

# **FUNKČNÝ OPIS MODELU TECHNICKÉHO RIEŠENIA SYSTÉMU TRASOVANIA OSÔB**

Na základe analýzy technických prvkov lokalizačných systémov a analýzy možných technických riešení je v nasledujúcich kapitolách opísaný celistvý model technického riešenia systému trasovania osôb. Prvá kapitola sa venuje jednotlivým komponentom systému vrátane ich technických parametrov. V druhej kapitole je opísané sieťové zapojenie systému. V tretej kapitole je schematicky znázornený dátový tok v systéme s vysvetlením jednotlivých častí. Dokument obsahuje aj Prílohu 1, ktorá obsahuje vytvorený procesný model technického riešenia na báze vybraných programovateľných stavebníc Raspberry Pi a Arduino.

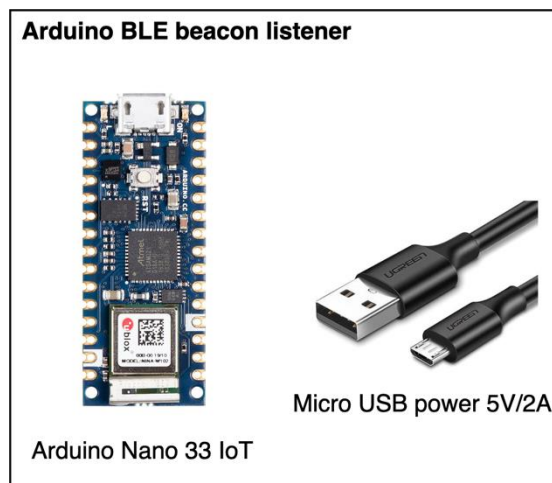
## **1. KOMPONENTY SYSTÉMU**

Pri výbere komponentov systému bolo dôležité analyzovať ich technické možnosti vrátane dostupnosti technického riešenia. Systému dominujú čítacie zariadenia (Arduino, Raspberry), ktoré je možné programovať podľa výsledkov analýzy požiadaviek. Tieto čítačky získavajú signál z tzv. tagov, beacon čipov. Sú to malé a cenovo dostupné nositeľné zariadenia, ktoré je možné prečítať s ich špecifickým kódom a tým prideliť ostatné informácie, napríklad informácie o polohe. Zapojenie do systému je realizované prostredníctvom servera Linux – Docker s dostupnými aplikáciami.

### **1.1. BEACON ČÍTAČKY**

Najbežnejším prístupom k takémuto riešeniu sú pevné body, na ktorých sú umiestnené čítačky v tzv. vnútornej mape. Vzhľadom na povahu signálu Bluetooth, ktorý môže byť ovplyvnený akoukoľvek prekážkou medzi odosielateľom a prijímačom, možno na získanie spoľahlivej polohy použiť rôzne techniky. Tieto umožňujú sledovanie zariadenia s podporou Bluetooth. Údaje o sledovaní, ktoré možno zhromaždiť z každého zariadenia, sa potom môžu odoslať do centralizovaného systému na analytické účely a ďalšie služby, ako je mapovanie čipov v reálnom čase. Takýmito prijímacími prvkami môžu byť programovateľné komponenty, ako je napríklad mikrokontrolér Arduino Nano IoT, zobrazený na Obrázku 1.1, ktoré disponuje ako Wi-Fi, tak Bluetooth konektivitou. Nakoľko sa však jedná o kontrolér, dokáže súčasne robiť iba jednu úlohu, a teda buď prijíma údaje od BLE zariadení, alebo ich posiela na server.

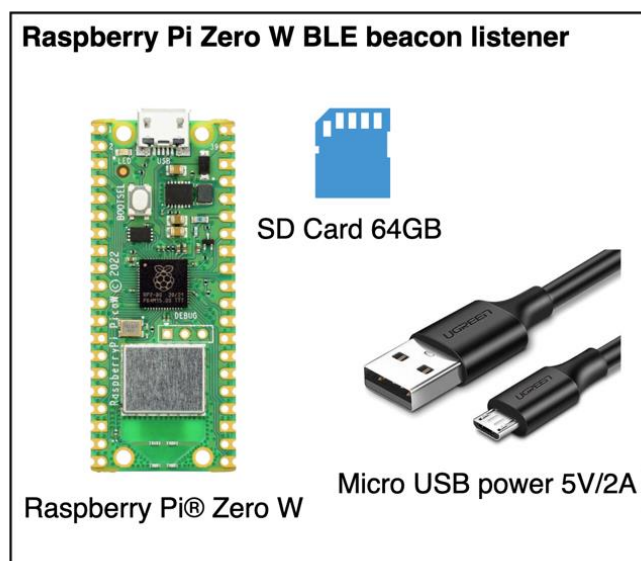
Druhou možnosťou je využitie programovateľného minipočítača Raspberry Pi® Zero W, zobrazený na Obrázku 1.2, ktorý dokáže súčasne prijímať aj odosielať údaje. Obidva čítacie zariadenia je možné napájať prostredníctvom mikroUSB konektora. Kompletné technické parametre jednotlivých komponentov sú uvedené v Tabuľke 1.1 a Tabuľke 1.2. Pri práci s minipočítačom, Raspberry Pi® Zero W, je potrebné využiť aj dodatočnú operačnú pamäť určenú pre funkčnosť, v našom prípade sme využili 64GB mikroSD karty.



Obrázok 1.1 – Arduino Nano 33 IoT

Tabuľka 1.1 – Technické parametre programovateľného mikrokontroléra Arduino Nano 33 IoT

Vlastnosť	Hodnota
Rozmery	45 x 18 mm
Microcontroller	SAMD21 Cortex®-M0+ 32bit low power ARM MCU
Pamäť CPU Flash	256 KB
Pamäť SRAM	32 KB
Rádio modul	u-blox NINA-W102
Konektivita	2,4GHz IEEE 802.11 b/g/n bezdrôtová sieť
	Bluetooth 4.2 BLE dual-mode
Operačné napätie	3,3 V
Prevádzková teplota	-20 °C až +70 °C



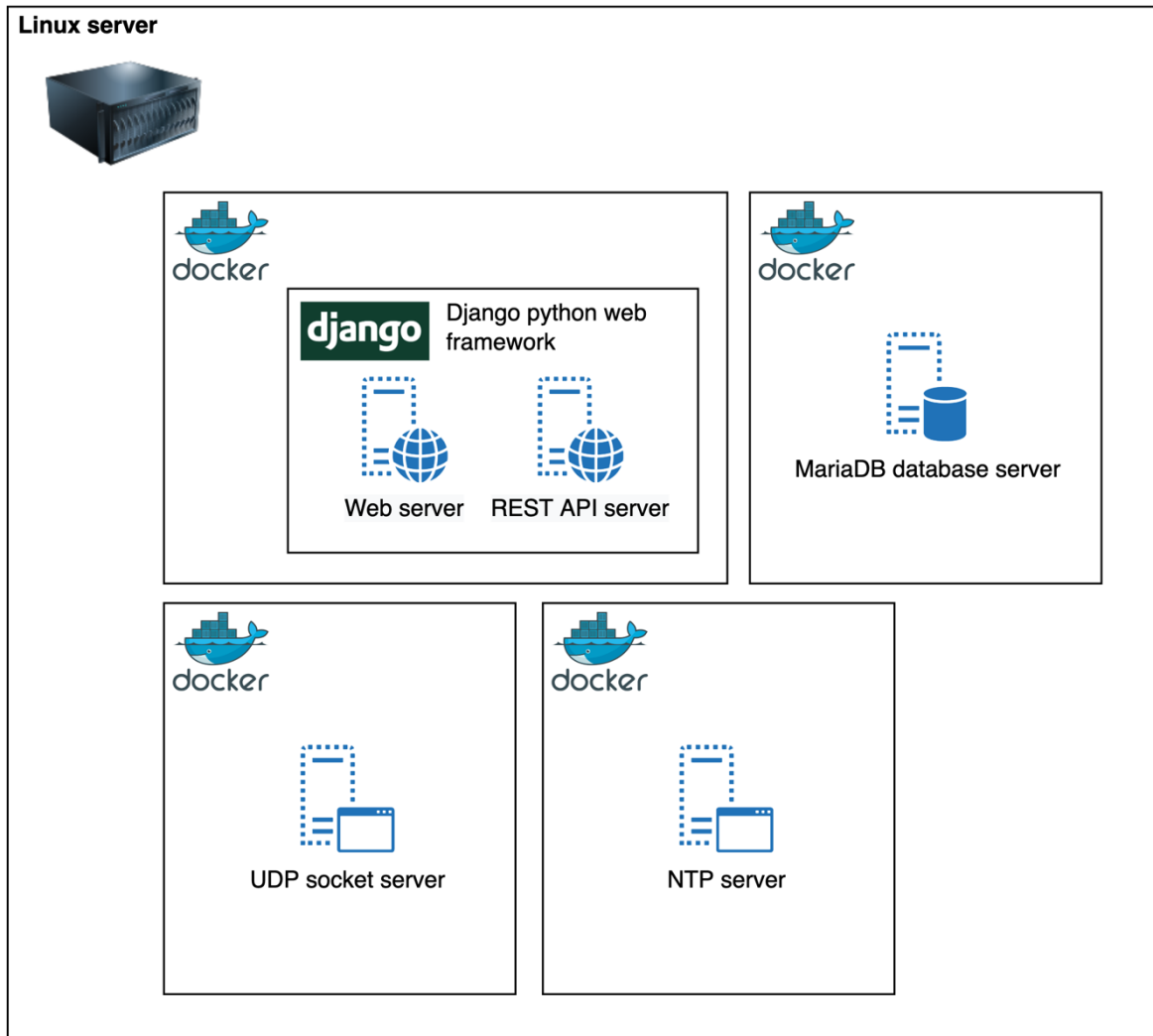
Obrázok 1.2 – Raspberry Pi® Zero W

Tabuľka 1.2 - Technické parametre minipočítača Raspberry Pi® Zero W

Vlastnosť	Hodnota
Rozmery	65 x 30 mm
Procesor	Broadcom BCM2710A1, quad-core 64-bit SoC (Arm Cortex-A53 @ 1GHz)
Pamäť	512 MB LPDDR2
Konektivita	2,4GHz IEEE 802.11 b/g/n bezdrôtová sieť
	Bluetooth 4.2 BLE
	Vstavaná anténa
	1 × rozhranie USB 2.0 s OTG
	Kompatibilné s 40pinovými rozširujúcimi doskami HAT
	Slot na microSD kartu
	Mini HDMI port
	CSI-2 konektor pre pripojenie kamery
Video	HDMI rozhranie
	Kompozitné video
	Dekódovanie H.264, MPEG-4 (1080p30)
	Kódovanie H.264 (1080p30)
	Grafika OpenGL ES 1.1, 2.0
Napájanie	5V DC, 2,5A
Prevádzková teplota	-20 °C až +70 °C

## 1.2. SERVER

Server je založený na operačnom systéme Linux distribúcia Ubuntu. Jednotlivé služby sú izolované vo virtuálnych kontajneroch. Tieto sú spravované pomocou servera Docker. To znamená, že jednotlivé aplikácie sú umiestnené a bežia v jednotlivých kontajneroch.



Obrázok 1.3 – Server

Server obsahuje nasledovné kontajnery:

- **Django python web framework** – poskytovanie dvoch služieb a to
  - web server, ktorý pomocou webového rozhrania umožňuje nastavovanie systému klientom a prezentáciu dát klientovi
  - REST API server, ktorý zabezpečuje komunikáciu zariadení v systéme s databázou
- **MariaDB databázový server**
  - Parametre systému

- dáta o Beacon čípoch získané pomocou čítačiek
- **UDP socket server** - zabezpečuje
  - prihlásenie čítacieho zariadenia do systému (vykoná sa vždy, keď sa čítacie zariadenie spustí a pripojí do siete)
- **NTP server** – zabezpečuje správny čas systému. Každá čítačka pri spustení si automaticky synchronizuje svoj lokálny čas. (čítačky na báze Arduino nemajú svoje vlastné počítanie času, preto je NTP server v systéme nevyhnutný)

### 1.3. BEACON ČIPY

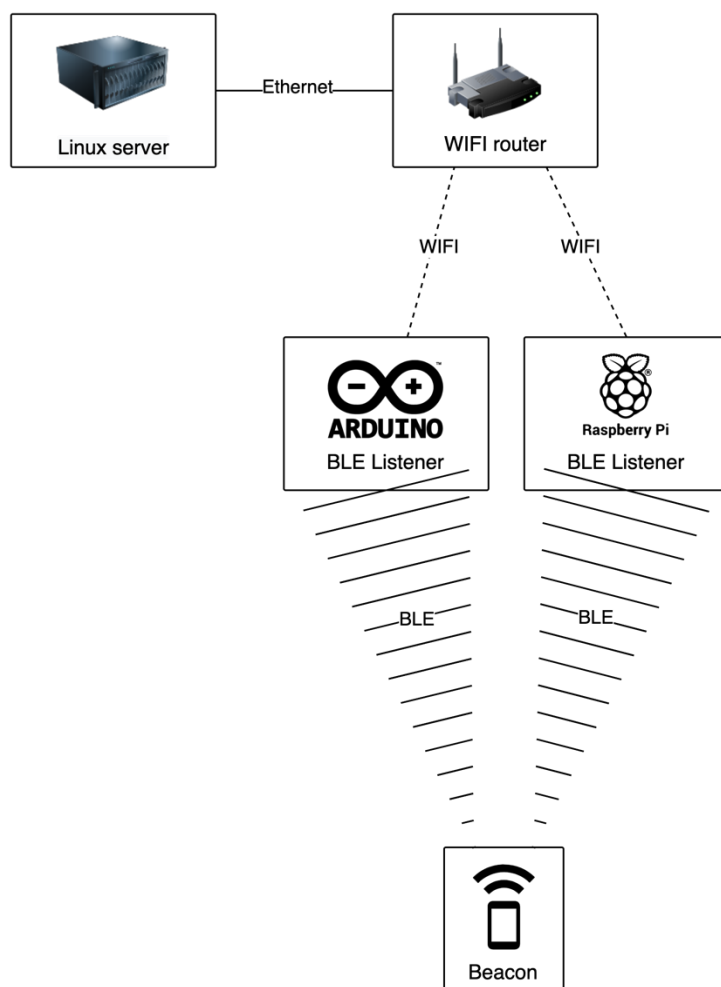
Z pohľadu vysielačích zariadení boli využité, vzhľadom na BLE komunikáciu, Beacon čipy, ktoré prostredníctvom BLE vysielať svoj identifikátor do blízkych prijímacích zariadení, v našom prípade čítačky na báze Arduina alebo Raspberry. V rámci identifikátora, ktorým je spravidla sériové číslo čipu, odosielať beaconové čipy aj hodnotu RSSI, pomocou ktorej je možné určiť v akej vzdialenosti sa nachádza čip od konkrétneho prijímacieho zariadenia. Beacon čipy majú viacero výhod, medzi ktoré nepochybne patrí ich veľkosť, životnosť (z pohľadu výdrže gombíkovej batérie, rádovo 6 mesiacov), rôznorodosť, jednoduchosť používania a mnoho iného. Trh ponúka naozaj rozsiahle možnosti, pričom medzi hlavných výrobcov môžeme zaradiť spoločnosti ako Gigaset, Feasycom či Holyiot. Prípady beaconových čipov od týchto spoločností sú zobrazené aj na obrázku 1.4.



Obrázok 1.4 – Príklad vyhotovenia beaconových čipov

## 2. SIEŤOVÉ ZAPOJENIE

Linux server obsahuje štyri kontajnery a tie sú pripojené do jednej virtuálnej siete, ktorú zabezpečuje Docker server a je nastavený pomocou aplikácie docker-compose. Celý fyzický server je pripojený pomocou ethernet prepojenia do WIFI smerovača. Jednotlivé služby kontajnerov sú v systéme viditeľné pomocou IP adresy servera a príslušného portu aplikácie. Na obrázku 2.1 je znázornená zjednodušená schéma sieťového zapojenia.



Obrázok 2.1 – Zjednodušená schéma sieťového zapojenia

Služby servera viditeľné v sieti systému nastavené nasledovne:

Webserver	http://<IP servera>:8000
REST API	http://<IP servera>:8000
UDP server	<IP servera>:26573
NTP server	<IP servera>:123

Čítacie zariadenia sú pripojené ku WIFI smerovaču pomocou bezdrôtového WIFI pripojenia. Čipy Beacon vysielajú informáciu o sebe čítačkám pomocou technológie Bluetooth Low Energy.

WIFI smerovač obsahuje DHCP server, ktorý sa stará o pridelenie IP adresy jednotlivým zariadeniam.

### **3. DÁTOVÝ TOK**

Dátový tok je zložený zo 7 hlavných aktivít, pričom každá aktivita (úloha) je nevyhnutná v dátovom procese.

#### **1. Zistenie umiestnenia servera v sieti.**

Každé čítacie zariadenie (Arduino aj Raspberry Pi) po pripojení do siete vyhľadá IP adresu servera. Toto sa vykoná pomocou broadcast udp paketu zaslaného do celej siete na port 26573. Ako správa paketu je použitý konfirmačný hash. Keď UDP server obdrží takúto správu odpovie späť pomocou iného daného konfirmačného hashu.

Pomocou tejto správy čítacie zariadenie zistí IP adresu servera.

#### **2. Nastavenie správneho lokálneho času čítačiek.**

Čítacie zariadenie si synchronizuje lokálny čas z NTP servera.

#### **3. Vysielanie Beacon čipov**

Beacon čipy vysielajú v pravidelných intervaloch (jedenkrát / sekunda) informáciu o sebe všetkými smermi. Pre systém sú dôležité MAC adresa čipu a RSSI (intenzita sily prijatého signálu). Čítacie zariadenia túto informáciu zachytávajú a spracúvajú.

#### **4. Spracovanie informácie z Beacon čipov**

Každé čítacie zariadenie 5 sekúnd počúva a zhromažďuje dáta z načítaných Beacon čipov. Následne tieto informácie spracuje a odošle pomocou UDP socketu na UDP server. Tento postup sa opakuje v nekonečnom cykle.

Arduino čítacie zariadenie obsahuje pre WIFI a Bluetooth jedno spoločné vysielacie zariadenie a nie je možné ho využívať súčasne. Preto je nutné pred 5-sekundovým načítaním Beaconov vypnúť WIFI a zapnúť Bluetooth. Naopak pri odosielaní dát na UDP server sa musí Bluetooth vypnúť a WIFI zapnúť.

Pri Raspberry PI Zero W toto prepínanie nie je nutné.

#### **5. Odoslanie Beacon údajov na REST API server**

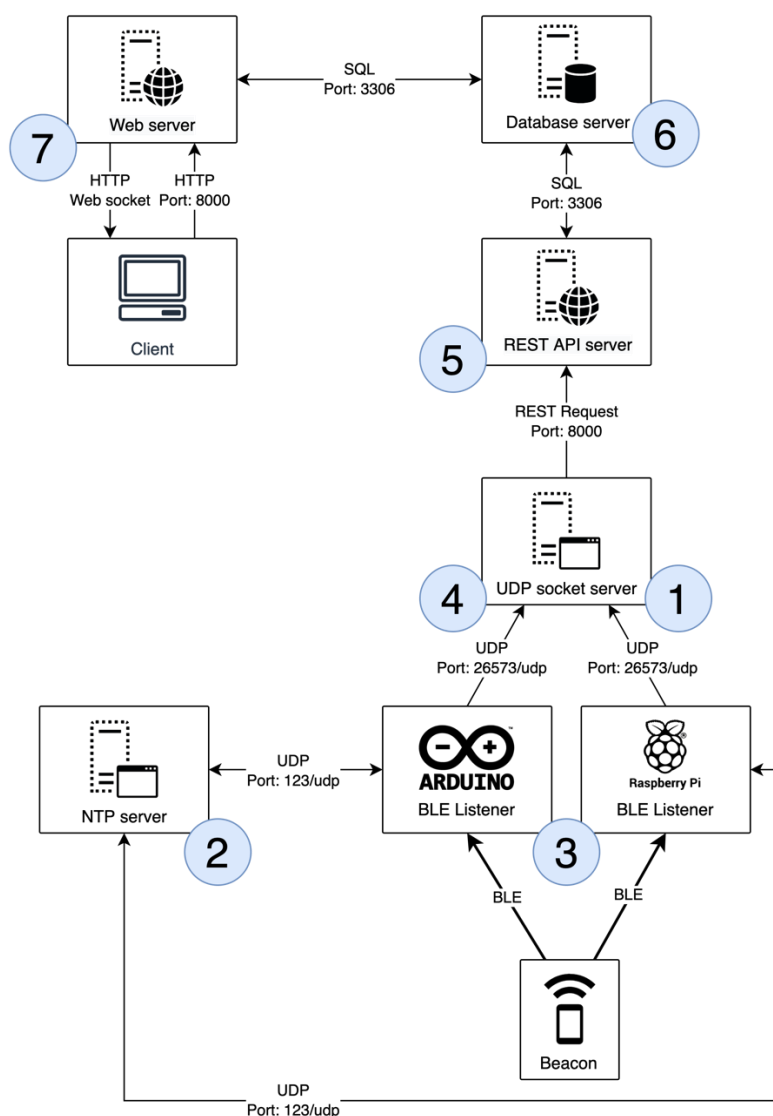
UDP server po prijatí socketov z čítacieho zariadenia tieto upraví a odošle na REST API server.

## 6. Uloženie do databázy

REST API server po prijatí požiadavky (requestu) uloží zodpovedajúce dáta do SQL databázy.

## 7. Web server a webové rozhranie

Web server pomocou webového rozhrania sprostredkúva nastavovanie parametrov systému a prezentuje získané dáta z databázy klientovi cez webové rozhranie.

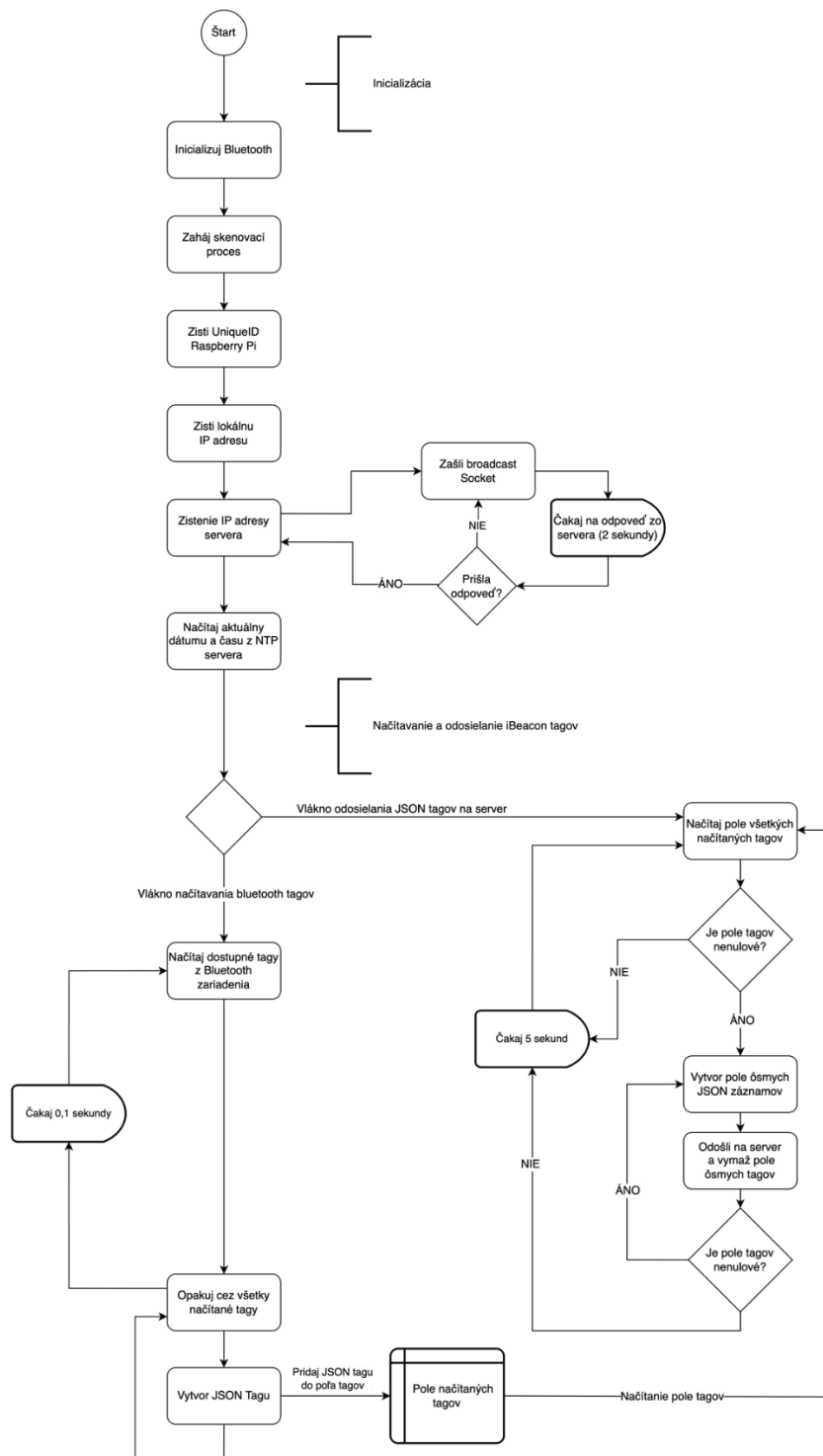


Obrázok 2.2 – Webové rozhranie



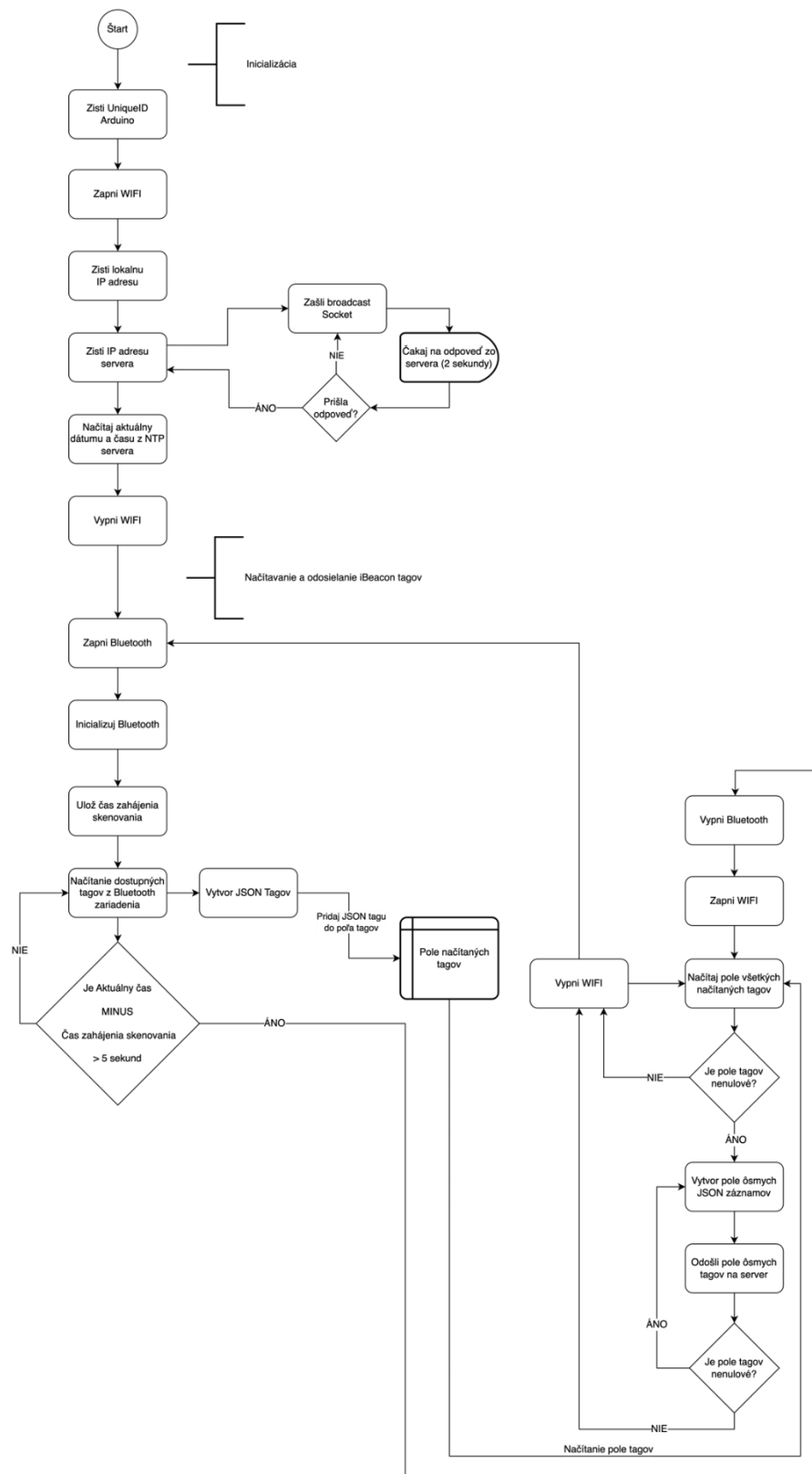
**Príloha 1 Modely technického riešenia**

Na nasledujúcich obrázkoch P1 a P2 je zobrazený funkčný (procesný) diagram (model) technického riešenia.

**Raspberry Pi**

**Obrázok P1 – Raspberry Pi® - model**

## Arduino



Obrázok P2 – Arduino - model