

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informační technologie

Locate.io – lokalizace Wi-Fi zařízení

Matěj Půhoný
Pardubický kraj

Pardubice 2021

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informační technologie

Locate.io – lokalizace Wi-Fi zařízení

Locate.io – Wi-Fi devices localization

Autoři: Matěj Půhoný
Škola: DELTA – Střední škola informatiky a ekonomie, s.r.o.
Ke Kamenci 151, 503 03 Pardubice
Kraj: Pardubický kraj
Konzultant: RNDr. Jan Koupil, Ph.D.

Pardubice 2021

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Pardubicích dne:

.....

Matěj Půhoný

Poděkování

Děkuji svému konzultantovi RNDr. Janu Koupilovi, Ph.D. za odborné vedení při tvorbě tohoto projektu, podmětné připomínkování a korekturu této práce.

Anotace

Cílem této práce SOČ je vytvořit systém, který umožňuje sledovat polohu uživatelů Wi-Fi sítě na základě využití pasivního odposlechu okolní komunikace Wi-Fi sítě. Systém lze využít například jako nástroj ke sledování docházky bez nutnosti instalace aplikace do jednotlivých zařízení. Systém využívá několika monitorujících majáků a serveru pro výpočet polohy zařízení.

Klíčová slova

Lokalizace, Wi-Fi sítě, docházkový systém, webová aplikace

Annotation

The goal of this SOČ project is to create a system that provides the possibility of monitoring positions of Wi-Fi client devices based on passive listening of surrounding Wi-Fi communications. The system can be used as a tool for time tracking employee's attendance without the requirement of installing any apps on tracked devices. The system uses multiple packet monitoring beacons and a server for computing positions of the devices.

Keywords

Localization, Wi-Fi networks, attendance system, web app

Obsah

1	Slovník pojmů.....	7
2	Úvod.....	8
3	Architektura systému	9
3.1	Majáky.....	9
		10
3.2	Backend.....	10
3.3	Výpočetní server	10
3.4	Databáze.....	11
3.5	Webová aplikace	11
4	Metody výpočtu polohy zařízení	12
4.1	Fingerprinty.....	12
4.2	Trilaterace	12
4.3	Neuronová síť.....	14
5	Zabezpečení	14
5.1	Server	14
5.2	Zabezpečení přístupu	14
5.3	Validace dat zaslaných pomocí REST API.....	15
5.4	Ochrana proti DDOS.....	15
6	MĚŘENÍ ÚBYTKU síly Wi-Fi signálu.....	15
6.1	Vlivy úbytku signálu.....	15
6.1.1	Vlastní měření.....	16
6.2	Náhodné MAC adresy u mobilních zařízení	17
7	Datový model systému.....	19
8	Webové UI/UX – funkcionality.....	20
8.1	Správa organizace	20
8.2	Plány a lokalizace.....	21
8.3	Záznamy stavu majáků a historie polohy zařízení	21
8.4	Zobrazení lokalizace a nastavení lokalizace	22
9	Výsledky lokalizací.....	23
9.1	Venkovní prostředí.....	23
9.2	Vnitřní prostředí	24
10	Závěr.....	25

11	Seznam obrázků a tabulek	29
12	Seznam příloh	30
12.1	Příloha 1: Snímky z webové aplikace	30
12.2	Příloha 2: Ukázka designu.....	30

1 SLOVNÍK POJMŮ

- 1 **Raspberry Pi** – Malý jednodeskový počítač, výkonnostně srovnatelný se slabšími stolními počítači. Určený pro méně náročné linuxové distribuce. [1]
- 2 **Python** – Vysokoúrovňový skriptovací jazyk.
- 3 **REST API** (Representational State Transfer) – Rozhraní pro komunikaci mezi jednotlivými částmi systému.
- 4 **Nodejs** – Open-source, cross-platformní javascriptový runtime. Umožňuje spouštět javascript v rámci konzole. Pro to je mezi programátory oblíben pro backendové servery.
- 5 **Wi-Fi SSID** – identifikátor bezdrátové Wi-Fi.
- 6 **MAC adresa (hardwarová adresa)** – Jedinečný identifikátor síťového zařízení.
- 7 **Backend server** – Server provádějící výpočty na datové vrstvě, spravuje přístupy uživatelů k datům a zpracovává informace pro frontend web.
- 8 **Frontend** – Rozhraní pro přístup uživatele.

2 Úvod

Cílem tohoto projektu bylo vytvořit systém pro lokalizování klientských zařízení pomocí Wi-Fi sítě. Data k lokalizaci jsou sbírána pomocí několika zařízení, které odposlouchávají okolní provoz Wi-Fi sítě. Výhodou lokalizace na základě Wi-Fi sítě oproti Bluetooth je, že není vyžadován žádný zásah do již běžících jiných systémů. Avšak jednou z nevýhod oproti Bluetooth, se kterou se již od začátku vývoje počítalo, je nižší přesnost určení konkrétního bodu polohy.

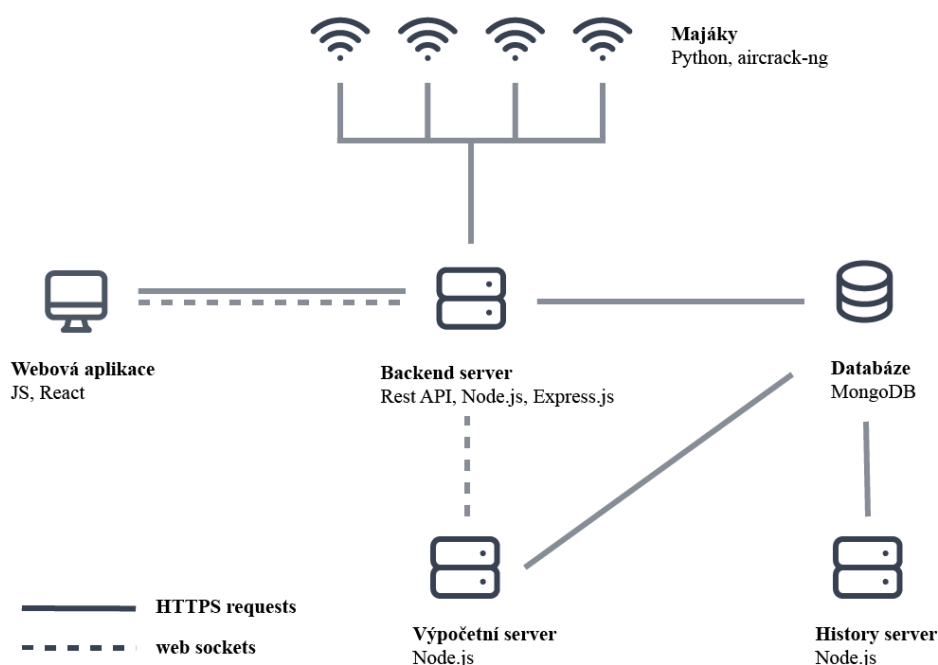
V dnešní době je často potřeba v pracovním prostředí zaznamenávat docházku zaměstnanců. Téměř veškeré docházkové systémy vyžadují od zaměstnanců interakci, ať už zapsat si čas příchodu nebo přiložit čipovou kartu ke čtečce. Tento projekt se snaží problém zaměstnanecké překonat, aniž by bylo nutné použít specializovanou aplikaci pro telefon, protože aplikace často využívají GPS pro určení polohy, což vede k velké spotřebě baterie a je potřeba stále spuštěná aplikace na pozadí telefonu.

Autorovo řešení umožňuje využít lokalizace i u jiných systémů, které využívají komunikaci na základě Wi-Fi sítě. Příkladem takového použití můžou být vnitřní laser tag (laser game) arény.

3 ARCHITEKTURA SYSTÉMU

Systém byl rozdělen do několika částí, které mezi sebou komunikují pomocí TCP/IP požadavků. Pro většinu komunikace se používají https požadavky přímo na backendový server. Pro rychlou aktualizaci zobrazení dat bylo použito WebSocketu, konkrétně řešení pomocí javascriptové knihovny Socket.IO [2]. Hlavním bodem veškeré komunikace vnitřních částí systému je backend server. Každá z těchto částí systému byla tvořena tak, aby ji bylo v budoucnosti možné škálovat při větší zátěži uživatelů na servery.

Při nižší zátěži fyzického serveru je možné zahostovat backend, výpočetní server a history server na jednom serveru, jako několik node.js procesů.



Obrázek 1- Architektura systému

3.1 Majáky

Takzvané majáky slouží k zachycení paketů v rámci Wi-Fi sítě. Pro stanovení pozice je zapotřebí využít několika majáků v rámci dané lokalizace, kterou plánujeme systém využít. Každý maják je tvořen minipočítačem Raspberry Pi a připojeným Wi-Fi adaptérem zapojeným prostřednictvím USB rozhraní. Konkrétně pro řešení tohoto projektu byly využity síťové adaptéry TP Link WN722N, které je možné díky svému chipsetu přepnout do síťového módu monitorování.

Na každém z majáků je při zapnutí minipočítače automaticky spuštěn monitorovací script z knihovny aircrack-ng [3], který zachytává síťové pakety okolní síťové komunikace a ukládá je do

csv souboru. Jako druhý je spuštěn python script, který zachycené informace o paketech v pravidelných intervalech parsuje a odesílá je pomocí REST API na backendový server.

Z každého zachyceného paketu se získává jen potřebné minimum dat. Těmito daty jsou hardwarová adresa zařízení (MAC), které paket odeslalo, a úbytek síly signálu.



Obrázek 2 - Foto majáku

3.2 Backend

Backendový server se stará o komunikaci v rámci jednotlivých částí systému, o validaci dat a jejich částečné zpracování a následné uložení do databáze. Jako technologie backend serveru byl zvolen javascriptový runtime node.js a Framework Express.js [4]. Tento server pomocí REST API ukládá přijatá data z majáků a data uživatelů, která zpět poskytuje do webové aplikace.

Veškeré API endpointy jsou zdokumentovány dle OpenAPI v3 specifikace a publikovány pomocí open-sourcového nástroje Swagger. Dokumentaci lze nalézt na adrese <https://api-wifi.puhony.eu/api-docs>.

3.3 Výpočetní server

Výpočetní server zpracovává samotné výpočty lokalizací. Aby výpočty nepozastavovaly uživatelské požadavky na REST API backend serveru, byly výpočty lokalizací odděleny do samostatného node.js procesu.

Data z backend serveru si výpočetní proces přebírá prostřednictvím databáze MongoDB a následně je zpracovává. Samotné výpočty lokalizací jsou prováděny z krátkodobých dat, která byla uložena v předchozích 10 vteřinách.

Výsledek výpočtu lokalizace je ukládán do cache kolekce v databázi a uživatel tato data získá pomocí HTTP GET požadavku na backend server. Tento výsledek výpočtu odesílá také pomocí WebSocketu uživatelům, kteří mají v té chvíli otevřené zobrazení lokalizace.

3.4 Databáze

K ukládání dat byla zvolena databáze MongoDB. Tento databázový systém se klasifikuje jako NoSQL a využívá dokumenty podobné JSONu. Výhodou MongoDB jsou funkce replikace a následné vyvažování databázových serverů. K replikaci se využívá sady replik, kdy každý člen sady může kdykoli zastávat funkci primární nebo sekundární repliky. Všechny zapisovací a čtecí operace implicitně provádí primární člen, sekundární členové udržují kopie, které je možné využít při výpadku serveru s primárním členem. [5]

Pro práci s touto databází byl zvolen přístup předem definovaných schémat. Tato schémata slouží k lepší strukturalizaci dat a jejich validaci již na úrovni databázového systému.

```
1  const { Schema } = require("mongoose")
2
3  const OrganizationSchema = new Schema({
4    name: {
5      type: String,
6      required: true
7    },
8    created: {
9      type: Date
10   },
11   __v: { type: Number, select: false }
12 })
13
14 module.exports = OrganizationSchema
15
```

Obrázek 3 - Databázové schéma organizace

3.5 Webová aplikace

Webová aplikace je postavena na javascriptové knihovně React [6]. React je populární knihovna pro tvorbu uživatelského rozhraní pomocí tvorby jednotlivých komponentů, které je možné opakovaně použít v rámci projektu. Tato knihovna je vyvíjena jako open-source společností Facebook. [7]

Řešení webové aplikace díky knihovně React umožňuje celý projekt kompilovat do js bundle, který je možné hostovat pomocí Content delivery network (CDN). CDN je síť počítačů vzájemně propojených skrze internet, která zvyšuje dostupnost uložených dat uživatelům. [8]

Nebyl využit žádný css framework a autor systému využil svého vlastního designového řešení. Styly aplikace jsou psané pomocí jazyka SASS, který je transpilován do běžných kaskádových stylů (CSS) [9].

4 METODY VÝPOČTU POLOHY ZAŘÍZENÍ

4.1 Fingerprinty

Tato metoda je jednou z nejvíce používaných, a to hlavně díky vyšší přesnosti oproti jiným metodám ve vnitřních prostorách budov. V porovnání s jinými metodami je jednodušší na implementaci a přesnější. Využívá vytvořených záznamů od uživatele, které se skládají z pozice na mapě a úbytku sil signálů na konkrétním místě. Systém pak při výpočtu hledá záznam, který je součtem rozdílů úbytku signálu nejnižší. [10]

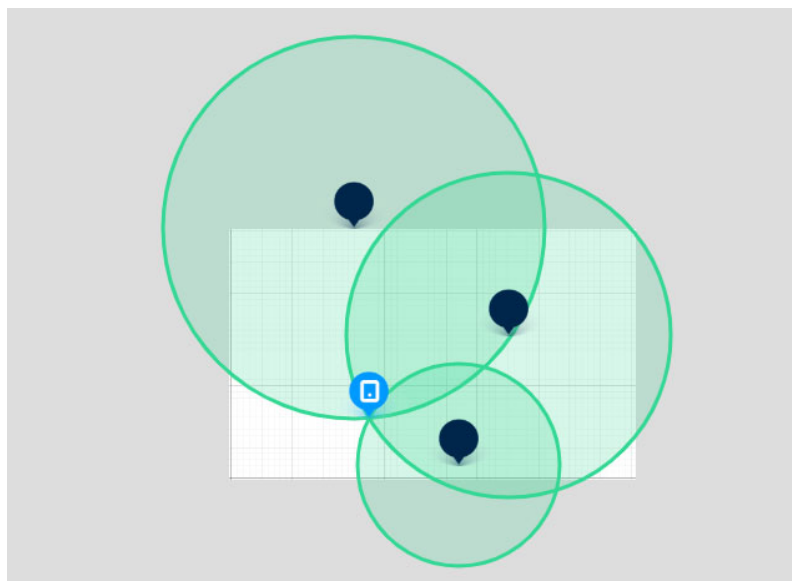
Samotná implementace v systému je taková že uživatel zvolí aktuální pozici zařízení na plánu budovy a následně počká na nově přichozí data z majáků. Data jsou odeslána na výpočetní server až po kontrole samotným uživatelem.

```
  _id: ObjectId("60207477829f5b04a8f8f5c8")
  x: 805
  y: 179
  f: 1
  beacons: Array
    0: Object
      _id: ObjectId("60207477829f5b04a8f8f5c9")
      deviceKey: "00000000298f6e70"
      rssi: -74
    1: Object
      _id: ObjectId("60207477829f5b04a8f8f5ca")
      deviceKey: "10000000d456a21"
      rssi: -50
    2: Object
      _id: ObjectId("60207477829f5b04a8f8f5cb")
      deviceKey: "00000000c05f6f00"
      rssi: -70
    3: Object
      _id: ObjectId("60207477829f5b04a8f8f5cc")
      deviceKey: "00000000889cd176"
      rssi: -82
  localizationId: "60206249b697774c304f8318"
  __v: 0
```

Obrázek 4 - Záznam polohy uložený v databázi MongoDB

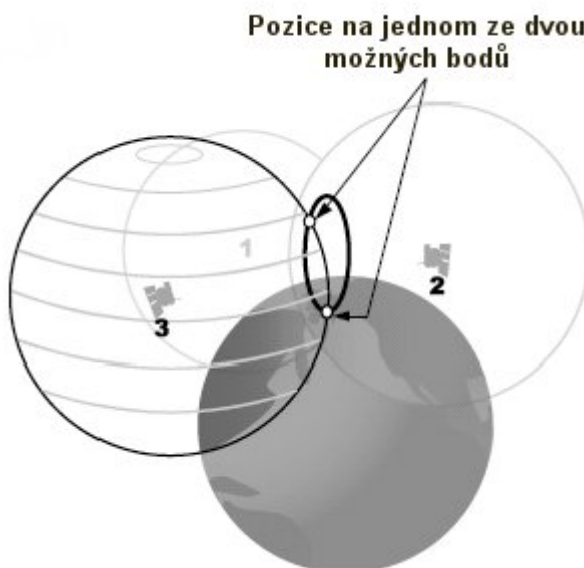
4.2 Trilaterace

Trilaterace vypočítává průnik několika kružnic, kdy průměr kružnic představuje vzdálenosti od majáku. Tato metoda je podobná triangulaci, která také umožňuje určit polohu zařízení, ale na základě znalosti úhlu, pod kterým signál dorazil. [11]



Obrázek 5 - Trilaterace v 2D prostoru

Tuto metodu je možné využít i ve trojrozměrném poli, kdy není oproti dvourozměrné roz-
dílňá, je však složitější ji vizualizovat. Lze ji využít i jako jednu z teoretických možností vý-
počtu polohy zařízení pomocí dat ze satelitů nebo výše položených antén. Jednotlivé vzdále-
nosti od satelitu či antény jsou představovány pláštěm koule. Při průniku dvou koulí získáme
kružnici, na které se zařízení může nacházet. Abychom určili polohu zařízení musíme přidat
další data. Při průniku 3 koulí již získáváme pouze 2 body, z nichž můžeme jeden odebrat,
pokud vezmeme v potaz znalost, že zařízení se nenachází ve vzduchu, v opačném případě mu-
síme přidat čtvrtý satelit nebo anténu. [12]



Obrázek 6 - Triangulace v 3D prostoru [12]

Trilaterace se také využívá při upřesňování polohy v mapách, kde GPS signál není dostatečně přesný. To se může stát, pokud zařízení je mezi vysokými budovami, kde je větší chybovost GPS. V tomto případě si zařízení získá jednotlivé síly signálu od okolních Wi-Fi access pointů a antén telefonní sítě. Tyto údaje uživatelské zařízení zašle na Google Geolocation API, která přijímá objekt těchto dat a vrací přibližnou polohu dle již dříve sesbíraných dat. [11] [13]

4.3 Neuronová síť

Jednou z dalších možností lokalizace je využití tzv. neuronové sítě. Neuronová síť je jeden z modelů používaný v umělé inteligenci. Jejím vzorem je chování neuronových sítí v biologii, které tvoří živý mozek. Pro využití tohoto modelu je zapotřebí jej nejdříve „vycvičit“ na modelových datech. Tato data se skládají ze vstupních hodnot a výsledku. Po fázi učení modelu se odeberou výsledky a vkládají se pouze vstupní hodnoty, model poté vrací predikce výstupních hodnot. [14]

Jedním z nejjednodušších řešení pro využití neurálních sítí v rámci node.js runtime je knihovna Brain.JS [15], která byla využita při testování této metody lokalizace.

Neuronové sítě pro výpočet polohy je možné využít ve venkovním prostředí, kde je možno určit závislost mezi naměřenými daty, avšak uvnitř budov, kde je signál ovlivněn velkým množstvím odrazů a ztráty síly signálu, je tato metoda, jak jsme zjistili, neúčinná.

5 ZABEZPEČENÍ

5.1 Server

System je hostován na virtuálních privátních serverech, pro komunikaci s vnější sítí je využito reverzní proxy Nginx. Veškeré požadavky jsou vyřizovány pouze přes SSL/HTTPS protokol. Požadavky, které dorazí pomocí nezabezpečeného připojení HTTP jsou přesměrovány. Nevyužívané porty, nebo porty určené pouze pro vnitřní komunikaci systému jsou blokovány pomocí firewallu.

5.2 Zabezpečení přístupu

Samotná verifikace uživatele na backend serveru byla provedena pomocí JWT tokenů, které mají omezenou dobu platnosti.

JSON Web Token je standardní metodou RFC 7519 [16], kdy server poskytne data a vygenerovaný podpis zakódované v Base64 tokenu. Tato data je uživatel schopen dekodovat, avšak už nemá možné získaný token upravit tak, aby byl nadále validní, bez znalosti neveřejného klíče.

Samotný token se skládá ze tří částí.

- Z hlavičky obsahující popis využitého algoritmu a typ tokenu.
- Z obsahu – dat zasláných serverem.

- Poslední část je podpis, kterým je jednostranně generovaný hash zasláné hlavičky a obsahu pomocí algoritmu, který využívá neveřejného klíče.

Tento token uživatel zasílá s každým požadavkem na backend server, kde se provede znovu hashovací algoritmus hlavičky a obsahu a následně se porovná zasláný podpis s podpisem zasláným uživatelem. [17]

K získání tokenu se uživatel musí autorizovat na základě zadání emailové adresy a hesla. Hesla jsou ukládána v databázi pomocí algoritmu bcrypt s přidanou solí.

Veškeré endpointy pro autorizaci uživatele jsou omezeny počtem požadavků za určitý časový úsek, a požadavky pro přihlášení jsou zpracovávány posupně.

5.3 Validace dat zasláných pomocí REST API

Data, která backend server přijímá od webové aplikace a majáků, jsou validována vůči samotné dokumentaci API pomocí knihovny express-openapi [18].

5.4 Ochrana proti DDOS

Aby byl systém ochráněn proti DDOS útokům je využita služba Cloudflare [19]. Tato služba při podezřelém chování uživatelů přesměrovává podezřelé IP adresy na virtuální frontu, kde je uživatel validován pomocí technologie reCaptcha [20].

6 MĚŘENÍ ÚBYTKU SÍLY WI-FI SIGNÁLU

6.1 Vlivy úbytku signálu

Veškerá Wi-Fi zařízení komunikují pomocí standardizovaného protokolu IEEE 802.11 [21] a pro samotný přenos se využívá rádiových vln. Nosným médiem je tedy vzduch. Samotná síla signálu je ovlivněna frekvencí a citlivostí antén, které přijímají rádiové vlny. [22] Nesmíme zapomenout na samotné prostředí šíření signálu, např. sádrokartonová zeď odebere zhruba 4 dB signálu, cihlová 8 dB a betonová mezi 10-15 dB. V neposlední řadě se intenzita signálu snižuje se vzdáleností, protože se energie elektromagnetických vln rozprostírá do většího prostorového úhlu. Toto je jev zvaný Free Space Path Loss (ztráta při šíření otevřeným prostorem). [23]

Vliv na prostředí	Materiály	Ztráta signálu [dB]
Nízký	Čiré sklo, dřevěné dveře, tvárniceové zdi, omítka	2-4
Střední	Cihlové zdi, mramor, drátěný plot, tmavé sklo	5-8
Vysoký	Betonové zdi, papír, tvrzené sklo	10-15
Velmi vysoký	Kov, zrcadlo	>15

Tabulka 1 - Typický útlum stavebního materiálu [24]

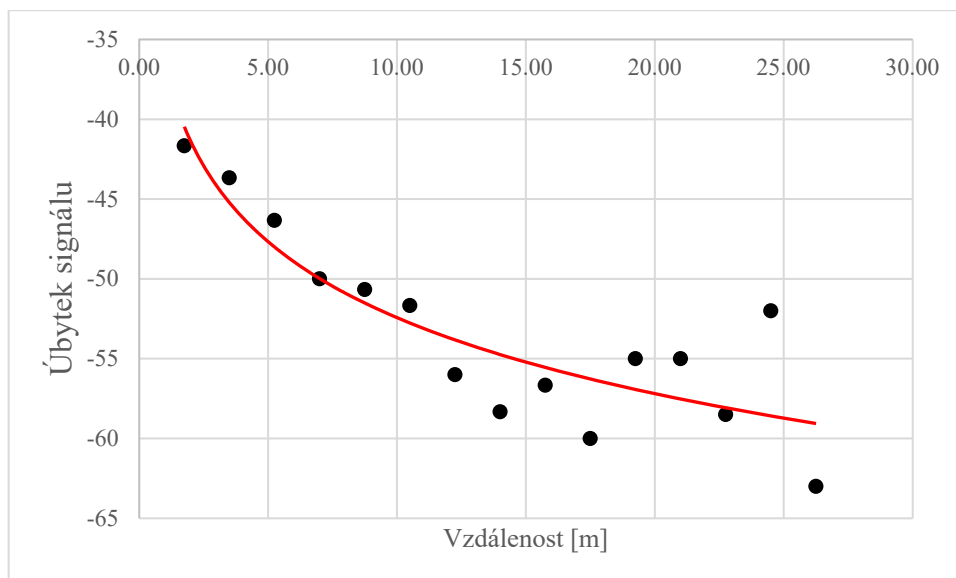
Další vliv na kvalitu a sílu signálu je vzájemné rušení na stejné vysílací frekvenci. Tímto rušení mohou být i jiné rádiové vysílače, například mikrovlnné trouby. [25] Při měření údaj o vzájemném rušení je označován zkratkou SNR (Signal to Noise Ratio), který udává poměr mezi silou signálu a úrovní rušení. [26]

6.1.1 Vlastní měření

Vlastní měření bylo prováděno venku za použití jednoho z Wi-Fi majáků a notebooku na frekvenci 2.4 GHz. Bylo využito notebooku, kvůli stabilnějšímu získu hodnot. Při použití mobilního telefonu se údaje RSSI příliš měnily, a to už i při naklonění telefonu, to samo už ale ukazuje na omezení použití útlumu signálu pro lokalizaci.

Vzdálenost [m]	1. měření [dBm]	2. měření [dBm]	3. měření [dBm]	Průměr [dBm]
1.75	-39	-43	-43	-42
3.50	-42	-44	-45	-44
5.25	-42	-50	-47	-46
7.00	-45	-53	-52	-50
8.75	-47	-55	-50	-51
10.50	-50	-50	-55	-52
12.25	-50	-57	-61	-56
14.00	-50	-61	-64	-58
15.75	-53	-50	-67	-57
17.50	-57	-63	-60	-60
19.25	-	-50	-60	-55
21.00	-	-50	-60	-55
22.75	-	-57	-60	-59
24.50	-	-52	-	-52
26.25	-	-63	-	-63

Tabulka 2- Vlastní měření úbytku sil signálu



Graf 1 - Vlastní měření úbytku sil signálu v závislosti na vzdálenosti. Vidíme, že od cca 10–15 metrů je náhodný šum významnější než vlastní charakter klesající křivky.

Po proložení logaritmické závislosti nám vyjde rovnice regresní křivky takto:

$$y = -6,87 \cdot \ln(x) - 36,611$$

Tuto rovnici by bylo možné teoreticky použít pro lokalizaci pomocí trilaterace, pokud by body byly blíže křivce a křivka jako taková nezávisela na použitém měřicím zařízení.

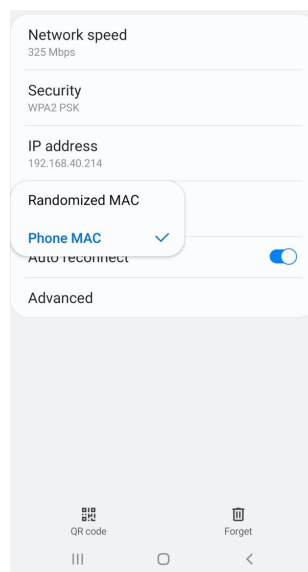
6.2 Náhodné MAC adresy u mobilních zařízení

Vývojáři mobilních telefonů v posledních letech zjistili trend sledování uživatelů v pomoci Wi-Fi sítě a z důvodu ochrany dat uživatelů a bezpečnosti telefony začaly využívat náhodně generované MAC adresy. Tyto MAC adresy se generují při prvním připojení k novému SSID nebo po zapomenutí sítě.

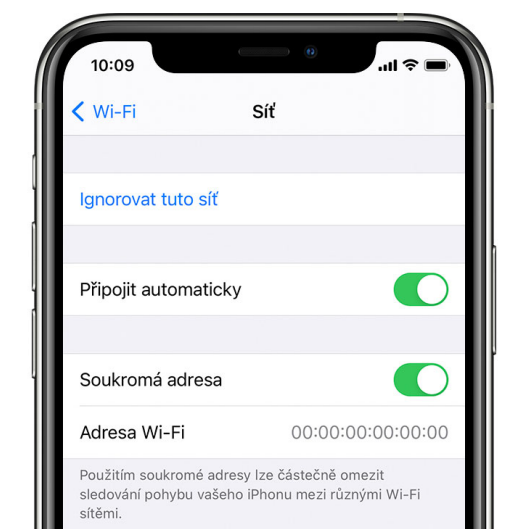
U systému android tato funkce přichází již ve verzi 8.0 kdy zařízení používá náhodnou adresu při sondování okolních sítí. Ve verzi 9 je již možno zapnout používání náhodných MAC adres v nastaveních pro vývojáře, v základním nastavení je však stále tato funkce vypnuta. Od verze operačního systému Android 10 je tato možnost již ze základu zapnuta pro klientský mód Wi-Fi, Wi-Fi direct a některé další funkce telefonu. [27]

V operačním systému IOS tato možnost je dostupná od verze IOS 14 pro mobilní zařízení iPhone, iPad, iPod touch. Pro chytré hodinky Apple Watch je tato funkce dostupná od aktualizace watchOS 7. [28]

V obou případech lze náhodné MAC adresy vypnout pro konkrétní Wi-Fi SSID.



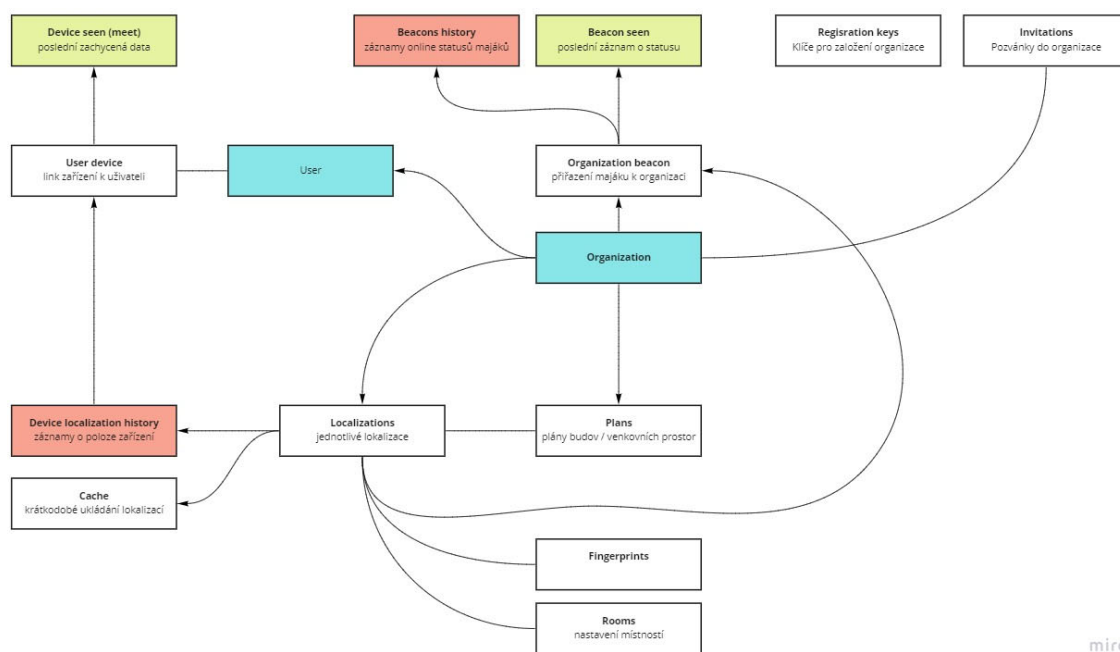
Obrázek 7 -Nastavení náhodné MAC adresy Android [29]



Obrázek 8 - Nastavení náhodné MAC adresy IOS [28]

7 DATOVÝ MODEL SYSTÉMU

Systém byl postaven na databázi MongoDB, která se klasifikuje jako NoSQL. Pro komunikaci s databází je využita knihovna mongoose [30], která umožňuje vytvářet modely pro jednotlivé typy objektů v databázi. V těchto modelech lze nastavit datové typy jednotlivých vlastností objektu a vzájemné propojení s dalšími objekty. Systém modely (nazývané též jako schémata) využívá pro většinu uložených dat.



Obrázek 9 – Diagram vzájemného propojení modelů databáze

V diagramu jsou označeny žlutě modely, které slouží pro krátkodobé uložení dat. Jedná se o zachycená data z majáků a datum posledních příchozích dat z majáků. Tyto objekty v databázi se po uplynutí krátkého časového úseku automaticky mažou. Mazání dat je automatizováno pomocí CRON funkcí, které jsou nastaveny na backend serveru.

Samotné přiřazení majáku k organizaci je uskutečněno v nezávislém modelu, kde je uloženo i nastavené pojmenování a popis reálného umístění od uživatele. Podobně tomu je u zařízení uživatelů. (v diagramu označeno jako „Organization beacon“ a „User device“)

Model lokalizace je napojen na několik dalších modelů. Místnosti slouží k pojmenování různých částí vrstvy plánu a uživatel je nastavuje pro každou lokalizaci. Fingerpriny jsou též ukládány pouze pro konkrétní lokalizaci a slouží pro výpočet polohy uživatelských zařízení. Výpočty lokalizací se provádějí vždy za daný časový úsek a spouští se pomocí CRON funkce na backend serveru. Systém výsledek výpočtů ukládá do nestriktního modelu cache, backend server uložené objekty poskytuje webové aplikaci pro zobrazení lokalizace.

Červeně označený model „Device localization history“ ukládá data o poloze uživatelských zařízení včetně názvů místností. Podle těchto dat uživatel vidí polohu zařízení v minulosti. Data

názvů místností se v rámci objektu po několika dnech smažou, samotné záznamy o tom, zda byla zařízení lokalizována, v databázi nadále zůstávají.

Model „Beacon history“ slouží k uložení dat kdy byly majáky aktivní, uživatel na základě těchto dat vidí historicky výpadky majáků.

8 WEBOVÉ UI/UX – FUNKCIONALITY

8.1 Správa organizace

Systém umožňuje uživateli vytvářet veškeré plány a lokalizace pouze v rámci organizace. Ve své organizaci může uživatel přidávat nové uživatele a k nim přiřazovat uživatelská zařízení. Aby uživatel mohl organizaci upravovat, byla vytvořena stránka „Organizace“.

Uživatel zde může přizvat nové členy do organizace, nebo odebrat stávající. K přidáním členům, lze následovně přidat jejich zařízení na základě MAC adresy. Tuto MAC adresu je nejjednodušší nalézt na Wi-Fi AP.

Tenda

Internet Status

Internet Settings

WiFi Settings

Guest Network

Parental Control

VPN

Advanced Settings

System Settings

Parental Control

English | Exit

Device Name	MAC Address	Uptime	Operation
? LAPTOP-IR160EBR 192.168.1.109	c0:b6:f9: [REDACTED]	6min 3s	
? Galaxy-A51 192.168.1.226	26:29: [REDACTED]	33min 33s	
? ATN3-raspberrypi 192.168.1.198	dc:a6: [REDACTED]	45min 25s	
? Galaxy-A50 192.168.1.155	f2:ec: [REDACTED]	1hour(s) 55min 2s	
? android-8dd8652277e446e1 192.168.1.119	38:18: [REDACTED]	2hour(s) 3min 5s	
? DESKTOP-SK9U4VS 192.168.1.7	84:4b: [REDACTED]	2hour(s) 3min 34s	

Obrázek 10 – Příklad získání MAC adres připojených k Wi-Fi AP

Na této stránce je možné i přidat nové majáky, ty se přidávají na základě jejich unikátního identifikátoru, kterým je device key. Právě pod tímto identifikátorem odesílají majáky data na backend server. Pro jednodušší zadání klíče je možné jej naskenovat pomocí QR kódu.

```
def getserial():
    cpuserial = "0000000000000000"
    try:
        f = open('/proc/cpuinfo', 'r')
        for line in f:
            if line[0:6]=='Serial':
                cpuserial = line[10:26]
        f.close()
    except:
        cpuserial = "ERROR00000000"

    return cpuserial
```

Obrázek 11 - Python – získání device key



Obrázek 12 - QR kód pro přidání majáku

8.2 Plány a lokalizace

Samotné nastavení lokalizace uživatel provede v rámci stránky „Lokalizace“. Na této stránce uživatel může nastavit nové plány. Do jednotlivých plánů si uživatel nahrává obrazové soubory s mapami pater budov nebo venkovních prostor. Uživatel soubory odešle při vytváření plánu na backend server, který je následně pomocí protokolu FTP uloží do samostatného souborového úložiště. Pro zobrazení těchto obrázků se používá pouze HTTPS protokolu.

Uložené plány uživatel může využít pro lokalizace. U vytváření lokalizace uživatel zvolí plán, typ lokalizace a samotné majáky, které budou využity pro vypočítávání polohy pro danou lokalizaci. Pro lokalizaci, kdy uživatel zvolí metodu výpočtu pomocí trilaterace, je povoleno použít plány s jednou vrstvou (patrem).

8.3 Záznamy stavu majáků a historie polohy zařízení

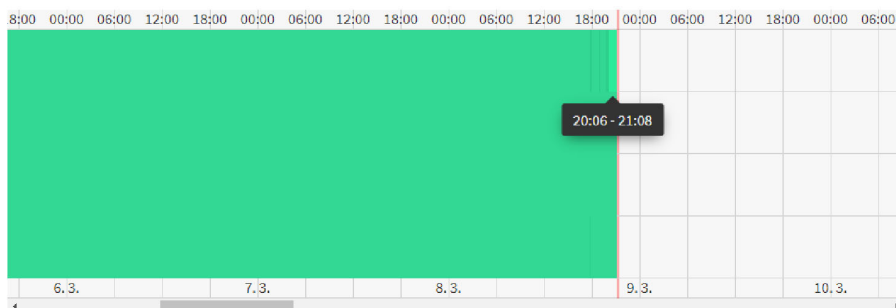
Systém jednou za několik minut provolává pomocí CRONU funkci, která porovná online stav jednotlivých majáků a poslední uložený stav v databázové kolekci uložených změn. Pokud se stav majáku liší, tak se uloží do databáze nový objekt změny s časovým údajem.

Při výpisu se projdou uložené objekty změn stavů a jsou převedeny na časové události, které mají definovaný začátek a konec aktivního statusu. Tyto události si uživatel může zobrazit díky časové ose, která se nachází na stránce „status majáků“. Data se dají zobrazit pro detail omezeného časového úseku (den, týden, měsíc).

Majáky 3/2021

- Měsíc +

ANT-4
00000000c
ANT-2
000000008
ANT-1
000000002
ANT-3
100000000



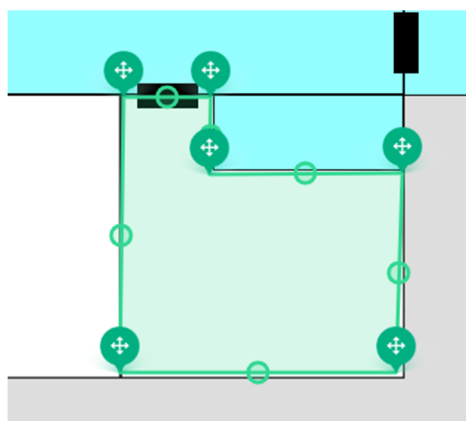
Stejné zobrazení je vytvořeno i pro polohu zařízení v rámci lokalizace. U historie polohy uživatelských zařízení jsou uloženy také názvy místností.

8.4 Zobrazení lokalizace a nastavení lokalizace

Nejdůležitější část webové aplikace je samotné zobrazení lokalizace. Pro zobrazení plánu budovy bylo využito knihovny Leaflet [31], která slouží pro vykreslování interaktivních map. Tato knihovna kombinuje absolutní pozicování elementů a vykreslování prvků na canvas. Dále je při vykreslování mapy využita knihovna Overlapping Marker Spidefier for Leaflet, která se stará o efekt rozdělení jednotlivých pinů na mapě při najetí myši.

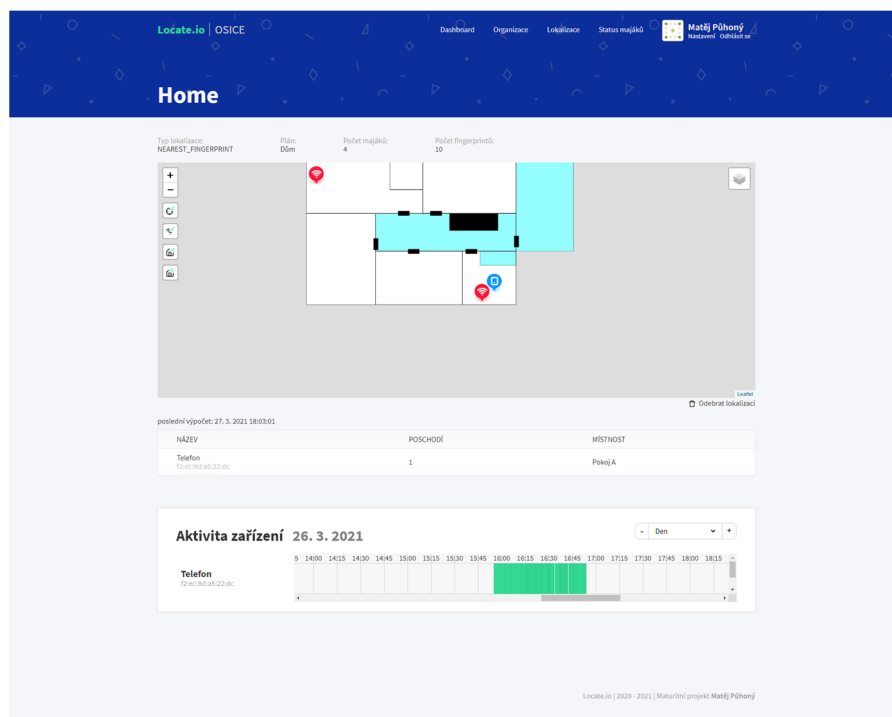
Pro nastavení lokalizace byla přidána tlačítka přímo do zobrazení plánu. Uživatel se zobrazují možnosti podle druhu výpočtů lokalizace. Jsou zde umístěna tlačítka editace polohy majáků, přidání fingerprintů a editace místností. Uložené nastavení a fingerprinty se využívají v rámci výpočtů lokalizace.

Pro editaci místnosti je využito vlastní řešení pro vyznačení polygonu na plánu. Uživatel má možnost přidávat neomezené množství počtu bodů polygonu. Po dokončení výpočtu lokalizace se provede na výpočetním serveru též funkce pro přiřazení místnosti pro konkrétní zařízení. Tato funkce zkoumá průnik bodu polohy zařízení s uloženými polygony místností. Pro výpočty průniku bodu s polygonem je využita knihovna point-in-polygon [32].



Obrázek 13 - Nastavení polygonu místnosti

Tato stránka zobrazení lokalizace si získává aktuální data díky kombinaci HTTP Get požadavků a dat získaných pomocí připojených WebSocketů. WebSocket server zprostředkovává hlavní backend server, avšak samotné aktualizace zasílá výpočetní server po úspěšném provedení výpočtu.



Obrázek 14 - Zobrazení lokalizace

9 VÝSLEDKY LOKALIZACÍ

9.1 Venkovní prostředí

Výsledkem využití metody trilaterace a fingerprintů systém dokáže určit polohu zařízení v rámci několika jednotek metrů. Přesnost polohy odpovídá i konkurujícím komerčním projektům, které se zabývají podobnou problematikou určení polohy zařízení. Tyto projekty udávají většinou přesnost určení polohy do 15 m. [33] [34].

Bohužel kvůli nepřesnostem měření díky překážkám (př. automobily, stromy), odrazů signálu a rušením ostatních zdrojů i směrovými charakteristikami přenosových antén, v některých případech systém určí polohu špatně.

Za účelem ukázky lokalizace bylo vytvořeno několik videí porovnávající výsledek lokalizace zobrazený na interaktivní mapce a reálnou pozicí zařízení. Tato videa jsou dostupná v přílohách práce.

Značné nepřesnosti měření v implementaci systému přináší problém při využití neuronové sítě. Před zahájením vývoje byl vytvořen fungující prototyp, který využíval simulované údaje, bez

započtení chyb měření o úbytku sil signálu. Kvůli nepřesnostem v měření v reálném prostředí se projevilo že využití neuronové sítě je v rámci implementace v projektu neúčinné a využitý model v rámci knihovny Brain.js [15] je i s větším množstvím učících dat nepřesný.

9.2 Vnitřní prostředí

Ve vnitřních prostorách systém určí polohu zařízení pomocí metody fingerprintů spolehlivě s přesností na jednotlivé místnosti v rámci budovy. Výsledek určení polohy je ovlivněn dostatečným počtem majáků a jejich správným rozmístěním. Uvnitř budov systému naopak oproti venkovnímu prostředí využívá ztráty signálu při průchodu různých zdí ke svému prospěchu. Fingerprinty díky průchodu signálu zdmi jsou od sebe více rozlišitelné. Systém také v rámci výpočtu polohy zařízení upřednostní fingerprinty, které jsou blíž k majáku, který zachytil nejmenší ztrátu signálu.

Kvůli již zmíněným úbytkům síly signálu při průchodů zdmi je systém naopak méně přesný, pokud se uživatel pohybuje či je v místě kde nejsou fingerprinty nasnímané. Avšak tyto nepřesnosti by bylo možné odfiltrovat před samotným výpočtem polohy zařízení.

10 ZÁVĚR

Autor projektu vytvořil systém, který umožňuje sledovat polohu Wi-Fi zařízení na základě ztráty síly Wi-Fi signálu. Uživatelé mohou spravovat jednotlivé interaktivní mapky v rámci své organizace za pomoci vytvořené webové aplikace. Na mapkách ve webové aplikaci uživatel vidí polohu sledovaných Wi-Fi zařízení v reálném čase. Výpočty polohy zařízení se plně provádí na serverech bez nutnosti instalovat aplikace do sledovaných zařízení. Tyto výpočty jsou prováděny z dat o ztrátě síly signálu, které zachytí autorem vytvořené majáky.

Autor porovnal jednotlivé metody lokalizace na základě údaje o ztrátě síly signálu. Metody, které jsou v této práci zmíněny jsou dostupné v rámci systému.

Některé plánované funkce systému, například určení polohy zákazníků v nákupních střediscích, nebo sčítání návštěvnosti místa se ukázalo v průběhu řešení projektu jako neproveditelné.

Použitá literatura

- [1] „Raspberry Pi,“ [Online]. Available: https://en.wikipedia.org/wiki/Raspberry_Pi. [Přístup získán 17 2 2021].
- [2] „Socket.IO,“ [Online]. Available: <https://socket.io/docs/v3>. [Přístup získán 13 3 2021].
- [3] AirCrack-ng. [Online]. Available: <https://www.aircrack-ng.org/>. [Přístup získán 13 3 2021].
- [4] „Express,“ [Online]. Available: <https://expressjs.com/>. [Přístup získán 13 3 2021].
- [5] „MongoDB,“ [Online]. Available: <https://cs.wikipedia.org/wiki/MongoDB>. [Přístup získán 27 2 2021].
- [6] „React,“ [Online]. Available: <https://reactjs.org/>. [Přístup získán 8 3 2021].
- [7] „React (JavaScript library),“ [Online]. Available: [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library)). [Přístup získán 8 3 2021].
- [8] „Content delivery network,“ [Online]. Available: https://cs.wikipedia.org/wiki/Content_delivery_network. [Přístup získán 8 3 2021].
- [9] „Sass (stylesheet language),“ [Online]. Available: [https://en.wikipedia.org/wiki/Sass_\(stylesheet_language\)](https://en.wikipedia.org/wiki/Sass_(stylesheet_language)). [Přístup získán 8 3 2021].
- [10] S. S. a. J.-Y. Pyun, „Practical Fingerprinting Localization for Indoor Positioning System by Using Beacons,“ 2017. [Online]. Available: <https://www.hindawi.com/journals/js/2017/9742170/>. [Přístup získán 25 02 2021].
- [11] M. Croak, „Wi-Fi Trilateration Using Node.js,“ 31 3 2020. [Online]. Available: <https://medium.com/better-programming/wi-fi-trilateration-using-node-js-74996a3f3d1>. [Přístup získán 12 2 2021].
- [12] „Jak funguje GPS?,“ 21 6 2006. [Online]. Available: svethardware.cz/jak-funguje-gps/21826-5. [Přístup získán 26 2 2021].
- [13] „Geolocation API - Overview,“ [Online]. Available: <https://developers.google.com/maps/documentation/geolocation/overview>. [Přístup získán 26 2 2021].
- [14] „Artificial neural network,“ [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Přístup získán 14 3 2021].

- [15] „Brain.js,“ [Online]. Available: <https://brain.js.org/#/>. [Přístup získán 14 3 2021].
- [16] „JSON Web Token (JWT),“ 5 2015. [Online]. Available: <https://tools.ietf.org/html/rfc7519>. [Přístup získán 23 2 2021].
- [17] „Introduction to JSON Web Tokens,“ [Online]. Available: <https://jwt.io/introduction>. [Přístup získán 16 2 2021].
- [18] K. Software, „express-openapi,“ [Online]. Available: <https://github.com/kogosoftwarellc/open-api/tree/master/packages/express-openapi#readme>. [Přístup získán 21 2 2021].
- [19] „Cloudflare,“ [Online]. Available: <https://www.cloudflare.com/>. [Přístup získán 15 3 2021].
- [20] „reCAPTCHA,“ [Online]. Available: <https://www.google.com/recaptcha/about/>. [Přístup získán 15 3 2021].
- [21] „IEEE 802.11,“ [Online]. Available: https://cs.wikipedia.org/wiki/IEEE_802.11. [Přístup získán 28 2 2021].
- [22] R. Fürst, „Měření a vizualizace signálu bezdrátových sítí v interiéru,“ 2015. [Online]. Available: https://dspace.cvut.cz/bitstream/handle/10467/65192/F3-BP-2016-Furst-Roman-mereni_a_vizualizace_signalu_bezdratovych_siti_v_interieru.pdf?sequence=1&isAllowed=y. [Přístup získán 28 2 2021].
- [23] „Free-space path loss,“ [Online]. Available: https://en.wikipedia.org/wiki/Free-space_path_loss. [Přístup získán 28 2 2021].
- [24] J. D. R. P. V. G. Kemisola Ogunjemilua, „An Investigation into Signal Strength of 802.11n WLAN,“ [Online]. Available: <https://www.semanticscholar.org/paper/An-Investigation-into-Signal-Strength-of-802.11n-Ogunjemilua-Davies/1928a0bf02f854844d7bb632c064f09b835de52f/figure/1>. [Přístup získán 28 2 2021].
- [25] „Do Microwaves Interfere With WiFi Signals?,“ [Online]. Available: <https://www.scienceabc.com/innovation/do-microwaves-interfere-with-wifi-signals.html>. [Přístup získán 22 3 2021].
- [26] „Wireless Signal Strength and Noise Levels,“ [Online]. Available: https://www.watchguard.com/help/docs/help-center/en-US/Content/en-US/Fireware/wireless/ap_wireless_signalstrength_c.html. [Přístup získán 22 3 2021].

- [27] „Android Develop,“ 28 10 2020. [Online]. Available: <https://source.android.com/devices/tech/connect/wifi-mac-randomization>. [Přístup získán 15 2 2021].
- [28] „Používání soukromých Wi-Fi adres v systémech iOS 14, iPadOS 14 a watchOS 7,“ 17 11 2020. [Online]. Available: <https://support.apple.com/cs-cz/HT211227>. [Přístup získán 16 2 2021].
- [29] M. Burton, „Wi-Fi MAC Randomization – Privacy and Collateral Damage,“ 7 10 2020. [Online]. Available: <https://www.extremenetworks.com/extreme-networks-blog/wi-fi-mac-randomization-privacy-and-collateral-damage/>. [Přístup získán 16 2 2021].
- [30] „Mongoose,“ [Online]. Available: <https://mongoosejs.com/>. [Přístup získán 3 7 2021].
- [31] „Leaflet,“ [Online]. Available: <https://leafletjs.com/examples.html>. [Přístup získán 8 3 2021].
- [32] „point-in-polygon,“ [Online]. Available: <https://www.npmjs.com/package/point-in-polygon>. [Přístup získán 8 3 2021].
- [33] S. Srivastava, „WiFi for Indoor Positioning,“ 24 5 2019. [Online]. Available: <https://www.iotforall.com/wifi-indoor-positioning>. [Přístup získán 26 3 2021].
- [34] „Indoor Positioning, Tracking and Indoor Navigation with Wi-Fi,“ [Online]. Available: <https://www.infsoft.com/technology/positioning-technologies/wi-fi>. [Přístup získán 26 3 2021].

11 SEZNAM OBRÁZKŮ A TABULEK

Obrázek 1 - Architektura systému	9
Obrázek 2 - Foto majáku	10
Obrázek 3 - Databázové schéma organizace	11
Obrázek 4 - Záznam polohy uložený v databázi MongoDB	12
Obrázek 5 - Trilaterace v 2D prostoru	13
Obrázek 6 - Triangulace v 3D prostoru [12].....	13
Obrázek 7 -Nastavení náhodné MAC adresy Android [29]	18
Obrázek 8 - Nastavení náhodné MAC adresy IOS [28]	18
Obrázek 9 – Diagram vzájemného propojení modelů databáze	19
Obrázek 10 – Příklad získání MAC adres připojených k Wi-Fi AP.....	20
Obrázek 11 - Python – získání device key.....	21
Obrázek 12 - QR kód pro přidání majáku.....	21
Obrázek 13 - Nastavení polygonu místnosti.....	22
Obrázek 14 - Zobrazení lokalizace	23
Tabulka 1 - Typický útlum stavebního materiálu [24]	16
Tabulka 2- Vlastní měření úbytku sil signálu	16
Graf 1 - Vlastní měření úbytku sil signálu v závislosti na vzdálenosti. Vidíme, že od cca 10–15 metrů je náhodný šum významnější než vlastní charakter klesající křivky.....	17

12 SEZNAM PŘÍLOH

12.1 Příloha 1: Snímky z webové aplikace

12.2 Příloha 2: Ukázka designu