen prohlížeči a pomocí Google Maps API zobrazen jako překryvná vrstva na správném místě mapy. Ukázka mapy s překryvnou vrstvou nad Hradcem Králové je na obrázku č. 24.



Obrázek č. 24 Mapa s překryvnou vrstvou

5.2.2 Překryvná vrstva – optimalizace

Generování překryvné vrstvy je stěžejní část, na které je založena většina funkcionality, s kterou pracuje uživatel webu. Proto je nutné, aby generování dlaždic do mapy bylo co nejrychlejší.

Optimalizace je vyřešena pomocí cachování, optimalizace skriptu, který dlaždice generuje a poslední způsob jak je vyřešeno snížení náročnosti na server, je použití polohy uživatele, kterou lze vyžádat od prohlížeče. Pokud zjištění své polohy uživatel umožní, bude mapa automaticky přiblížena na jeho polohu a tím se sníží množství přístupových bodů, které celkem na dlaždicích jsou a tím pádem i zvýší rychlost vykreslování, která je na počtu bodů přímo závislá.

Další použitá funkcionalita, která snižuje vytížení serveru, je omezení vykreslování překryvné vrstvy podle přiblížení mapy. Při velmi malém přiblížení bude generována vrstva s textem nabádajícím uživatele pro přiblížení mapy.

5.2.2.1 Cachování překryvných dlaždic

Nejjednodušší způsob optimalizace je cachování. Tento způsob se hodí pro aplikace s velkým množstvím požadavků vracejících stejná data, a také tam, kde se samotná data moc často nemění.

V této aplikaci není očekáváno nijak obrovské množství požadavků, ale bude využito cachování, a to zejména z důvodu, že generování překryvné vrstvy využívá

systemové zdroje serveru. A pokud uživatel mapu několikrát posune, oddálí, přiblíží na stejném místě, tak není žádoucí, aby se pro každou takovou událost generovaly všechny dlaždice znovu, ale mohou se využít již vygenerované dlaždice z cache.

V Nette Frameworku je cache již implementována. Její použití je velmi jednoduché. Stačí pouze určit úložiště, v tomto případě bude použita složka na serveru, kterou pokud neexistuje, skript automaticky vytvoří. Poté stačí vytvořit objekt z třídy `Nette\Caching\Cache`, předat mu úložiště, s kterým bude pracovat a následně tento objekt využívat. Objekt má k práci s cache implementováno několik metod, například `save()` pro uložení dat do cache a `load()` pro načtení dat z cache. Načítání i ukládání je založeno na použití unikátního klíče, pod kterým budou data uložena [23]. V této aplikaci je klíč pro cache dlaždic překryvné vrstvy generován automaticky na základě nastaveného módu a parametrů překryvné vrstvy. Při použití módu vypočtení polohy se dlaždice necachují.

Při ukládání hodnot do cache je možné nastavit čas expirace, to znamená čas, po který konkrétní data zůstanou v cache. Ten je v této aplikaci v rozsahu 30 minut až 1 den, podle nastaveného přiblížení, viz Tabulka č. 13. Číselná hodnota úrovně přiblížení znamená hodnotu podle map Google. V závorce je poté vysvětlení úrovně přiblížení vztažené na oblast.

Úroveň přiblížení	Doba uložení v cache
0 – 9 (celá planeta až úroveň krajů)	1 den
10 (konkrétní kraj)	16 hodin
11, 12 (úroveň okresů)	8 hodin
13 (úroveň většího města, například Hradec Králové)	4 hodiny
14, 15 (úroveň městské části)	2 hodiny
16, 17, 18 (úroveň bloku budov)	1 hodina
19, 20, 21 (úroveň ulic)	30 minut

Tabulka č. 13 Čas expirace cache podle úrovně přiblížení mapy

5.2.2.2 Optimalizace skriptu

Další metodou optimalizace, která byla použita, je samotná optimalizace kódu části aplikace, která se stará o generování dlaždic překryvné vrstvy.

K optimalizaci kódu byly použity dva způsoby – optimalizace získání dat z databáze a optimalizace tvorby obrázků v PHP.

Optimalizace získání dat z databáze byla provedena pomocí odstínění od Nette databáze, která všechny výsledky automaticky vrací v objektech Row, a také pomocí vybrání pouze sloupců potřebných pro tvorbu překryvné vrstvy. Nahrazením Nette databáze za PDO a nahrazením objektů asociativním polem došlo k výraznému zrychlení aplikace. Díky této optimalizaci bylo dosaženo snížení času potřebného pro vygenerování překryvné vrstvy přibližně na desetinu a zároveň bylo sníženo zatížení serverové operační paměti.

Poslední kódová optimalizace, která pomohla ještě více snížit čas potřebný k vygenerování překryvné vrstvy, bylo použití přímo funkcí PHP ke generování obrázku, místo třídy Image z Frameworku Nette.

5.2.2.3 Zjištění polohy uživatele

Zjištění polohy uživatele není optimalizace kódu, která by přímo zrychlovala načtení konkrétních dlaždic. Ve výsledku ale může pomoci hodně, protože první načtení nebude muset načítat dlaždice pro celou republiku, což je nastaveno v případě, že uživatel polohu nepovolí, nebo prohlížeč tuto funkcionalitu nemá, ale pouze dlaždice okolí zjištěné polohy uživatele. Vzhledem k tomu, že rychlost vykreslení dlaždice je přímo závislá na počtu přístupových bodů na dlaždici, není tato optimalizace zanedbatelná.

Funkce je založena na použití HTML lokalizačního API, které funguje tak, že uživatel je požádán o poskytnutí informací o poloze. Pokud informace poskytne, tak prohlížeč pomocí svých funkcí získá polohu a vrátí jí, pokud ne tak je vyvolána obslužná událost při chybě. V případě této aplikace je ve výchozím nastavení mapa vycentrována a přiblížena tak, aby byla vidět celá Česká Republika. Pokud uživatel povolí polohu a podaří se ji zjistit, tak je mapa vycentrována na polohu uživatele a přiblížena na úroveň okresů až větších měst. Funkcionalitu lze využít v kombinaci s Google Maps například způsobem uvedeným na obrázku č. 25.

```

1.  if (navigator.geolocation) {
2.      navigator.geolocation.getCurrentPosition(
3.          function (position) {
4.              map.setCenter(new google.maps.LatLng(position.coords.latitude, position.coords.longitude));
5.              map.setZoom(12);
6.              map.overlayMapTypes.insertAt(0, new CoordMapType(new google.maps.Size(256, 256)));
7.          },
8.          function () {
9.              map.overlayMapTypes.insertAt(0, new CoordMapType(new google.maps.Size(256, 256)));
10.         });
11.     }
12.     else {
13.         map.overlayMapTypes.insertAt(0, new CoordMapType(new google.maps.Size(256, 256)));
14.     }

```

Obrázek č. 25 Použití HTML geolokace v kombinaci s Google Maps

5.2.3 Detail bodu

Aplikace umožňuje zobrazení detailu bodu v mapě, kde je možné zjistit, jestli poloha bodu je vypočtená nebo jde o konkrétní měření. Také zde jsou další informace, například typ zabezpečení, použitý kanál, datum přidání bodu, MAC adresa, SSID sítě, nadmořská výška a přesnost polohy. Některé hodnoty nejsou dostupné u všech zdrojů, zejména používaný kanál a přesnost polohy. Všechny detaily se zobrazují v info okně, což je funkcionality Google Maps API.

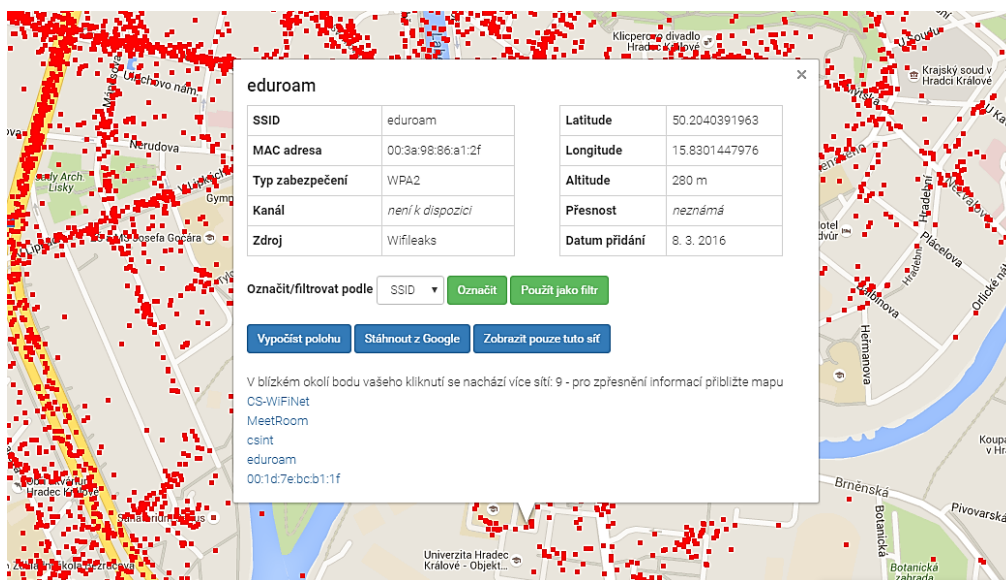
Při kliknutí do mapy jsou na server odeslány zeměpisné souřadnice bodu kliknutí, rozsah souřadnic viditelné části mapy a nastavené přiblížení. Vzhledem k tomu, že je vysoce pravděpodobné, že uživatel svým kliknutím neklikne na místo odpovídající přesným souřadnicím, tak skript vytvoří okolo bodu kliknutí okolí, které je používáno jako rozsah souřadnic, v kterém je hledán přístupový bod. Jako další parametry se používá nastavený mód. Pokud je nastavený mód vyhledávání nebo zobrazení pouze jedné sítě, bude se tento filtr aplikovat i při zobrazování detailu. Pokud by se filtr neaplikoval, mohlo by se stát, že uživatel klikne do plochy, v které nic nevidí z důvodu nastaveného filtru, ale i přesto by mu byl zobrazen detail bodu, který neodpovídá nastavenému filtru.

Ke každému nalezenému bodu z databáze je pomocí Pythagorovy věty vypočtena vzdálenost k původnímu bodu kliknutí. Zobrazený detail odpovídá nejbližšímu nalezenému bodu k bodu kliknutí. Počet ostatních nalezených bodů je

zobrazen v info okně detailu také. A pro 5 nejbližších bodů je v tabulce zobrazeno SSID, případně MAC adresy, v případě, že bod SSID nemá. Důvodem tohoto výpisu je možnost chyby, kdy jsou dva body vedle sebe, a uživatel chtěl na některý z nich kliknout, ale k bodu kliknutí byl blíže jiný bod. Proto by byl zobrazen detail jiného bodu než uživatel zamýšlel a v seznamu dalších tak může na požadované SSID nebo MAC adresu kliknout pro zobrazení detailu daného bodu.

Data do info okna jsou vrácena ve formátu JSON. Konkrétně jsou vráceny 3 hodnoty – samotný obsah info okna vygenerovaný na serveru, informace o úspěchu nebo neúspěchu požadavku a souřadnice bodu, jehož detail je zobrazován. Na tyto souřadnice je umístěn počátek info okna tak, aby zobrazení vycházelo z daného bodu.

V info okně se nachází ovládací prvky, pomocí kterých je možné provést akci, kterou lze iniciovat jediné na základě některého bodu. Konkrétně se jedná o akci vypočtení polohy, zobrazení pouze dané sítě, nebo požadavek pro získání dat z Google. Dále zde jsou ovládací prvky pro nastavení módu označení podle parametru a případně je možné tyto hodnoty použít jako filtr. Na obrázku č. 26 je ukázka zobrazení detailu bodu.

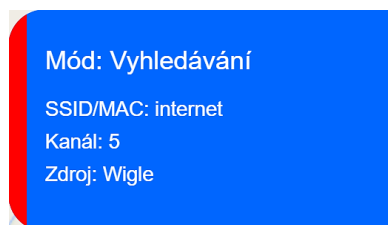


Obrázek č. 26 Okno s detailem bodu

5.2.4 Zobrazení aktuálního nastavení mapy

Aplikace umožňuje zobrazení aktuálně nastavených parametrů mapy. Funkcionalita je v grafickém rozhraní umístěna v modročerveném bloku vpravo

nahoře na mapě. Tento blok je ve výchozím stavu skrytý. Funkce slouží pouze k usnadnění orientace uživatele na webu. Na tomto bloku se uživatel dozví, který mód je právě aktivní a s jakými parametry. Informace se do bloku načítají při každé změně nastavení zobrazení. Ukázka informačního bloku je na obrázku č. 27.



Obrázek č. 27 Zobrazení aktuálního nastavení mapy

5.2.5 Vyhledávání v Google mapách

Vyhledávat lze nejen podle údajů týkajících se bodů, ale také podle místa. K tomu slouží vyhledávání míst podobně jako v klasických mapách Google. Funguje zde napovídání míst a přecházení na vybrané místo. Funkcionalitu poskytuje Google Maps API jen je nutné při načítání API přidat parametr `libraries` s hodnotou `places`, což je knihovna, v které je funkce implementována. API vrátí JSON s informacemi o poloze a dalšími údaji, přičemž o posun mapy na místo, označení místa a další se už musí postarat konkrétní aplikace.

5.3 Další funkce aplikace

Kromě získávání dat a jejich vizualizace má aplikace také další funkce. Jedná se o stránku statistik, stránku vytvořených požadavků, funkci pro export dat do CSV a sdílení nastavení mapy pomocí odkazu.

5.3.1 Statistiky

Stránka statistik slouží jako informační stránka, kde uživatel nalezne další informace o sítích z databáze. Statistiky jsou automaticky generovány každou půlnoc a jsou ukládány do databáze. To je z důvodu optimalizace, protože provádět na tak velkém množství dat výpočty při každém načtení stránky statistik by bylo velmi náročné.

Na stránce statistik lze zjistit celkový počet bodů v databázi, počet nezabezpečených bodů, změnu od předchozího dne, graf s vývojem těchto počtů

v čase, a graf s podílem zabezpečených a nezabezpečených sítí. Dále jsou zde statistiky podílů jednotlivých zdrojů dat a typů zabezpečení sítí, vždy s grafem vývoje v čase a koláčovým grafem znázorňujícím podíl.

5.3.2 Vytvořené požadavky

Další součástí aplikace je stránka vytvořených požadavků. Zde je možné nalézt seznam vytvořených požadavků pro získání dat ze zdrojů Wigle a Google vytvořených rozsahem souřadnic. Stránka se skládá ze dvou bloků, jeden pro Wigle a druhý pro Google. V každém bloku se nachází tabulka požadavků, v které jsou zobrazovány informace jako datum vytvoření požadavku, rozsah zeměpisné šířky a délky, informace o zpracování požadavku a pokud je požadavek již zpracovaný tak datum zpracování požadavku.

Dále se zde nachází mapa s grafickým zvýrazněním požadavků. Požadavky jsou v mapě vykresleny jako obdélníky nacházející se nad požadovanou plochou. Obdélníky jsou vybarveny červenou nebo zelenou barvou podle stavu zpracování – nezpracované červeně, zpracované zeleně.

Vykreslení obdélníku do Google mapy je jednoduché, stačí určit barvy, mapu, v které bude obdélník vykreslen, a počáteční a koncové souřadnice. Ukázka skriptu pro vykreslení obdélníku je na obrázku č. 28.

```
1. var rectangle = new google.maps.Rectangle({
2.   strokeColor: '#FF0000',
3.   strokeOpacity: 0.8,
4.   strokeWeight: 2,
5.   fillColor: '#00FF00',
6.   fillOpacity: 0.35,
7.   map: mapWigle,
8.   bounds: {
9.     north: 50.5,
10.    south: 50.22,
11.    east: 16.11,
12.    west: 15.83
13.  }
14. });
```

Obrázek č. 28 Skript pro přidání obdélníku do Google Mapy

5.3.3 Export do CSV

Další funkcí aplikace je export dat do souboru formátu CSV. Při exportu budou aplikovány filtry z vyhledávacího formuláře. Viditelný rozsah souřadnic je při

exportu ignorován a jsou exportována skutečně všechna data z databáze, která odpovídají nastavenému filtru.

Nejprve jsou zpracovány parametry, podle kterých bude filtrováno, a vygenerován název souboru, který odpovídá aktuálnímu datu, času a nastaveným parametrům. Na první řádek souboru je zapsána hlavička s vysvětlivkami sloupců a poté jsou podle parametrů získány body. Body jsou po tisíci zapisovány do souboru. Nakonec je soubor uložen na serveru do složky temp a cesta k němu předána prohlížeči, který iniciuje stažení souboru do počítače uživatele.

Formát CSV dokáží zpracovat programy pro práci s tabulkami, například Excel [20]. Vytvoření souboru funguje stejně jako vytváření jakéhokoliv jiného souboru v PHP. Slouží k tomu funkce `fopen`, které je předán název souboru a metoda otevření. Pokud soubor neexistuje tak je automaticky vytvořen. Pro zápis CSV je potřeba sestavit pole hodnot. Poté stačí použít metodu `fputcsv`, které je předán soubor, pole hodnot a oddělovač hodnot. Po zápisu je nutné soubor uzavřít funkcí `fclose`. Ukázka vytvoření CSV souboru v PHP je na obrázku č. 29.

```
1. $file = fopen("soubor.csv", "w");  
2. fputcsv($file, array("hodnota 1. sloupce", "hodnota 2. sloupce", "hodnota 3. sloupce"), ";");  
3. fclose($file);
```

Obrázek č. 29 Vytvoření CSV v PHP

5.3.4 Sdílení pomocí odkazu

Pro usnadnění sdílení informací je v aplikaci implementována možnost sdílet nastavení mapy pouhým zkopírováním odkazu. Nastavení mapy včetně umístění, přiblížení, použitého módu a parametrů je přidáváno do URL i přesto, že tyto požadavky probíhají asynchronně. Díky tomu je možné zkopírovat obsah adresního řádku v prohlížeči a poslat ho jako odkaz někomu jinému. Při načtení aplikace, kde v URL jsou nastavené nějaké parametry, jsou tyto parametry zpracovány a aplikovány. Díky tomu bude aplikace otevřena na stejném místě a se stejným filtrem jako u osoby, která odkaz posílala.

5.4 Problémy při implementaci

Při implementaci bylo nutno vyřešit některé problémy. Několik problémů nastalo hned v úvodní části aplikace. Například problém s využitím lokalizačních služeb Google, které neposkytují jiné informace než polohu a její přesnost, a k získání této polohy je nutno předat alespoň 2 existující MAC adresy. Dalším problémem implementace byly rozdíly v datech a omezení Wigle API. V části uživatelského rozhraní nastal problém s rychlostí vykreslování dlaždic pro překryvnou vrstvu mapy.

Problém s využitím lokalizačních služeb tkví v tom, že je nutné předat alespoň 2 MAC adresy a sílu jejich signálu, podle kterých lokalizační služby Google určí polohu. Ideální je použít bod, jehož poloha je požadována, jako bod s velkou silou signálu, a bod, který je v blízkosti požadovaného simulovat jako bod s nízkou silou signálu. Problém byl vyřešen tím, že u požadavků, které jsou zadané rozsahem souřadnic pro zdroj Google, jsou nejprve data získána z Wigle, protože pokud by na vybrané ploše nebyly žádné body tak z Google není možné nic získat. Proto jsou data získána z Wigle a poté je pro každý získaný bod vytvořen požadavek na Google.

Proto také bylo nutné implementovat funkci blokování požadavků, kdy při vytvoření požadavku, zadaného rozsahem zeměpisných souřadnic, pro zdroj Google, jsou vytvořeny požadavky dva: jeden pro Wigle a jeden pro Google. Požadavek pro Google je blokován do té doby, než je zpracován požadavek pro Wigle. Problém s neposkytnutím jiných informací než polohy byl vyřešen tak, že z bodu, o kterém byly požadovány informace, jsou kopírovány informace jako SSID sítě, použitý kanál a další, a jako poloha bodu je nastavena poloha získaná z Google.

Rozdíly v datech se týkají především způsobu určení zabezpečení. U dat z Wifileaks je zabezpečení udáno jako číselná hodnota, která znamená určitý typ zabezpečení, ale z Wigle je informace o zabezpečení rozdělena do několika parametrů. Byl tedy naprogramován algoritmus, který hodnoty sjednocuje. U hodnot získaných z Wigle se jedná o příznak freenet a paynet a hodnotu vrácenou v atributu wep. Příznak paynet a freenet je jedna z hodnot {„Y“, „N“ nebo „?“}, hodnoty v atributu wep jsou popsány v tabulce č. 14. U většiny hodnot stačí hodnota

atributu wep, ale pokud je hodnota atributu wep „?“ nebo „N“, tak je nutno dále rozhodnout.

Hodnota atributu wep	Význam hodnoty (použité zabezpečení)
?	neznámé
Y	WEP
N	Není WEP, otevřená síť
W	WPA1
2	WPA2

Tabulka č. 14 Význam atributu wep získaného z Wigle

Pokud hodnota atributu wep je „?“ a zároveň příznak freenet nastaven na „Y“, pak je hodnota zabezpečení nastavena jako „otevřená síť“. V opačném případě je zabezpečení klasifikováno jako neznámé. Pokud je hodnota atributu wep „N“, příznak freenet je jiná hodnota než „N“ a zároveň síť není označena jako placená, tak je hodnota zabezpečení nastavena jako otevřená síť, jinak neznámá.

Omezení Wigle API jsou velmi přísná a z toho důvodu jsou v aplikaci funkcionality pro minimalizaci požadavků. Pokud nový požadavek určený zeměpisnými souřadnicemi zasahuje do již existujícího požadavku, tak je požadavek rozdělen tak, aby byly přidány pouze části, o které ještě žádáno nebylo. Také je zde analýza překryvné vrstvy, která slouží k odhadu hustoty bodů v dané ploše. Podle hustoty je velký rozsah souřadnic rozdělen na několik menších ploch.

V uživatelském rozhraní nastal problém s rychlostí vykreslování dlaždic překryvné vrstvy. Rychlost vykreslování byla optimalizována několika způsoby. Konkrétně vynecháním databázové vrstvy Nette. Dále také na překryvnou vrstvu stačí z databáze získávat pouze hodnoty polohy, SSID sítě a MAC adresy bodu. Tyto hodnoty jsou předány vykreslovací metodě v asociativním poli místo objektu, čímž se náročnost snížila. Samotná vykreslovací metoda byla optimalizována tím, že místo třídy Image z Nette Frameworku, která usnadňuje práci s obrázky, bylo použito metod, které pro práci s obrázky poskytuje jazyk PHP. Dále je zde také snaha minimalizovat plochy, které budou generovány. Toho je docíleno tím, že uživatel může povolit svou polohu prohlížeči a mapa bude přiblížena na zjištěnou polohu. Jednotlivé dlaždice překryvné vrstvy jsou také ukládány do cache.

6 Zhodnocení výsledků

Porovnávána budou data sítě hkfree.org. Nejprve bude určena metoda porovnávání dat, poté výpočet ukázán na konkrétním případě a následně již shrnutí výsledků pro vybrané porovnávané body. Nakonec bude zhodnocení všech získaných výsledků.

6.1 Metoda porovnání zdrojů dat

K porovnání přesnosti zdrojů dat byla použita data sítě hkfree.org. Je žádoucí porovnávat data, která jsou k dispozici ze všech zdrojů, takže nejprve z poskytnutého souboru všech SSID byly vyfiltrovány pouze ty záznamy, ke kterým je k dispozici MAC adresa ze zdroje Wifileaks. K těmto přístupovým bodům následně byly přiřazeny známé polohy podle druhého souboru obsahujícího polohy bodů. V analyzovaných datech mohou některé přístupové body mít stejnou polohu, například pokud je jich více ve stejné budově. Porovnání se bude týkat pouze několika vybraných bodů, o kterých se podařilo získat data ze všech zdrojů.

Pro usnadnění porovnání byla do aplikace přidána funkcionality, která vypočte vzdálenosti všech bodů s určitou MAC adresou od zadaných zeměpisných souřadnic a seřadí výsledky podle vzdálenosti vzestupně, aby nahoře byl nejpřesnější výsledek. Výpočet vzdálenosti pro seřazení je prováděn pomocí Pythagorovy věty. Tento výpočet sice pro účely seřazení postačuje, protože tak malé plochy téměř žádné zaoblení nemají a proto je lze považovat za rovinu. Ale je nepřesný a hodnota rozdílu zeměpisných souřadnic pro člověka není moc vypovídající. Z toho důvodu byla implementována metoda pro výpočet vzdálenosti v metrech. K výpočtu vzdálenosti po kulové ploše slouží takzvaný Haversinův vzorec.

6.1.1 Haversinův vzorec

Haversinův vzorec slouží k výpočtu vzdálenosti mezi dvěma body na kulové ploše. Výpočet není přesný, protože považuje planetu Zemi za hladkou kouli – ignoruje jakékoli kopce [24]. To v případě této aplikace není problém, protože důležitá je přímá vzdálenost vzdušnou čarou.

Kód metody pro výpočet vzdálenosti vychází z algoritmu popsaného na webu [24], za poloměr země je zde považována hodnota 6 371 000 metrů. Kód algoritmu je na obrázku č. 30. Výpočet je nutné provádět v radiánech.

```
1.  /**
2.   * @return int
3.   */
4.  public function getDistanceInMetres() {
5.      $r = self::EARTH_RADIUS;
6.      $dlat = $this->toRad($this->getDeltaLat());
7.      $dlon = $this->toRad($this->getDeltaLon());
8.      $a = sin($dlat/2) * sin($dlat/2) +
9.          cos($this->toRad($this->lat_start)) * cos($this->toRad($this->lat_end))
10.         * sin($dlon / 2) * sin($dlon / 2);
11.      $c = 2 * atan2(sqrt($a),sqrt(1-$a));
12.      $d = $r * $c;
13.      return $d;
14.  }
15.  /**
16.   * @param float $degrees
17.   * @return float
18.   */
19.  public function toRad($degrees) {
20.      return $degrees * pi() / 180;
21.  }
```

Obrázek č. 30 Implementace Haversinova vzorce v PHP

6.2 Porovnání přesnosti zdrojů

Porovnání přesnosti probíhá pro všechny body stejně, proto stačí algoritmus výpočtu ukázat na jednom bodu. U ostatních bodů zde budou pouze výsledky. Hodnoty, z kterých bylo počítáno, a soubory s daty jsou přiloženy na CD.

6.2.1 Porovnání pro jeden bod

Chyba jednotlivých zdrojů pro každý bod je počítána jako vzdálenost od reálné polohy. Pokud v databázi pro jednu MAC adresu je více záznamů ze stejného zdroje (týká se zdrojů Wigle a Google), je vypočtena minimální, maximální a průměrná chyba pro tyto zdroje. Pokud je k dispozici vypočtená poloha z Wigle, jako průměrná chyba je určena vzdálenost vypočtené polohy od reálné, jinak je chyba vypočtena jako aritmetický průměr chyb všech bodů získaných pro MAC adresu z daného zdroje. Směrodatná hodnota porovnání bude právě průměrná chyba.

Přesné hodnoty chyby jsou aktuální pouze v době porovnání. Z důvodu průběžné aktualizace dat se naměřené výsledky mohou lišit od aktuálních.

Příklad výpočtu s MAC adresou 00:0b:6b:81:c4:09: známé souřadnice bodu jsou latitude 50.17732 a longitude 15.84642. V databázi je celkem 7 záznamů pro

tento přístupový bod, 1 z Wifileaks, 4 z Wigle a 2 z Google. Viz Obrázek č. 31 Body v databázi pro určitou MAC adresu.

MAC: Reálná latitude: Reálná longitude:

Chyba Wifileaks	Chyba Wigle MIN	Chyba Wigle MAX	Chyba Wigle AVG	Chyba Google MIN	Chyba Google MAX	Chyba Google AVG
1949.6930536412	101.62700748473	2645.2854086261	1039.8456356793	58.436370046527	113.9789552706	86.207662658563

ID	Zdroj	latitude	longitude	ssid	mac	vypočtená?	vzdálenost od realne	vzdálenost v metrech
3390674	Google	50.1778439	15.8463554	hrbitov2.hkfree.org	00:0B:6B:81:C4:09	0	0.0005278677580581553	58.436370046527
3390570	Wigle	50.17776871	15.8451767	hrbitov2.hkfree.org	00:0B:6B:81:C4:09	0	0.001321792553353878	101.62700748473
3390673	Google	50.1778553	15.845055	hrbitov2.hkfree.org	00:0B:6B:81:C4:09	0	0.0014662097701198628	113.9789552706
3390572	Wigle	50.18061829	15.85078812	hrbitov2.hkfree.org	00:0B:6B:81:C4:09	0	0.0054734988132346795	480.89262796604
3390569	Wigle	50.18050385	15.83431911	hrbitov2.hkfree.org	00:0B:6B:81:C4:09	0	0.012512731101347136	931.57749864031
1624622	Wifileaks	50.1939146667	15.8375772593	hrbitov2.hkfree.org	00:0b:6b:81:c4:09	0	0.018803643954605737	1949.6930536412
3390571	Wigle	50.19797516	15.82798576	hrbitov2.hkfree.org	00:0B:6B:81:C4:09	0	0.027684956908094565	2645.2854086261

Obrázek č. 31 Body v databázi pro určitou MAC adresu

Z těchto údajů byly vypočteny chyby pro jednotlivé zdroje, u Wifileaks chyba činí přibližně 1950 metrů, pro Wigle je minimální chyba 102 metrů, maximální 2645 metrů a průměrná 1040 metrů a pro data z Google činí minimální chyba 58 metrů, maximální 114 metrů a průměrná 86 metrů. Z toho vyplývá, že pro tento konkrétní bod byla nejpřesnější data získána z lokalizačních služeb Google.

6.2.2 Výsledky porovnání všech bodů

Některé body nebylo možné získat ze zdroje Google s dostatečnou přesností, a některé se nepodařilo získat ani ze zdroje Wigle. Tyto body jsou z porovnání vyjmuty. Porovnávají budou průměrné chyby, čím nižší průměrná chyba, tím přesnější data z daného zdroje jsou. Porovnání pro vybrané body je v tabulce č. 15.

MAC adresa	SSID síť	Chyba Wifileaks [m]	Chyba Wigle [m]	Chyba Google [m]
00:0b:6b:86:5b:6c	andre5g-s1.kocourkov.hkfree.org	174	801	343
00:0c:42:2b:38:86	fugas2.hkfree.org	252	298	295
00:0b:6b:81:c4:09	hrbitov2.hkfree.org	1950	1040	86
00:0b:6b:37:03:29	hrubinova.kocourkov.hkfree.org	148	784	301
00:0c:42:8d:f3:bf	hrubinova5g-s2.kocourkov.hkfree.	213	491	466
00:0c:42:8d:f4:03	hrubinova5g-s3.kocourkov.hkfree.	85	759	249
00:0c:42:8e:d6:69	hrubinova5g-s6.koc.hkfree.org	992	882	507
90:a4:de:7d:8d:b9	lesopark.hkfree.org	50	451	189
00:0c:42:8e:b0:8d	list-5G-S.hkfree.org	735	682	94
d4:ca:6d:11:a8:95	male.s2.hkfree.org	115	254	209
00:1b:b1:04:be:96	navi2.stezirky.hkfree.net	434	2984	321
00:1b:b1:01:6c:e9	ne2g.ap.plch.hkfree.org	1182	1297	1200
d4:ca:6d:30:2c:57	piletice-350.hkfree.org	901	819	286
d4:ca:6d:30:0b:0b	s1.bretislavova.hkfree.org	166	119	203
00:0b:6b:df:2a:83	s1.lesopark.hkfree.org	53	248	105
d4:ca:6d:30:89:a1	s4.bretislavova.hkfree.org	47	14	56
4c:5e:0c:11:1f:b8	severni.s1.hkfree.org	202	575	201
00:1b:b1:05:52:bf	severni.s3.hkfree.org	612	536	181
00:0b:6b:df:ba:d8	sokolik.s1.hkfree.org	119	115	118
00:0b:6b:df:ad:ce	sokolik.s2.hkfree.org	436	145	140
00:1b:b1:04:c7:bf	sramkuv.statek.hkfree.org	8	48	108
00:0c:42:49:1f:54	StaraVoda.hkfree.org	554	1053	278
00:0c:42:23:1f:3c	sw2g.ap.plch.hkfree.org	1527	1282	1184
00:1b:b1:00:b1:4a	msplaciceS1.hkfree.org	611	1033	40614

Tabulka č. 15 Porovnání zdrojů pro vybrané přístupové body

U některých bodů lze vidět, že chyby zdrojů se liší pouze v řádu jednotek metrů (00:0b:6b:df:ba:d8 - sokolik.s1.hkfree.org), naopak u jiných bodů se chyba pohybuje i v řádech kilometrů (00:1b:b1:00:b1:4a - msplaciceS1.hkfree.org). U posledního jmenovaného bodu dosahuje chyba ze zdroje Google extrémní hodnoty a to více než 40km. Tento bod by mohl být považován za chybu.

6.2.3 Porovnání zdrojů dat

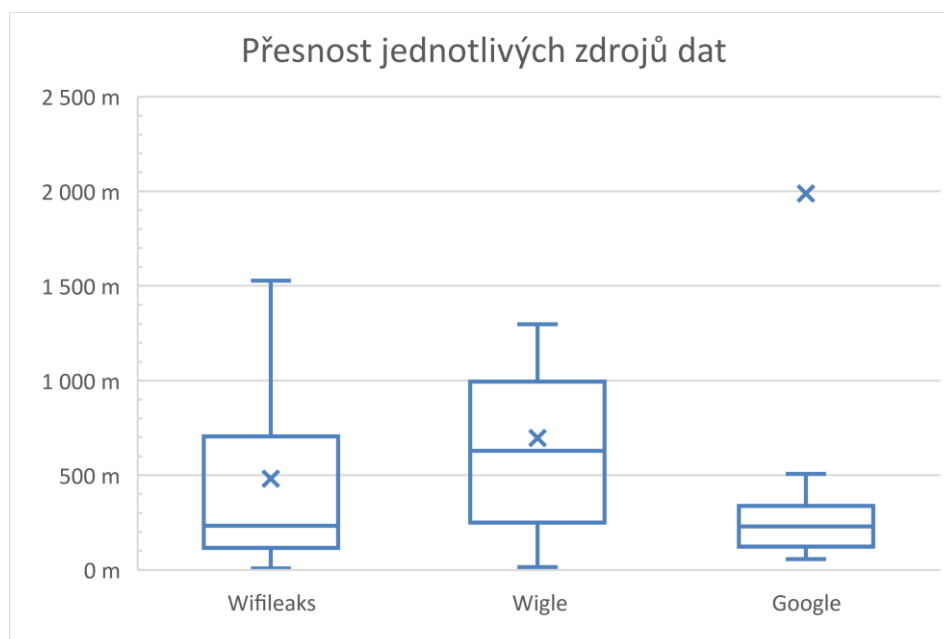
Počty nejlepších a nejhorších výsledků pro všechny zdroje se nachází v tabulce č. 16.

Zdroj dat	Nejlepší	Nejhorší
Wifileaks	11x	8x
Wigle	3x	12x
Google	10x	4x

Tabulka č. 16 Porovnání zdrojů dat

Z tabulky porovnání zdrojů vyplývá, že zdroj Wifileaks podal nejlepší výsledek 11krát, Google 10krát. Zároveň ale data z Wifileaks byly 8krát nejméně přesná. Z toho vyplývá, že data z Google jsou přesnější. Data ze zdroje Wigle byly nejpřesnější pouze ve 3 případech a zároveň ve 12 případech byla nejhorší.

Porovnání konkrétních hodnot chyby se nachází v krabicovém grafu na obrázku č. 32. Rozsah osy Y grafu byl snížen, aby byly patrné rozdíly mezi jednotlivými přístupy (mimo rozsah Y se nachází jediný bod z Google s chybou 40km). Konkrétní vypočtené hodnoty jsou v tabulce č. 17.



Obrázek č. 32 Graf porovnání přesnosti zdrojů

Z grafu je vidět, že medián zdroje Google a Wifileaks je velmi podobná hodnota. Střední hodnota zdroje Google je zde velmi ovlivněna extrémní chybou v jednom případě. Také je vidět, že data ze zdroje Google nemají velký rozptyl chyby,

to může být způsobeno tím, že porovnání bylo provedeno pouze u sítí, které lokalizační služby Google dokázaly lokalizovat na základě předaných informací o WiFi sítích.

Zdroj	Wifileaks	Wigle	Google
Minimum	8	14	56
Maximum	1950	2984	40614
První kvartil (1Q)	116	250	123
Medián (2Q)	233	629	229
Třetí kvartil (3Q)	704	995	338
Střední hodnota	482	696	1989

Tabulka č. 17 Porovnání přesnosti zdrojů (uvedena chyba v metrech)

V tabulce je vidět, že zdroje Wifileaks a Google jsou podobně přesné (medián 233 a 229 metrů). Výhra Google se potvrzuje třetím kvantilem, z čehož vyplývá, že v datech z Google je u 75 % chyba menší než 338 metrů, u zdroje Wifileaks je tato hodnota 704 metrů.

7 Závěry a doporučení

Data ze zdroje Google, jsou pro testovaná data překvapivě nejpřesnější a to i přes to, že tato aplikace nepoužívá přímo WiFi databáze společnosti Google, jako u ostatních zdrojů, ale pouze lokalizačních služeb Google. Pro získání všech potřebných dat je ale nutné využít informace získané z jiného zdroje.

Kvůli velkým omezením zdroje Wigle a zároveň nejhorším výsledkům z porovnávaných zdrojů nelze zdroj Wigle doporučit pro samostatné použití. Vzhledem k tomu, že data z Google byla získávána na základě již existujících bodů z databáze, tak nelze vyloučit možnost, že přesnost dat z Google je vylepšená právě díky velkému množství bodů ze zdroje Wigle.

Díky informacím týkajícím se přístupových bodů, které jsou v databázi, lze aplikaci rozšířit o další funkcionality používající data jiným způsobem. Další možnosti vylepšování aplikace týkající se zejména efektivity aplikace v získávání dat jsou například přidání dalších zdrojů dat nebo snížení omezení, která při získávání dat nastávají pomocí více paralelních miniaplikací, která by data získávala.

Mezi možná rozšíření aplikace patří přidání dalších módů zobrazení – například zobrazení pouze volných sítí. Další způsoby využití dat, jako například určení polohy uživatele webu, který by vyplnil názvy WiFi sítí, které má v dosahu a jejich sílu signálu, a aplikace by se na základě vlastních dat v databázi pokusila určit polohu uživatele. Dále také přidání dalších zdrojů dat nebo metod získávání dat – konkrétně například přidání možnosti vytvořit bod uživatelem webu - zde by bylo nutné zavést kontrolu validity a přesnosti dat.

Aplikaci je možné vylepšit také z hlediska optimalizace. Časy, po které se do cache ukládají dlaždice překryvné vrstvy, by mohly být delší. K tomu by bylo nutné vždy po získání nových dat do databáze zneplatnit cache pro všechny dlaždice překryvné vrstvy, na kterých se nové body budou nacházet.

Užitečné by bylo také vylepšení, které by odstranilo nebo snížilo omezení Wigle API. Možným řešením je vytvoření více miniaplikací, jejichž účelem by bylo pouze získávat data, a jejich rozmístění na více serverech, nebo vytvoření více uživatelských účtů, pod kterými by data byla získávána. Možné by bylo tyto metody také zkombinovat.

Vzhledem k tomu, že bezdrátové sítě jsou nejčastěji využívány na mobilních zařízeních, jako jsou například chytré telefony, tak stojí za zmínku možnost vytvořit mobilní aplikaci. Užitečnost aplikace by však velmi závisela na způsobu využití dat o přístupových bodech.

Jako poslední vybraná možnost rozšíření je rozšíření API, které zatím slouží pouze pro import MAC adres, jejichž informace jsou požadovány, a pro export přístupových bodů z databáze, například o možnost získání konkrétních dat pro určitou MAC adresu, nebo podle jiných parametrů. K tomu by bylo vhodné v aplikaci přidat další rozhraní, v kterém by se data nezobrazovala do mapy, ale v tabulce.

Cílem práce bylo vytvořit aplikaci, která bude využívat databáze Wifileaks, Wigle a lokalizační služby Google, s tím, že uživatel bude mít možnost požádat o získání dat. Dále také byla vytvořena vizuální část aplikace, kde uživatel může vidět přístupové body v mapě a ovlivnit způsob jejich zobrazení.

Pro tvorbu aplikace byl použit PHP framework Nette, ve vizuální části pak Mapy Google. Jak mapy, tak aplikace využívají JavaScript, v kombinaci s knihovnou JQuery, proto aplikace pro svůj běh požaduje povolený JavaScript v prohlížeči.

Podařilo se vytvořit aplikaci, která dokáže na pozadí získávat data na základě uživatelem vytvořených požadavků, které aplikace sama zpracuje. Aplikace také obsahuje vizuální část, která umožňuje různé módy zobrazení dat, zobrazení detailu bodů, filtrování dat, export dat do CSV a další funkcionality.

Z porovnání zdrojů vyplynulo, že nejpřesnějším zdrojem porovnávaných dat byly lokalizační služby Google a nejméně přesným zdrojem byl Wigle, který také pro získání dat má největší omezení.

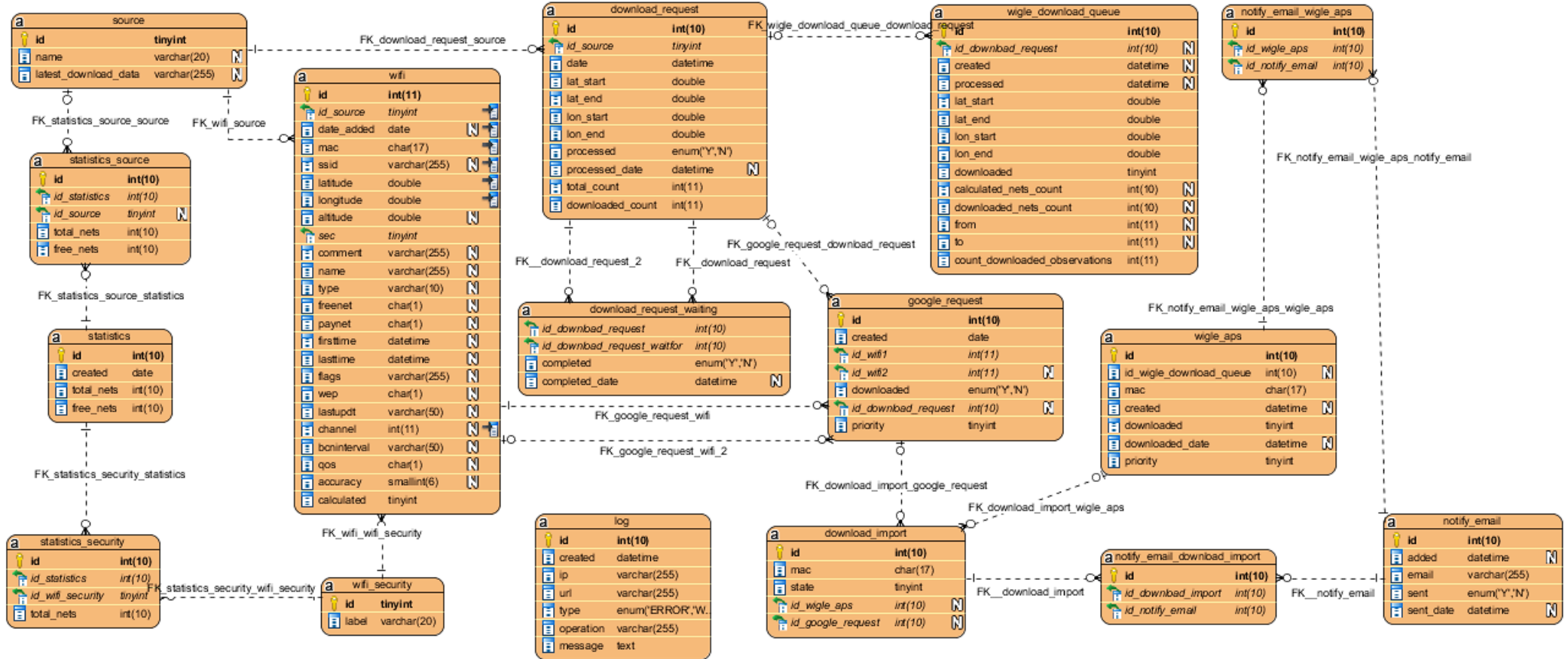
8 Seznam použité literatury

1. BUCZKOWSKI, A. Geoawesomeness. *Location-Based Services – Technologies* [online]. 2011 - 2012 [cit. 2016-03-23]. Dostupné z: <http://geoawesomeness.com/knowledge-base/location-based-services/location-based-services-technologies/>
2. *Hledání Wi-Fi hotspotů* [online]. [cit. 2016-03-13]. Dostupné z: <http://www.windowsphone.com/cs-cz/how-to/wp8/connectivity/find-wi-fi-hotspots>
3. *How Google-and everyone else-gets Wi-Fi location data.* [online]. 16. 11. 2011 [cit. 2015-06-20]. Dostupné z: <http://www.zdnet.com/article/how-google-and-everyone-else-gets-wi-fi-location-data/>
4. *Wifileaks* [online]. [cit. 2016-01-16]. Dostupné z: <http://www.wifileaks.cz/>
5. *Frequently Asked Questions* [online]. 2001 - 2016 [cit. 2016-03-17]. Dostupné z: <https://wagle.net/faq>
6. *What is Symfony* [online]. [cit. 2016-01-16]. Dostupné z: <https://symfony.com/what-is-symfony>
7. *Přehled a vývoj PHP frameworků* [online]. 28. 3. 2008 [cit. 2016-01-16]. Dostupné z: <http://www.root.cz/clanky/prehled-a-vyvoj-php-frameworku/>
8. *About* [online]. 2006 - 2016 [cit. 2016-01-16]. Dostupné z: <http://framework.zend.com/about/>
9. *Rychlý a pohodlný vývoj webových aplikací v PHP* [online]. 2008 - 2016 [cit. 2016-01-16]. Dostupné z: <https://nette.org/>
10. *Formuláře* [online]. 2008 - 2016 [cit. 2016-01-16]. Dostupné z: <https://doc.nette.org/cs/2.3/forms>
11. *1. díl – Úvod do Nette frameworku pro PHP* [online]. 2015 [cit. 2015-01-16]. Dostupné z: <http://www.itnetwork.cz/uvod-do-php-frameworku-nette>
12. *Markers* [online]. 10. 5. 2015 [cit. 2016-01-16]. Dostupné z: <https://developers.google.com/maps/documentation/javascript/markers>

13. *Info windows* [online]. 1. 3. 2016 [cit. 2016-03-17]. Dostupné z: <https://developers.google.com/maps/documentation/javascript/examples/infowindow-simple>
14. *Events* [online]. 10. 5. 2015 [cit. 2016-01-16]. Dostupné z: <https://developers.google.com/maps/documentation/javascript/events>
15. *Overlay Map Types* [online]. 1. 3. 2016 [cit. 2016-03-17]. Dostupné z: <https://developers.google.com/maps/documentation/javascript/maptypes#OverlayMapTypes>
16. Oracle. *MySQL The World's Most Popular Open Source Database* [online]. 2015 [cit. 2016-03-23]. Dostupné z: <http://www.oracle.com/us/products/mysql/overview/index.html>
17. MySQL. *MySQL* [online]. 2016 [cit. 2016-03-23]. Dostupné z: <http://www.mysql.com/>
18. LINDLEY, C. *JQuery Kuchařka programátora*. Překlad Ondřej BAŠE. Computer Press, a.s. 2010 [cit. 2016-03-23]. ISBN 978-80-251-3152-7.
19. ASLESON, R. a N. T. SHUTTA. *AJAX Vytváříme vysoce interaktivní webové aplikace*. Brno: Computer Press, a.s. 2006 [cit. 2016-03-23]. ISBN 80-251-1285-3.
20. VRÁNA, J. *1001 tipů a triků pro PHP*. Brno: Computer Press, a.s. 2012 [cit. 2016-03-23]. ISBN 978-80-251-2940-1.
21. KRŮŽ, P. *Automated WiFi-based Location and Visualization of Wireless Network*. 2014 [cit. 2016-03]. DOI 10.3233/978-1-61499-434-3-432.
22. ULLMAN, L. E. *PHP: pokročilé programování pro world wide web*. Překlad Martin BARTEL. Praha: SoftPress, 2002. ISBN: 80-86497-36-4.
23. Nette Framework. *Cache* [online]. 2008 - 2016 [cit. 2016-03-24]. Dostupné z: <https://doc.nette.org/cs/2.3/caching>
24. VENESS, C. Movable Type Scripts. *Calculate distance, bearing and more between Latitude/Longitude points* [online]. 2012 - 2014 [cit. 2016-04-02]. Dostupné z: <http://www.movable-type.co.uk/scripts/latlong.html>

9 Přílohy

Struktura databáze



```
1. CREATE TRIGGER `download_request_after_update` AFTER UPDATE ON `download_request` FOR EACH ROW BEGIN
2. IF NEW.downloaded_count = NEW.total_count THEN
3.     UPDATE download_request_waiting SET completed_date = NOW(),completed='Y' WHERE id_download_request_waitfor = NEW.id;
4. END IF;
5. END
```

Trigger pro označení požadavku jako dokončený

```
1. CREATE TRIGGER `wicle_download_queue_after_update` AFTER UPDATE ON `wicle_download_queue` FOR EACH ROW BEGIN
2. IF NEW.count_downloaded_observations = NEW.downloaded_nets_count THEN
3.     UPDATE download_request SET downloaded_count = downloaded_count + 1 WHERE id = NEW.id_download_request;
4. END IF;
5. END
```

Trigger pro zvýšení počtu již získaných dat

Obsah přiloženého CD

1. Zdrojové kódy aplikace v souboru ZIP
2. Export databáze (struktura, struktura a data)
3. Analyzované soubory
4. Návod k nasazení aplikace
5. Text práce v PDF
6. Soubor TSV s daty Wifileaks

Oskenované zadání práce

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2015/2016

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Vlček Roman	Březová 53, Meziměstí - Březová	I1301380

TÉMA ČESKY:

Mapa WiFi sítí

TÉMA ANGLICKY:

Wireless Networks Map

VEDOUcí PRÁCE:

Ing. Pavel Kříž, Ph.D. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Cíl:

Navrhnout a implementovat webovou aplikaci, která zobrazí polohy resp. pokrytí WiFi přístupových bodů v mapě. Data budou čerpána z dostupných databází (Wigle, WifiLeaks, ...) a lokalizačních služeb (Google,...).

Osnova:

1. Úvod
2. Přehled zdrojů pokrytí
3. Analýza
4. Návrh a implementace
5. Zhodnocení výsledků
6. Závěr

SEZNAM DOPORUČENÉ LITERATURY:

<https://doc.nette.org/cs/2.3/>

<https://developers.google.com/maps/>

Kriz, P.: Automated WiFi-based Localization and Visualization of Wireless Network, DOI 10.3233/978-1-61499-434-3-432

Podpis studenta:

Vlček

Datum:

13. 10. 15

Podpis vedoucího práce:

Kříž

Datum:

13. 10. 15