

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## NÁVRH VIZUALIZAČNÍ APLIKACE PRO BEZDRÁTOVÉ SENZOROVÉ SÍTĚ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ KOČAN

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# NÁVRH VIZUALIZAČNÍ APLIKACE PRO BEZDRÁTOVÉ SENZOROVÉ SÍTĚ

WIRELESS SENSOR NETWORK VISUALISATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ KOČAN

VEDOUcí PRÁCE

SUPERVISOR

Ing. FRANTIŠEK ZBOŘIL, Ph.D.

BRNO 2013

## **Abstrakt**

Cílem této bakalářské práce je vytvořit webovou aplikaci, která by sloužila k vizualizaci, tzn. grafické znázornění dat ze senzorů bezdrátové sensorové sítě na mapovém podkladu. Kromě toho má poskytovat možnosti sledování historických dat ze senzorů pomocí grafů. Základem aplikace jsou jazyky JavaScript a HTML spolu s využitím dalších knihoven jako jqwidgets, jQuery, amCharts a aplikačního rozhraní služby Google Maps. V této práci je popsán úvod do bezdrátových sensorových sítí, popis architektury aplikace, návrh a implementace aplikace, jakož i její testování a další vývoj.

## **Abstract**

The aim of this work is to create a web application that would serve to visualize, ie. graphical representation of data from sensors on maps. In addition, provide tools to track historical data from the sensors with the use of diagrams. The basis of application are languages JavaScript and HTML together with the use of other libraries such as jqwidgets, jQuery, amCharts and application interface of Google Maps. This paper describes an introduction to wireless sensor networks, description of application architecture, design and implementation of application, as well as the testing and further development.

## **Klíčová slova**

bezdrátové sensorové síte, vizualizační aplikace, sledování veličin, Google Maps, Cosm, jQuery

## **Keywords**

wireless sensor networks, visualization application, monitoring, Google Maps, Cosm, jQuery

## **Citace**

Tomáš Kočan: Návrh vizualizační aplikace pro bezdrátové sensorové síte, bakalářská práce, Brno, FIT VUT v Brně, 2013

# Návrh vizualizační aplikace pro bezdrátové senzorové sítě

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Františka Zbořila, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Tomáš Kočan

7. května 2013

## Poděkování

Chcel by som sa poďakovať Ing. Františkovi Zbořilovi, Ph.D. za vedenie práce a Ing. Lubomírovi Mrázovi za cenné rady a pripomienky pri jej tvorbe.

© Tomáš Kočan, 2013.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Teoretický úvod</b>	<b>4</b>
2.1	Bezdrôtové senzorové siete . . . . .	4
2.1.1	História . . . . .	4
2.1.2	Technológia . . . . .	5
2.1.3	Využitie . . . . .	6
<b>3</b>	<b>Architektúra aplikácie</b>	<b>8</b>
3.1	Architektúra . . . . .	8
3.1.1	Cosm . . . . .	9
3.1.2	Vizualizačná aplikácia . . . . .	9
3.2	Analýza požiadaviek na aplikáciu . . . . .	9
3.2.1	Požiadavky . . . . .	9
<b>4</b>	<b>Návrh aplikácie</b>	<b>11</b>
4.1	Užívateľské rozhranie . . . . .	11
4.2	Mapový podklad . . . . .	12
4.2.1	Udalosti spojené so značkami . . . . .	13
4.3	Horný panel . . . . .	14
4.3.1	Režim now . . . . .	14
4.3.2	Režim now(socket) . . . . .	14
4.3.3	Režim history . . . . .	15
4.3.4	Ďalšie nástroje . . . . .	15
4.4	Pravý panel . . . . .	16
4.4.1	Vyhľadávanie . . . . .	16
4.4.2	Vytvorenie uzlu . . . . .	19
4.4.3	Vytvorenie budovy . . . . .	19
4.4.4	Informácie o kľúči . . . . .	20
4.5	Graf . . . . .	21
4.5.1	Použitie . . . . .	21
4.5.2	Typ grafu . . . . .	22
4.5.3	Upozornenia pre užívateľa . . . . .	23
<b>5</b>	<b>Implementácia</b>	<b>24</b>
5.1	Model, view, controller . . . . .	24
5.2	Použitie technológií na strane servera (Cosm/Pachube) . . . . .	25
5.2.1	Komunikácia so službou Cosm . . . . .	25

5.3	Použité technológie na klientskej strane . . . . .	27
5.3.1	HTML5,CSS . . . . .	27
5.3.2	Javascript . . . . .	28
5.4	Nástroje a aplikácie využívané pri implementácií . . . . .	29
5.5	Optimalizácie . . . . .	29
5.5.1	Funkcia setTimeout . . . . .	29
5.5.2	Kompresia zdrojových kódov . . . . .	29
5.5.3	Optimalizácia JavaScript kódu . . . . .	29
5.5.4	Systémové nároky . . . . .	30
<b>6</b>	<b>Testovanie</b>	<b>31</b>
6.1	1. fáza testovania . . . . .	31
6.1.1	Kontrola správnosti dát . . . . .	31
6.1.2	Testovanie UI . . . . .	32
6.2	2. fáza testovania . . . . .	32
6.3	Ďalší vývoj . . . . .	33
<b>7</b>	<b>Záver</b>	<b>34</b>
<b>A</b>	<b>Obsah DVD</b>	<b>36</b>

# Kapitola 1

## Úvod

Cieľom práce bolo vytvoriť aplikáciu OutdoorView, ktorá by slúžila na vizualizáciu údajov zo senzorov a samotných bezdrôtových senzorových sietí v exteriéroch. Bezdrôtové jednotky, ktoré vytvárajú bezdrôtové senzorové siete môžu byť umiestnené prakticky kdekoľvek. Ich hlavným cieľom je snímanie a zber dát z okolitého prostredia. Potrebu takéhoto snímania dát zo senzorov na geograficky rozsiahlom území ukázala nedávna udalosť v japonskej Fukushime. Štandardné riešenie pomocou statických meracích zariadení (hotspotov) sa ukázalo ako geograficky nedostatočné a predovšetkým nákladné. Reakciou na túto udalosť bolo založenie projektu Safecast. Hlavnou myšlienkou je snímanie rádioaktivity pomocou cenovo dostupných senzorov veľkým počtom jednotlivcov, ktorí nasnímané dáta zdieľajú prostredníctvom internetu. Tým je možné vytvoriť geograficky detailnú mapu meranej veľičiny v ideálnom prípade až v celosvetovom meradle. Pre tento typ zdieľania existuje termín „community sensing“. Podobný projekt zameraný na kontrolu kvality ovzdušia má názov AirQualityEgg, kde malá senzorová jednotka sníma hodnoty CO, NO<sub>2</sub>, NO<sub>x</sub> a prašné častice PM.

Spoločným problémom spomenutých projektov je systém vizualizácie dát. Dáta sú obvykle uložené v databáze, ale zobrazenie dát v reálnom čase a hlavne intuitívnym spôsobom pre človeka zatiaľ chýba. Splniť tieto požiadavky si za cieľ dáva táto bakalárska práca.

Ďalším dôvodom jej vzniku bolo začlenenie do súboru ďalších nástrojov, ktorých úlohou je vizualizácia dát zo senzorov umiestnených v exteriéroch aj interiéroch. Popis tohto konceptu a architektúry sa nachádza v kapitole 3.

Nasledujúca kapitola sa bude zaoberať návrhom OutdoorView a takisto objasní všetky možnosti, ktoré ponúka.

V kapitole 5 sa budeme venovať samotnou implementáciou a v kapitole 6 testovaním aplikácie.

V závere bude nasledovať zhrnutie a zhodnotenie práce a takisto návrh ďalších rozšírení a možných využití aplikácie.

# Kapitola 2

## Teoretický úvod

Táto kapitola je zameraná predovšetkým na teoretické poznatky o senzorových sieťach, ich histórii, technológií a možných využitíach. Táto časť čerpá najmä z publikácií [1, 9].

### 2.1 Bezdrôtové senzorové siete

Senzorové siete (celosvetovo označované WSN z anglického Wireless Sensor Networks) pozostávajú z veľkého počtu malých senzorových uzlov s vlastným napájaním. Zbierajú dáta alebo detekujú špeciálne udalosti, navzájom bezdrôtovo komunikujú s konečným cieľom odoslať údaje na centrálnu stanicu. Tieto tri činnosti – meranie, spracovanie a komunikácia, sú kľúčové elementy týchto malých zariadení – uzlov. Vďaka tomu neustále narastá počet možností ich využitia. Faktory, ktoré ovplyvňujú architektúru senzorových sietí [1]

- Schopnosť vysporiadať sa so zlyhaniami niektorých uzlov
- Nízke výrobné náklady
- Prenositelnosť uzlov a s tým súvisiaca častá zmena topológie
- Možnosť nasadenia v sťažených prírodných podmienkach
- Malá energetická náročnosť, využitie batérií alebo možnosť využiť iné sebestačné zdroje energie
- Škálovateľnosť

#### 2.1.1 História

Senzorové siete sa do povedomia prvýkrát dostávajú koncom 90-tych rokov [5] spolu s príchodom senzorových uzlov a operačného systému TinyOS (operačný systém určený predovšetkým pre senzorové siete [9]). Odvtedy ich vývoj neustále napreduje a nachádzajú si stále nové využitia. Prvotný impulz k vývoju bezdrôtových senzorových sietí prišiel z vojenského prostredia. A to kvôli požiadavke na sledovanie fyzikálnych veličín na rozľahlom území. Náhodne rozmiestnené veľké množstvo senzorov vyhodnocovalo zmeny teploty, vibrácií, tlaku, čím včas upozorňoval o presune nepriateľských vojsk. Táto technológia si v krátkej dobe našla využitie aj v civilnom sektore - environmentálne aplikácie, automatizácia budov, doprava, ...

## 2.1.2 Technológia

Senzorová sieť pozostáva z uzlov, ktoré obsahujú tieto komponenty [9, 1]:

1. Senzorová jednotka – môže obsahovať jeden alebo viacero senzorov.
2. Procesorová jednotka – najčastejšie sa jedná o malý mikrokontrolér.
3. Prijímacio-vysielacia jednotka – rádiový transceiver<sup>1</sup> alebo nejaký iný druh bezdrôtového komunikačného zariadenia.
4. Napájacia jednotka – zdroj energie, zvyčajne batéria.

Základnou myšlienkou WSN je maximálna úspora energie a vysoká flexibilita, tomu je podriadený celý systém, teda ako výpočetný výkon uzlu a programové prostriedky, tak prenosové protokoly. Uzol využíva rôzne hladiny „spánku“, kedy výkon procesoru, pamäť a periférií je riadený podľa aktuálnej potreby (snímanie veličín, spracovanie dát, komunikácia, nečinnosť atď.) [5].

Počet uzlov v typickej sensorovej sieti sa môže pohybovať od stoviek až po tisícky. Topológia je vytváraná dynamicky a teda ani poloha senzorov nemusí byť vopred daná. Veľkosť takéhoto jedného senzoru sa môže pohybovať od veľkosti niekoľkých desiatok centimetrov až po pár milimetrové zariadenia. Taktiež aj cena sa môže pohybovať od stoviek eur až po centy [9] v závislosti od veľkosti sensorovej siete a komplexnosti jednotlivých uzlov a senzorov. Ďalšími charakteristikami sú nároky na zdroje energie, pamäť, výpočtovú a prenosovú rýchlosť.

Senzorové siete sa odlišujú od tradičných bezdrôtových adhoc sietí vo viacerých aspektoch [9, 1]:

- Počet uzlov v sensorových sieťach môže byť niekoľkonásobne vyšší a taktiež je ich umiestnenie oveľa hustejšie. Uzly v sensorových sieťach sú náchylné na poruchu, a preto je sieť navrhovaná tak, aby bola dostatočne robustná voči takýmto zlyháním. Taktiež aj topológia v takýchto sieťach sa mení veľmi často. Ako komunikačný vzor sa najčastejšie využíva širokopásmové vysielanie (broadcast).
- Uzly sú väčšinou sieťové prvky typu transceiver, nie len prijímač.
- Sú tu isté obmedzenia na napájanie, výpočtovú silu a dostupnú pamäť kvôli snahe o čo najmenšiu veľkosť a obmedzeným možnostiam nabitia (resp. napájania).
- Uzly častokrát nemajú ani globálnu identifikáciu kvôli veľkej réžii, ktorá by s tým súvisela.

Výsledkom týchto rozdielností od tradičných bezdrôtových adhoc sietí bolo niekoľko výziev, s ktorými sa bolo treba vysporiadať. Ako napríklad manažment spotreby, smerovanie, tvorba požiadaviek, synchronizácia, lokalizácia, ... Postupom času sa vynárali nové požiadavky na operačný systém, podporu programovacích jazykov, virtuálnych strojov, simulátorov, debugovacích nástrojov a taktiež aj vysporiadanie sa s bezpečnostnými rizikami.

---

<sup>1</sup>jedno zariadenie obsahujúce ako vysielateľ, tak aj prijímač

### 2.1.3 Využitie

Táto kapitola čerpá najmä z [1].

Senzorové siete môžu pozostávať z veľa rozdielnych typov senzorov ako sú napríklad seizmické, magnetické, termálne, infračervené, pohybové, akustické alebo radar. Sú schopné monitorovať veľké množstvo okolitých podmienok, ako:

1. teplota
2. vlhkosť
3. pohyb
4. svetelné podmienky
5. tlak
6. pôdne zloženie
7. úrovne hluku
8. prítomnosť, resp. neprítomnosť určitých objektov
9. mechanické opotrebenie objektov
10. alebo aktuálne hodnoty rýchlosti, smeru, veľkosti, ...

Senzorové uzly sú určené na nepretržité snímanie, detekciu špecifických udalostí alebo sledovanie polohy. Koncept malých senzorov schopných bezdrôtovo komunikovať prináša veľmi veľa možností na ich využitie. Oblasti, v ktorých sú využívané môžeme rozdeliť na: vojenské, environmentálne, zdravotné účely, komerčná ale aj domáca oblasť. Toto rozdelenie je možné rozšíriť ešte o niekoľko ďalších kategórií ako výskum vesmíru, chemické spracovanie a ochrana pred živelnými pohromami.

- Vojenské účely – bezdrôtové senzorové siete môžu byť integrovanou súčasťou vojenského riadenia, kontroly, komunikácie, sledovania alebo aj zameriavacích systémov vďaka možnosti rýchleho nasadenia, žiadnej nutnosti precíznej topológie a tolerancí chýb. Keďže sú senzorové siete založené na hustom rozložení senzorov a ich cena je veľmi nízka, tak prípadné zničenie niekoľkých uzlov neovplyvní vojenské operácie tak ako deštrukcia tradičných senzorov. Vďaka tomuto sú senzorové siete vhodným adeptom na nasadenie na bojiská. Typickým vojenským využitím je monitorovanie spojeneckých jednotiek, monitorovanie stavu zariadení a munície, sledovanie bojísk, prieskum nepriateľských jednotiek a terénu, zameriavanie, zhodnotenie následkov útoku a detekcia nukleárnych, biologických a chemických hrozieb. Každý vojak, vozidlo, vybavenie alebo munícia môže byť vybavená senzorom, ktorá neustále hlási ich stav. Tieto hlásenia sú zbierané centrálnymi uzlami (nazývané sink nodes) a odosielané veliteľom. Dáta môžu byť odosielané ďalej v riadiacej hierarchii a agregované a posudzované s dátami z iných oblastí.
- Environmentálne využitie – zahŕňa sledovanie pohybu vtákov, menších zvierat, hmyzu, ďalej to je monitorovanie podmienok, ktoré majú vplyv na úrodu a hospodárske zvieratá, kontrola zloženia pôdy, vody, vzduchu, detekcia lesných požiarov, povodní. Meteorologický a geofyzikálny výskum, ... Jednou z novinek je využívanie v tzv. presnej agrikultúre, kedy sa pomocou senzorov sleduje úroveň pesticídov, zloženie pôdy a prípadne sa vzdialene reguluje.

- Zdravotné účely – senzorové uzly ponúkajú možnosti na monitorovanie pacientov, diagnostiku, administráciu liekov v nemocniciach (znižuje šancu na predpísanie zlého lieku ); vzdialené monitorovanie fyziologického stavu človeka; sledovanie pacientov a doktorov v nemocnici – v tomto prípade majú pacienti viaceré ľahké senzory, ktoré zaznamenávajú napr. pulz alebo krvný tlak.
- Domáce využitie – senzorové siete si nachádzajú využitie aj v domácom prostredí, senzory sa môžu nachádzať v domácich spotrebičoch ako vysávače, mikrovlnné rúry, chladničky. Všetky tieto senzory môžu spolu vzájomne komunikovať a odosielať údaje s pomocou internetu alebo satelitov. To zároveň umožňuje vzdialené ovládanie týchto zariadení.
- Komerčné využitie – senzorové siete nachádzajú bohaté využitie aj v komerčnej sfére. Príklady využitia: monitorovanie únavy materiálu, spravovanie inventáru, monitorovanie kvality produktov, riadenie klimatizácie v budovách, interaktívne hračky, nástroj proti krádeži áut.

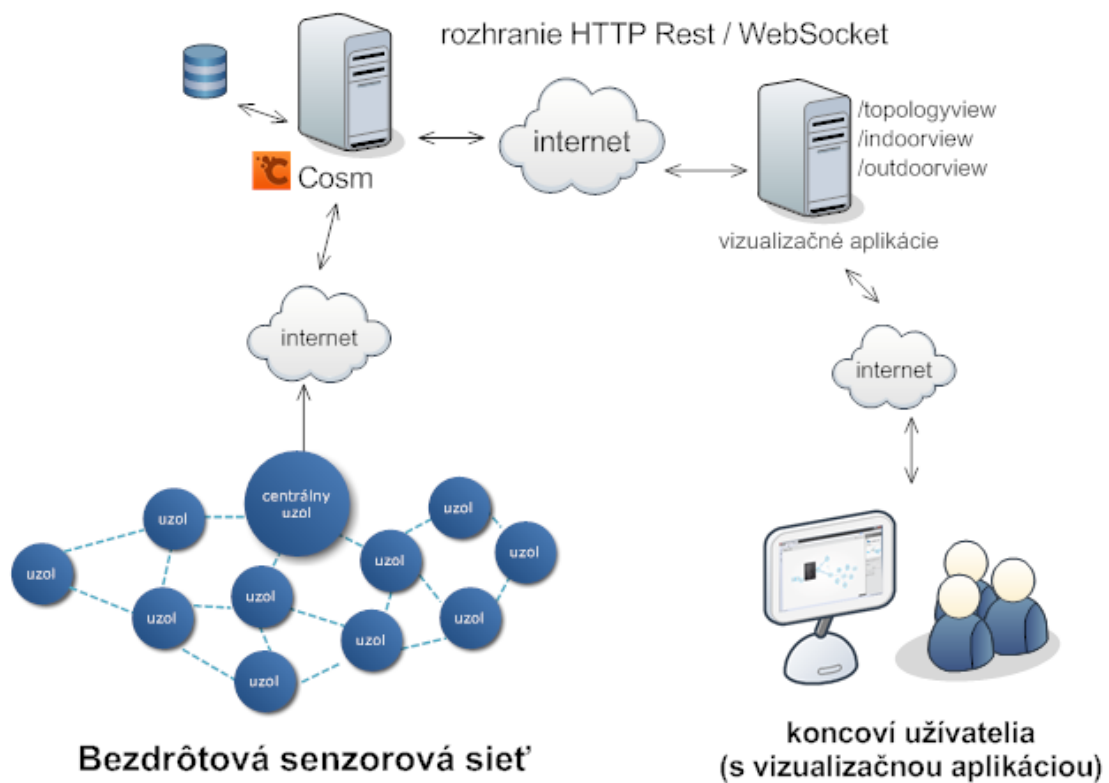
## Kapitola 3

# Architektúra aplikácie

Táto kapitola je zameraná na popis základnej architektúry aplikácie, jej konceptu a zaradenie do rady nástrojov pre prácu s bezdrôtovými sieťami vyvíjanými na VUT. Ďalšia časť sa zaoberá požiadavkami na túto aplikáciu a stanovením základných bodov, ktoré má spĺňať. Tieto body a požiadavky vychádzajú z konzultácií s potenciálnymi používateľmi aplikácie.

### 3.1 Architektúra

Na Ústave Telekomunikácií FEKT VUT bola vyvinutá aplikácia, ktorá umožňuje zobrazenie senzorových sietí v interiéroch. Táto práca je rozšírením vizualizačného systému o senzorové uzly umiestnené v exteriéroch. Celková architektúra je možná vidieť na obr. 3.1.



Obrázek 3.1: Architektúra systému vizualizácie senzorových sietí. Obrázok prevzatý z <http://wislab.cz/our-work/research-project-2010-2013-adaptive-wireless-sensor-networks-awsn>

Z obrázku je možné vidieť, že existuje množstvo uzlov vytvárajúcich bezdrôtovú senzorovú sieť. Spomedzi nich je jeden centrálny, ktorý odosiela merané údaje pomocou internetu na servery služby COSM (bližšie popísané v sekcii 3.1.1). Údaje z databázy Cosm sú následne požadované vizualizačnou aplikáciou (bližšie vid' 3.1.2), ktorá poskytuje rozhranie pre koncových používateľov.

### 3.1.1 Cosm

Cosm (predtým Pachube) je internetová služba zameraná na zdieľanie senzorických dát. Poskytuje rozhranie a databázu pre prácu a ukladanie dát zo senzorov a senzorových sietí. Každá takáto senzorická jednotka je v službe reprezentovaná ako **uzol** (z angl. feed).

Uzol je jednoznačne identifikovaný unikátnym číslom, má svoj názov, polohu, značky podľa ktorých sa dá vyhľadávať a predovšetkým obsahuje niekoľko tzv. **dátových tokov** (z angl. datastream). Každý jeden tento dátový tok reprezentuje jeden senzor, ktorý sníma jednu konkrétnu veličinu, ktorú si užívateľ tejto aplikácie praje vizualizovať. (môže sa jednať o teplotu, napätie, ...)

### 3.1.2 Vizualizačná aplikácia

Ako bolo spomenuté OutdoorView je súčasťou väčšej aplikácie schopnej sledovať ako exteriérové, tak aj interiérové uzly. Architektúra tejto rady aplikácií je vidieť na obr. 3.2. Jednou z týchto aplikácií je OutdoorView, ktorý sleduje exteriérové uzly (podkladom na zobrazovanie sú google mapy) a vytvára premostenie do ďalšej súčasti tohto systému – IndoorView.

IndoorView vizualizuje vnútorné uzly a teda aj podkladom na zobrazovanie sú pôdorysy konkrétnych budov. Posledným z tejto rady aplikácií je nástroj TopologyView, ktorý znázorňuje fyzickú topológiu senzorových sietí postavenú na sile rádiového signálu medzi jednotlivými uzlami siete.

## 3.2 Analýza požiadaviek na aplikáciu

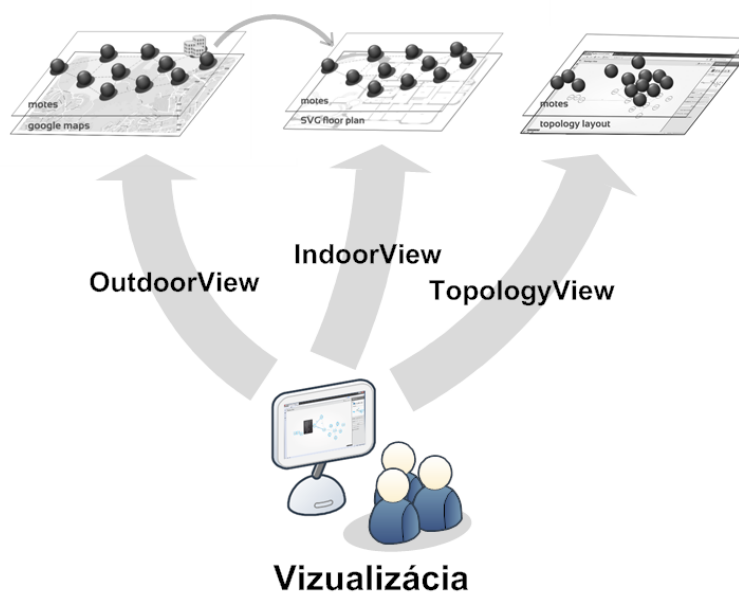
OutdoorView sa má stať rovnocennou aplikáciou k nástroju IndoorView, z čoho pre ňu vyplývajú aj hlavné požiadavky, a to aby ponúkala rovnakú funkcionálnosť a podobné užívateľské rozhranie ako IndoorView.

### 3.2.1 Požiadavky

Aplikácia má byť nasadená online, má byť multiplatformná, jednoducho prenositeľná a umožňovať komunikáciu s ostatnými online službami, konkrétne s aplikačným rozhraním služby Cosm/Pachube a Google Maps. Z týchto požiadaviek vyplýva nutnosť vyvíjať aplikáciu ako webovú aplikáciu.

Má poskytovať možnosť vyhľadávať uzly podľa zadaných parametrov, prípadne tieto parametre kombinovať. Z nájdených uzlov filtrovať exteriérové a zobrazovať na mape s tým, že užívateľ má mať možnosť si jednoducho prehliadnúť všetky merané veličiny z týchto uzlov. Okrem toho sa priamo na mape má zobrazovať aktuálna hodnota užívateľom vybranej veličiny.

Od aplikácie sa očakáva, aby nielen pasívne sledovala uzly, ale mala možnosť ich meniť alebo vytvárať. Užívateľ teda má mať možnosť editovať vlastnosti uzlov, meniť ich pozíciu, mazať ich.



Obrázek 3.2: Nástroje na vizualizáciu dát zo senzorov

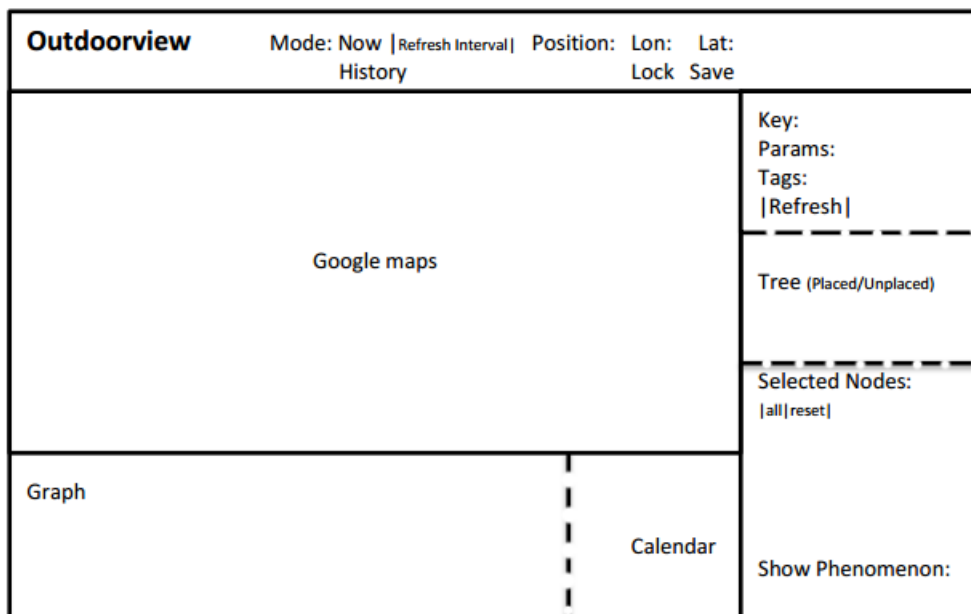
Ďalšou dôležitou súčasťou má byť možnosť sledovať historické dáta z rôznych senzorov v časových rozsahoch definovaných užívateľom a taktiež porovnávať dáta z viacerých senzorových uzlov pomocou grafov.

## Kapitola 4

# Návrh aplikácie

Hlavným cieľom aplikácie je vizualizácia čo možno najväčšieho množstva informácií o senzorovej sieti a to čo možno najprehľadnejším spôsobom. Od toho sa odvíja aj samotný návrh užívateľského rozhrania.

### 4.1 Užívateľské rozhranie

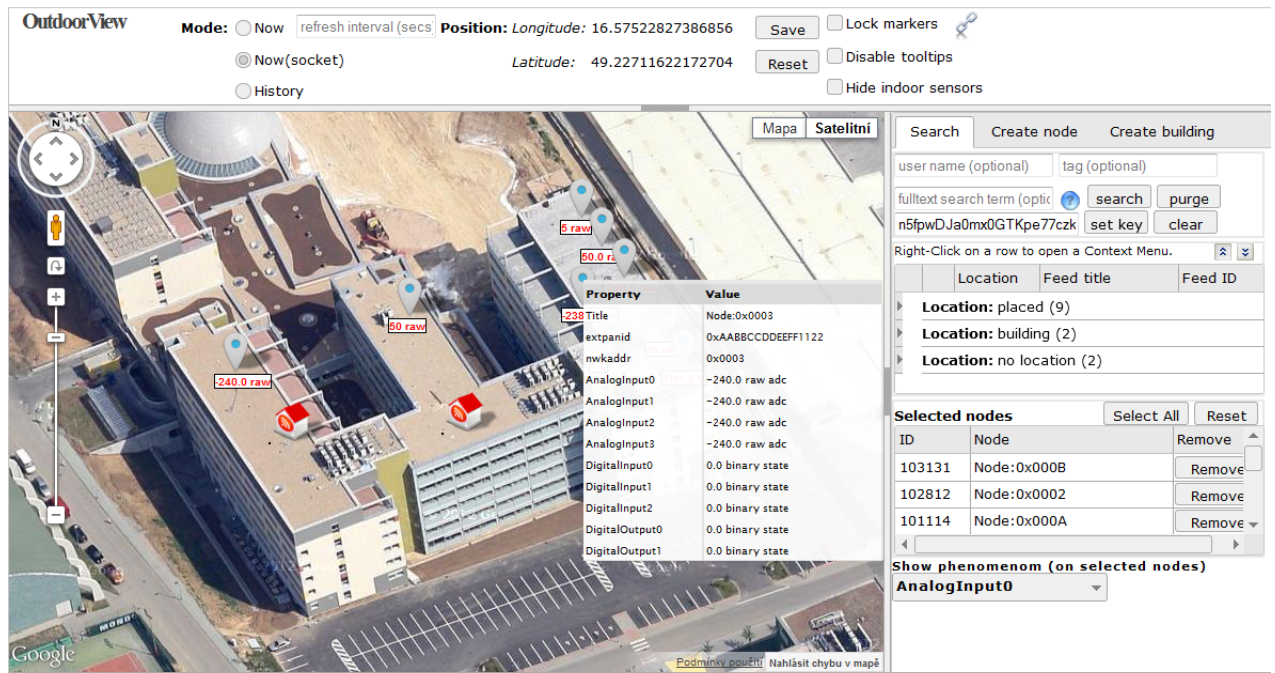


Obrázek 4.1: Prvotný návrh aplikácie

Užívateľské rozhranie je opticky rozdelené na štyri časti, pričom bočné panely sa dajú minimalizovať, čím sa maximalizuje práve mapový podklad, ktorý je dominantným prvkom celej aplikácie. Aplikácia nemá pevné rozmery (prispôsobuje sa rozmerom zobrazovacieho zariadenia), čím je použiteľná nielen na počítačoch, ale aj na mobilných zariadeniach.

Aplikácia sa môže nachádzať v troch rôznych stavoch a to now, now(socket) a history. Na obrázku 4.2 je vidieť aplikácia, kde boli vyhľadane všetky uzly od konkrétneho užívateľa,

ďalej je tu pri každej značke zobrazená hodnota zo senzoru *AnalogInput0* a zobrazené všetky merané údaje na uzle Node 0x0003.



Obrázek 4.2: Vyhľadané a označené všetky uzly od konkrétneho užívateľa

## 4.2 Mapový podklad

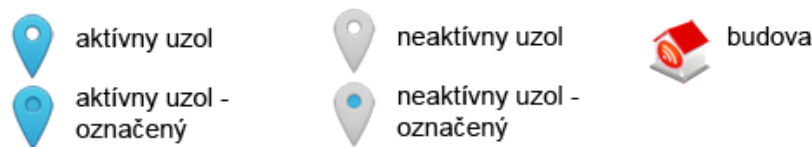
Ako mapový podklad slúžia google mapy, ktoré poskytujú pomerne široké a postačujúce aplikačné rozhranie pre potreby tejto aplikácie. To znamená plnohodnotné pokrytie oblastí, kde sa môžu uzly vyskytovať, možnosť prepínať medzi druhmi máp, možnosť približovať a oddaľovať pohľad na mapu a takisto možnosť vytvoriť značky a pomocné okná informujúce o uzloch. Na zobrazenie uzlov slúžia rôzne typy značiek (markerov):

- building - predstavujúci budovu.
- live feed - predstavuje *aktívny uzol*, teda taký, ktorého posledná nameraná hodnota je stará najviac 15 minút.
- frozen feed - predstavuje *neaktívny uzol*, od ktorého neprišli žiadne údaje za posledných 15 minút.

Aktívne aj neaktívne uzly môžu byť označené užívateľom, graficky sa prejaví modrým lemovaním. Táto logika však nie je štandardne súčasťou Google Maps API, preto boli použité rozširujúce knižnice *markerwithlabel*<sup>1</sup> a *infobox*<sup>2</sup>. Keďže však vykresľovanie týchto značiek spolu so všetkými informáciami je pomerne pomalá operácia (v porovnaní s ostatnými), boli použité iste heuristiky podrobnejšie popísané v časti 5.5.

<sup>1</sup><http://google-maps-utility-library-v3.googlecode.com/svn/trunk/markerwithlabel/docs/reference.html>

<sup>2</sup><http://google-maps-utility-library-v3.googlecode.com/svn/trunk/infobox/docs/reference.html>



Obrázek 4.3: Rôzne typy a stavy značiek

#### 4.2.1 Udalosti spojené so značkami

**prechod myšou nad značkou** pri prejení myšou nad značkou sa vytvorí tabuľka s informáciami o uzle, konkrétne jeho názov a hodnoty všetkých meraných veličín daným uzlom (originálne nazývané datastream - dátový tok) spolu s jednotkou meranej veličiny. Údaje zaznamenané v posledných 15 minútach sú zvýraznené modrou farbou.

**klik ľavým tlačidlom na aktívne/neaktívne uzly** - po stlačení ľavého tlačidla sa daná značka označí, tzn graficky bude zvýraznená na mape (viď obr. 4.3), bude umožnené užívateľovi zobraziť graf k vybranému uzlu a taktiež v ďalšom paneli vybrať dátové toky, ktoré tento uzol zaznamenáva. Všetky udalosti spojené s označením značky budú ďalej rozobraté v kapitole 4.4.

**klik pravým tlačidlom** - po stlačení pravého tlačidla sa užívateľovi vytvorí kontextová ponuka, kde má možnosť si vybrať jeden z dátových tokov, ktorého aktuálne hodnoty sa budú zobrazovať priamo na mape pri značke. V historickom móde je táto možnosť zablokovaná.

**drag&drop - presun značky** - užívateľ môže každú značku, ktorá reprezentuje konkrétny uzol, presúvať po mape. Pokiaľ nemá k takejto akcii potrebné práva, tak sa obnoví pôvodná poloha značky a užívateľ je upozornený. Samotný presun značky ešte neznamená uloženie aktuálnej polohy, na to slúži tlačidlo **Save position(s)** (viď 4.3).

Zvláštnym prípadom je interakcia s budovami. Keďže ich úlohou je vytvárať odkaz na aplikáciu IndoorView, tak neobsahujú žiadne senzory, a teda je zablokovaný výber dátových tokov pomocou pravého tlačidla.

Stlačenie ľavého tlačidla neoznačí budovu, ale vyvolá kontextovú ponuku (viď obr. 4.4). Kde jednotlivé položky sú tvorené informáciami o budove. Teda počet poschodí a ich názvy

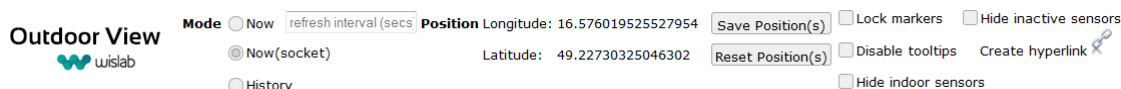


Obrázek 4.4: Dialógové okno na prepojenie s aplikáciou IndoorView

sú pod položkou **Floor**, parameter **Extpanid**<sup>3</sup> zahŕňa aj možnosť vybrať **none** v prípade, že tento parameter nie je známy. Poslednou možnosťou je výber užívateľa. Je možnosť vybrať buď vlastníka budovy alebo žiadneho.

Po stlačení tlačidla **Go** sa v novom okne (resp. novej záložke, podľa nastavení prehliadača užívateľa) načíta aplikácia **IndoorView** so zadanými parametrami. Súčasťou predaných parametrov je aktuálne nastavený kľúč v **OutdoorView**. Takže ak chce používateľ manipulovať s uzlami v tejto novej aplikácii, musí sa uistiť, že má zadaný vhodný kľúč.

## 4.3 Horný panel



Obrázek 4.5: Globálne nastavenia pre aplikáciu

Poskytuje možnosti na kontrolu aplikácie, prácu so značkami a možnosti na zmenu režimov, v ktorých je aplikácia. Môže byť v troch typoch režimov: **now**, **now(socket)** a **history**.

### 4.3.1 Režim now

V režime **now** sa získavajú namerané údaje s využitím štandardného REST rozhrania, tzn. v pravidelných intervaloch odosiela aplikácia požiadavku na COSM servery, spracuje odpoveď a vhodne zobrazí. Jedná sa teda o dotazovací mód (angl. **polling**) založený na protokole **HTTP**.

Tento mód bol vyvíjaný s ohľadom na to, aby bolo možné využiť vlastné servery (teda nie služby COSM), kde je implementácia REST rozhrania jednoduchšia ako socketové rozhranie. V tomto režime má užívateľ možnosť zadať jeden parameter a to interval, v ktorom sa budú požadovať dáta zo serverov. Horná hranica nie je limitovaná, minimum je 1 sekunda z dôvodu, aby neboli požiadavky z **OutdoorView** považované za DoS útok a tým pádom blokové alebo odkladané.

### 4.3.2 Režim now(socket)

Režim **now(socket)** funguje na princípe websocketu, pomocou ktorého sa automaticky odosielajú aktuálne údaje. Tie sú okamžite prijaté a spracované. V praxi sa aplikácia zaregistruje na odber dát pre konkrétny dátový tok (**datastream**) alebo uzol (**feed**), a následne pri zmene dát je táto informácia odoslaná automaticky aplikácii.

Na lepšie pochopenie rozdielu medzi režimami **now** a **now(socket)** môžeme jednotlivé režimy prirovnať k získaniu aktuálnych novín. Režim **now** (dotazovací mód) si môžeme predstaviť tak, že pravidelne kontrolujeme stánok a zisťujeme, či vyšlo nové číslo. Naproti tomu režim **socket** pripomína predplatné, ktoré príde vždy, keď vyjde nové číslo.

Výhoda socketového režimu spočíva samozrejme v tom, že dáta sú prijímané, len keď príde k zmene. To znamená, že nedochádza k opakovanému a zbytočnému zisťovaniu stavu. Nevýhodou je náročnejšia implementácia a vyššia réžia na strane servera.

<sup>3</sup>unikátna identifikácia bezdrôtovej senzorovej siete

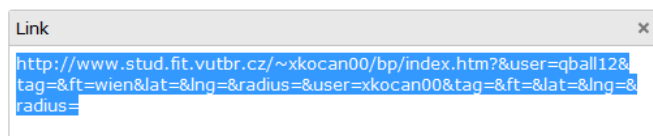
### 4.3.3 Režim history

Režim history ponúka užívateľovi možnosť zobrazovať historické dáta vo forme grafu, ktorý umožňuje vzájomné porovnanie dát na vybraných uzloch. Po nastavení tohto módu sa všetky aktuálne hodnoty na mape prepnú do stavu N/A a užívateľovi sa umožní vybrať časový rozsah pre graf pomocou kalendára.

Graf je komplexná komponenta, ktorá poskytuje posuvnú a približiteľnú časovú os, interpoláciu dát a zobrazenie historických dát interaktívne priamo na mapovom podklade. Všetkými možnosťami grafu sa bližšie zaoberá kapitola 4.5.

### 4.3.4 Ďalšie nástroje

Okrem módov má užívateľ na hornom paneli možnosť vygenerovať hypertextový odkaz na aktuálne vyhľadane a zobrazené uzly, viď obr. 4.6.



Obrázek 4.6: možnosť vytvorenia odkazu na aktuálne vyhľadane uzly

Tvorba tohto odkazu pripomína tvorenie odkazu na google mapách, na ktorý sú užívatelia zvyknutí. Teda po stlačení tlačidla sa vytvorí nové okno, v ktorom je už vyznačený daný odkaz. Odkaz je v tvare

domenové meno?user=p1&tag=p2&ft=p3&lat=p4&lng=p5&radius=p6, kde parametre p1, p2, p3, p4, p5 a p6 tvoria jedno vyhľadanie. Počet takýchto vyhľadání nie je obmedzený. Napríklad po zadaní adresy

index.htm?user=qball12&tag=&ft=wien&lat=&lng=&radius=&user=xkocan00&tag=&ft=&lat=&lng=&radius= sa vyhľadajú všetky uzly od užívateľa qball12 vo Viedni a všetky uzly od užívateľa xkocan00.

Pomocou zaškrývacieho políčka **lock markers** môže uzamknúť značky na mape, aby nebola možná zmena ich polohy. V štandardnom stave je táto možnosť vypnutá a je teda možné okamžite meniť polohu uzlov.

Ďalšie zaškrývacie políčko **Disable tooltips** vypne zobrazovanie okien v mapách s hodnotami všetkých meraných veličín. V štandardnom stave je táto možnosť vypnutá.

Tlačidlo **Save Position(s)** slúži na uloženie polohy všetkých značiek, ktorým bola zmenená poloha a naopak pomocou tlačidla **Reset Position(s)** je možné prinavrátiť všetky značky na pôvodnú pozíciu.

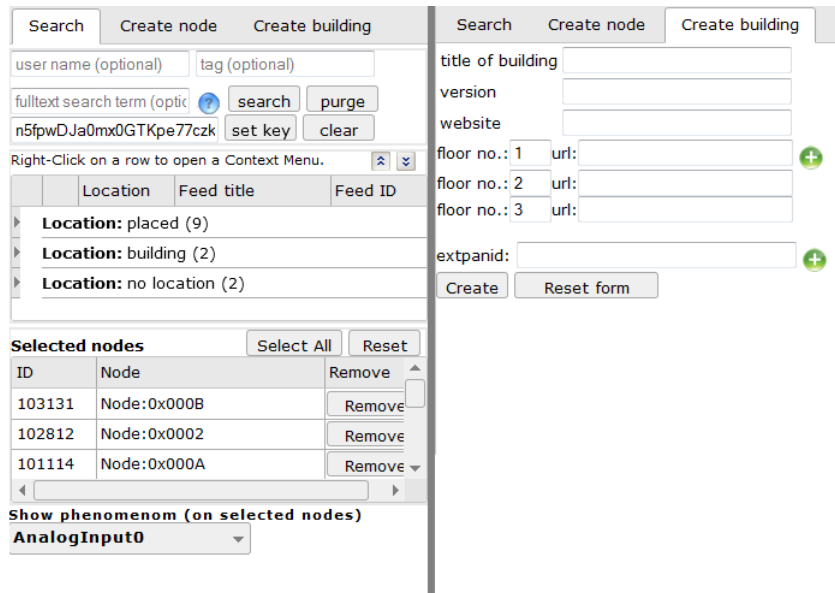
**Position** udáva súradnice miesta, na ktorom má aktuálne užívateľ kurzor. Pozícia je určená zemepisnou šírkou (latitude) a zemepisnou dĺžkou (longitude).

Na hornom paneli sa nachádza aj skrytá oblasť, ktorá zobrazuje pomocné správy pre užívateľa (napríklad ako umiestniť uzol) a zobrazí sa len v prípade, keď sú tieto pomocné správy relevantné pre užívateľa.

Na pravej strane je možnosť zvoliť, či zobrazovať aj vnútorné uzly – **Hide indoor sensors**. Hoci sa primárne zobrazujú len vonkajšie uzly, v niektorých prípadoch (podrobnejšie rozobrané v 5.2.1) sa užívateľovi zobrazia aj vnútorné. Môže sa však stať, že mu tieto značky budú prekážať, preto je tu možnosť vypnúť ich zobrazovanie.

## 4.4 Pravý panel

Pravý panel predstavuje základ pre prácu s aplikáciou. Nachádzajú sa tu záložky pre vyhľadávanie a prácu s existujúcimi uzlami, možnosť tvorby nových uzlov a skrytá záložka s informáciami o použítom kľúči.



Obrázek 4.7: Prehľad možnosti pravého panelu, na ľavej strane možnosti vyhľadávania spolu s výberom uzlov, na pravej vytvorenie budovy

### 4.4.1 Vyhľadávanie

Prvá záložka ponúka možnosti na vyhľadávanie uzlov, ich označovanie, výber dátových tokov a taktiež zobrazenie grafu.

Pokiaľ nebola aplikácia načítaná z vygenerovaného odkazu, tak sa tu nachádzajú len vyhľadávacie polia, do ktorých užívateľ zadáva reťazce, podľa ktorých chce vyhľadávať. Pokiaľ užívateľ ešte nič nenapísal, tak je v nich menej výraznou farbou uvedená pomôcka, na čo pole slúži a v zátvorke je uvedená povinnosť, resp. nepovinnosť vyplniť pole.

**user name - užívateľské meno (nepovinné)** - pomocou tohto poľa môže užívateľ vyhľadať všetky uzly daného užívateľa. Pole je nepovinné. Napríklad: xkocan00.

**tag - značky (nepovinné)** - pomocou tohto poľa sa vyhľadajú všetky uzly, ktoré obsahujú danú značku. Tieto značky zadáva užívateľ pri vytváraní uzlu a slúžia na uľahčenie vyhľadania uzlov. Pole je nepovinné. Napríklad: temperature, humidity, ...

**fulltext (nepovinné)** - vyhľadá všetky uzly, ktoré obsahujú vo svojom uzle tento reťazec. Môže to byť názov uzlu, umiestnenie uzlu, webová stránka uzlu, popisné tagy a pod. Pole je nepovinné. Napríklad: london, office, brno.

Z textu vyššie teda vyplýva, že všetky polia sú nepovinné, avšak na vyhľadanie musí byť aspoň jedno vyplnené. Pri vyplnení viacerých sa vyhľadávajú len tie uzly, ktoré spĺňajú

všetky podmienky súčasne. Vyhľadanie sa dá ovplyvniť aj pomocou tlačidla **Set geo area**, ktoré vymedzuje geografickú oblasť, v ktorej budú uzly vyhľadávané. Po jeho stlačení je užívateľ vyzvaný na určenie stredu kruhu. Následne zväčšovaním, resp. zmenšovaním tohto kruhu určuje veľkosť oblasti, v ktorej sa budú hľadať uzly.

Ďalší textový vstup slúži na zadanie kľúča užívateľa, podľa ktorého sa určujú všetky oprávnenia užívateľa a umožňuje samotné vyhľadávanie. Vopred nastavený kľúč umožňuje vyhľadávanie všetkých verejných uzlov a čítanie dát (teda neumožňuje vykonávať žiadne zmeny). Samozrejme užívateľ môže zadať vlastný kľúč. Po potvrdení pomocou tlačidla **set key** sa validuje platnosť kľúča a vytvorí sa ďalšia záložka v tomto paneli, ktorá obsahuje informácie o kľúči, viď 4.4.4.

Na samotné vyhľadanie slúži tlačidlo **search**. Po vyhľadaní sa užívateľovi sprístupnia ďalšie komponenty panela, a to:

1. Stromová komponenta - Zobrazujú sa všetky uzly, ktoré spĺňajú podmienky vyhľadania zaradené do rôznych skupín:

**s polohou - placed** – do tejto skupiny sa zaradia všetky uzly, ktoré majú zadanú polohu

**bez polohy - no location** – uzly, ktoré nemajú známu polohu

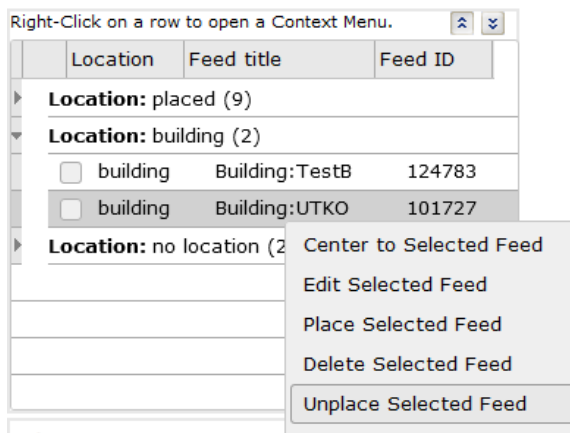
**budova - building** – uzly, ktoré sú označené ako budova

Po rozvinutí danej skupiny sa zobrazia už konkrétne výsledky. Každý uzol je identifikovaný svojím jedinečným číslom – **feed ID** a svojím názvom – **feed title**. Pri uzloch patriacich do skupiny s polohou alebo bez polohy má užívateľ možnosť ho označiť pomocou zaškrtávacieho tlačidla, čím sa označí na mape a zaraďí sa do zoznamu označených uzlov. Po stlačení pravého tlačidla nad výsledkom vyhľadávania sa užívateľovi sprístupní kontextová ponuka, v ktorej má možnosť:

- centrovať mapu na vybraný uzol (**center to selected feed**) – vycentruje mapu tak, že vybraný uzol bude v jej strede
- editovať uzol (**edit selected feed**) – užívateľ môže zmeniť názov uzlu, webovú stránku alebo verziu. Sú potrebné práva na zmenu uzlov.
- umiestniť uzol (**place selected feed**) – užívateľ môže nastaviť geografickú polohu uzlu priamo jeho umiestnením do mapy. Aj na túto akciu sú potrebné práva na zmenu uzlov.
- vymazať uzol (**delete selected feed**) – umožní vymazať daný uzol aj so všetkými historickými údajmi zo sensorov. Nevratná operácia. Sú potrebné práva na mazanie uzlov.
- zrušiť umiestnenie uzlu (**unplace selected feed**) – užívateľ má možnosť zrušiť nastavenú polohu uzlu. V tomto prípade bude po ďalšom vyhľadaní uzol zaradený medzi uzly bez polohy. Sú potrebné práva na zmenu uzlov.

Výsledky vyhľadania je možné triediť podľa filtra a taktiež je možné ich abecedne zoradiť. Filter ponúka možnosti filtrovať riadky podľa zadaného jedného alebo dvoch hesiel. Operácia medzi týmito dvoma heslami môže byť *konjunkcia*, resp. *disjunkcia*, tzn. sa zobrazia len riadky, ktoré obsahujú zároveň heslo 1 a heslo 2, resp. heslo 1 alebo heslo 2. Ponuka na tieto akcie sa vyvolá stlačením tlačidla v hlavičke stromovej komponenty.

Na obrázku 4.8 je možné vidieť stromovú komponentu s rozvinutými všetkými budovami a kontextovú ponuku na prácu s vybraným uzlom.

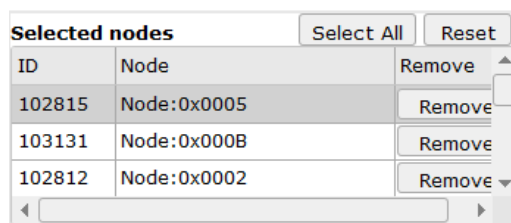


Obrázek 4.8: Stromová komponenta

2. Komponenta s označenými uzlami – V tejto časti sa užívateľovi zobrazujú všetky označené uzly. Sú identifikované jedinečným číslom – **feed ID** a svojim názvom – **feed title**. Okrem toho je možnosť vymazať daný uzol spomedzi vybraných uzlov pomocou tlačidla **remove**.

Po označení uzlu sa užívateľovi sprístupnia jeho dátové toky v rolovacej ponuke (spolu so všetkými ostatnými dátovými tokmi označených uzlov, teda je definovaná operácia zjednotenia na množinách dátových tokov označených uzlov).

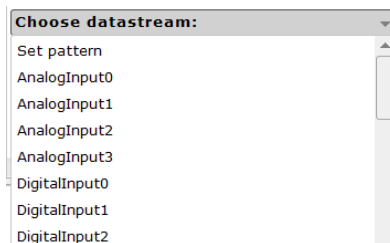
Pre jednoduchší výber a prácu s uzlami má užívateľ možnosť označiť všetky uzly naraz pomocou tlačidla **Select All**, respektíve zrušiť označenie všetkých uzlov pomocou tlačidla **Reset**



Obrázek 4.9: Komponenta s označenými uzlami

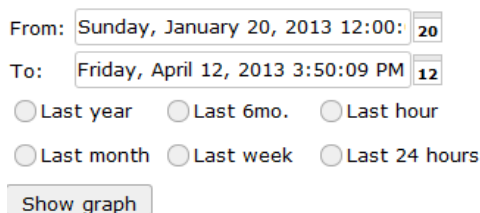
3. Rolovací zoznam – Ako bolo vyššie spomenuté, v tomto zozname sa nachádzajú všetky dostupné dátové toky označených uzlov. Následne po vybratí jednej položky z tohto zoznamu, sa pri každej značke na mape zobrazí aktuálna hodnota vybraného dátového toku. Ak niektorý z uzlov na mape nemá daný dátový tok, tak sa na danej značke zobrazí hodnota **N/A**. Podľa hodnoty vybratej v tejto ponuke sa vyberajú aj dáta, ktoré sa zobrazia v grafe.

Okrem konkrétnych dátových tokov môže užívateľ zvoliť aj možnosť **Set pattern**. Po zadaní reťazca sa pri každom uzle vyberie posledný dátový tok, ktorý zodpovedá zadanému vzoru. Táto možnosť bola pridaná hlavne z toho dôvodu, že názvy dátových tokov si určujú sami používatelia a tým pádom môžu byť niektoré senzory zbytočne filtrované, hoci zaznamenávajú rovnakú veličinu. Príkladom môže byť zadanie vzoru *teplota*, kde po jeho zadaní sa na všetkých označených uzloch zobrazí dátový tok, ktorého názov obsahuje tento reťazec (teda napr. *Teplota1*, *teplotavdome*,...)



Obrázek 4.10: Rolovací zoznam

4. Kalendár – Kalendár je komponenta, ktorá je v stavoch `now` a `now(socket)` pre užívateľa skrytá. Pokiaľ však vyberie možnosť `history`, zobrazí sa mu ponuka, v ktorej môže vybrať časový rozsah, v ktorom ho zaujímajú historické dáta z vybraných uzlov. Môže si nastaviť buď konkrétny čas alebo vybrať jednu z ponúknutých možností pre dáta za poslednú hodinu, deň, týždeň, mesiac, pol rok alebo rok.



Obrázek 4.11: Kalendár

#### 4.4.2 Vytvorenie uzlu

Po kliknutí na záložku **Create node** v pravom paneli má užívateľ možnosť si vytvoriť nový uzol. Musí vyplniť všetky požadované položky, teda názov uzlu, webovú adresu, verziu. Voliteľné položky sú značky (tagy) a dátové toky. ID bude uzlu priradené automaticky.

Po stlačení tlačidla `create` je uzol vytvorený a je okamžite vyhľadateľný. Samozrejme na vytvorenie uzlu je potrebné zadať kľúč, ktorý ma dostatočné práva.

#### 4.4.3 Vytvorenie budovy

Táto záložka ponúka podobnú možnosť ako predošlá, je však prispôbená pre využitie v rámci aplikácií `OutdoorView-IndoorView`, keďže ponúka priamo možnosť vytvoriť uzol, ktorý spĺňa špecifikácie budovy. To znamená upravuje jej názov (budovy majú štandardizovaný názov `Building:názov`). Ďalej je tu užívateľovi sprístupnená možnosť vytvárať

poschodia v danej budove a pre každé poschodie vložiť hypertextový odkaz pre dané poschodie do IndoorView aplikácie. Možnosť vytvoriť ďalšie poschodie je umožnená po stlačení ikony vedľa vstupných polí. Ďalšou (voliteľnou) položkou je vložiť jeden alebo viacero popisov extpanid<sup>4</sup>. Tieto popisy poschodí a extpanid sú automaticky vkladané medzi značky (tagy) vytvorenej budovy.

#### 4.4.4 Informácie o kľúči

Aby mohol užívateľ využívať funkcie OutdoorView, musí zadať platný kľúč s patričnými povoleniami. Možnosť zadať vlastný kľúč má v prvej záložke. Po jeho potvrdení tlačidlom **set key** sa vytvorí nová záložka – **info**, v ktorej sú všetky informácie o zadanom kľúči. Teda v prvom rade, či je to platný kľúč, jeho názov, počet ním vytvorených dátových tokov, maximálny počet dátových tokov a všetky operácie, ktoré sú s ním povolené:

- get** – možnosť vyhľadávať uzly, len čítanie
- put** – možnosť editovať uzly
- post** – možnosť vytvárať uzly
- delete** – možnosť mazať uzly

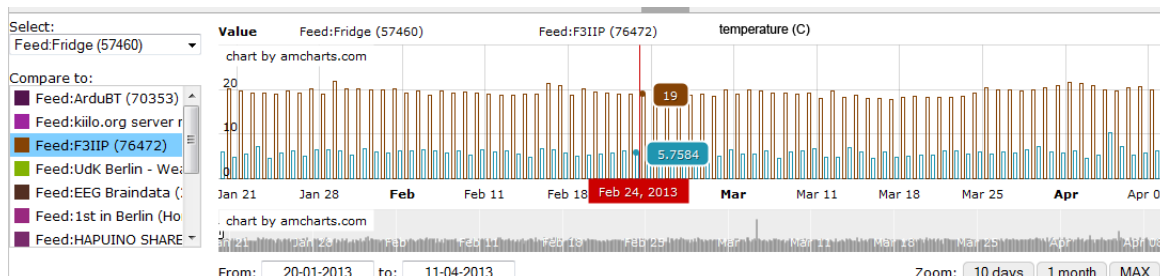
Search	Create node	Create building	Info
<b>The key has been set</b>			
Information:			
api_key	n5fpwDJa0mx0GTKpe77czke9mi6SAKx		
label	keyToAll		
private_access	true		
global_triggers	true		
datastreams_count	6		
datastreams_permitted	99999999		
history	99999999		
creatable_roles	signup_plan		
permissions	get,put,post,delete		

Obrázek 4.12: Informácie o kľúči

<sup>4</sup>unikátne id bezdrôtovej senzorovej siete

## 4.5 Graf

Jednou z hlavných požiadaviek na aplikáciu bola možnosť zobrazovať historické dáta. Na najprehľadnejšie zobrazenie takéhoto množstva dát slúži graf. Jeho typ a použitie bude vysvetlené v tejto kapitole.



Obrázek 4.13: Načítané dva grafy z dát uzlov Fridge a F3IIP za posledný rok

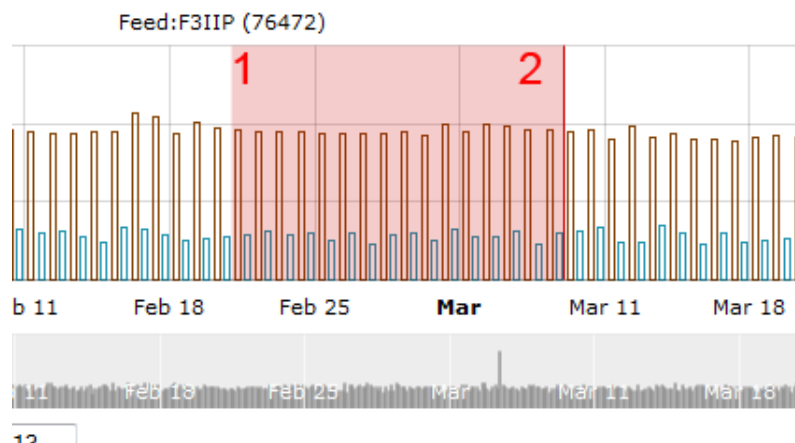
### 4.5.1 Použitie

V aplikácii je možnosť zobrazíť graf možná len vtedy, ak je nastavený režim history. Po prepnutí do tohto režimu má užívateľ možnosť zvoliť časový rozsah z kalendára, ktorý chce zobrazíť na grafe a takisto sa mu sprístupní tlačidlo na vytvorenie grafu – **show graph**. Po jeho stlačení sa minimalizuje mapa a vytvorí sa nový panel, v ktorom je samotný graf a ostatné nástroje na prácu s ním.

Na export dát z grafu slúži tlačidlo **Export**, pomocou je umožnené užívateľovi uložiť všetky dáta z grafov na svoj lokálny počítač. Dáta sú vo formáte CSV<sup>5</sup> a obsahujú názov uzlu, čas a hodnotu v danom čase. Užívateľ má širokú ponuku nástrojov na prácu s grafom. Pokiaľ je vybratých viacero uzlov, ktoré majú rovnaký názov dátového toku a tento je označený v rolovačom zozname, tak sa mu po ľavej strane sprístupní ponuka, kde môže vybrať uzol, z ktorého sa vytvorí primárny graf. Následne pod ním môže vybrať ďalšie uzly, ktoré sa vykresľujú do toho istého panela a užívateľ má možnosť okamžite porovnávať hodnoty. Po prejdení kurzorom nad grafom sa hodnoty každého vykresleného grafu zobrazujú v bublinách.

V pravej časti sa nachádza ponuka na konkrétnejší výber časového obdobia, teda buď maximálny možný rozsah alebo jeho dielčie časti (mesiac, týždeň, ...). Užívateľ môže samozrejme zvoliť aj jemnejší rozsah, napríklad v prípade sledovania výkyvov v dátach. V tomto prípade stačí myšou zvýrazniť požadovaný rozsah, použitie je možné vidieť na obr. 4.14. Graf je prepojený aj s mapovou časťou aplikácie, konkrétne ak užívateľ klikne do ľubovoľnej oblasti stĺpca grafu, tak sa všetky značky (markery) aktualizujú na hodnotu, akú mali v danom čase. Pokiaľ nie je graf priblížený do takej úrovne, aby sa zobrazovali jednotlivé namerané hodnoty, tak sa tieto hodnoty zoskupujú do väčších celkov (podľa časového rozsahu do týždňov, mesiacov). Hodnota takéhoto zoskupeného stĺpca je určená ako medián všetkých hodnôt, ktoré ho tvoria. Preto ak užívateľ klikne do takéhoto stĺpca, tak sa nezobrazí hodnota daného mediánu, ale konkrétna hodnota v danom čase.

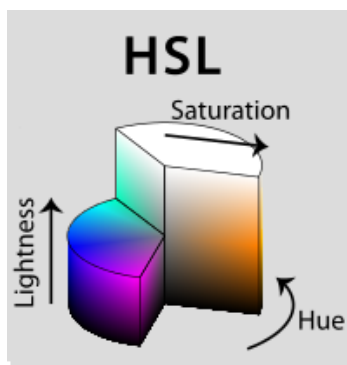
<sup>5</sup>CSV (Comma separated values format, hodnoty oddelené čiarkami) – jednoduchý súborový formát určený na výmenu a konvertovanie dát tabuľkovými aplikáciami [8].



Obrázek 4.14: Presnejšie zobrazenie údajov bolo zvolené ťahaním kurzora z bodu 1 do bodu 2

#### 4.5.2 Typ grafu

Ako štandardný typ grafu je zvolený stĺpcový, keďže pri čiarovom dochádza k skresľovaniu vplyvom interpolácie, ak neboli dáta zo senzorov v pravidelných intervaloch alebo nejaké chýbali. Používateľ má však možnosť zmeniť toto nastavenie a dáta zobraziť pomocou čiarového grafu alebo interpolovanej krivky. Slúžia na to tlačidlá v pravom paneli s ikonami znázorňujúcimi jednotlivé typy grafov. Graf je tvorený z dát všetkých označených uzlov, na x-ovej osi sa zobrazuje čas (v závislosti od rozsahu to môžu byť mesiace, dni, hodiny, minúty) a na y-ovej ose sa zobrazujú hodnoty dátového toku. Ako bolo spomenuté, v paneli sa môže zobrazovať aj viacero grafov naraz. Z tohto dôvodu je vhodné ich farebne odlišiť. Toto farebné odlíšenie bolo navrhnuté tak, aby sa pre každý graf generovala náhodná farba,



Obrázek 4.15: Farebný model HSL. Zdroj [10]

avšak bolo nutné zabezpečiť, aby vygenerované farby neboli príliš svetlé, málo satureované a pod. Z tohto dôvodu bol využitý model HSL. HSL je užívateľsky orientovaný farebný model. Nastavuje:

- sýtosť (saturation) – náhodne generovaná v rozmedzí 60-100%
- farebný tón (hue) – generovaný v plnom rozsahu, teda 1-360

- jas (lightness) – bol zvolený rozsah od 30 do 70% (aby sa vyhlo tmavým, respektíve bledým farbám)

Nevýhodou tohto modelu je netriviálny prevod do modelu RGB, ktorý sa využíva na vykresľovanie. Na tento prevod bol využitý algoritmus [10].

### 4.5.3 Upozornenia pre užívateľa

Užívateľ je na vykonané akcie upozornený dvoma spôsobmi.

V prvom prípade sa zobrazí modálne okno v okamihu, keď chce vykonať nepovolenú operáciu. Modálne okno je typ dialógového okna, s ktorým musí užívateľ interagovať, aby mohol pokračovať v práci s aplikáciou [6]. Je využité v prípade, ak sa snaží premiestniť uzly, na ktoré nemá práva, ak zadal neplatný kľúč, ak sa snaží vymazať uzol bez dostatočných práv, zrušiť umiestnenie uzlu, ktorý ešte nemá polohu atď.

V druhom prípade je využité skryté okno, ktoré len v prípade potreby zobrazuje oznámenia pre užívateľa. Napríklad ak sa podarí uložiť polohu uzlov alebo oznámenie o tom, že môže umiestniť uzol.

## Kapitola 5

# Implementácia

Pri vývoji aplikácie bol kladený veľký dôraz na optimalizáciu. Aplikácia totiž pracuje s veľkým množstvom dát a musí vykazovať primeranú odozvu. Výber takýchto optimalizácií je v podkapitole 5.5. V ďalších podkapitolách bude popísaná komunikácia so serverom, implementácia na klientskej strane a použité technológie.

Architektúra aplikácie je postavená na návrhovom vzore MVC (model, view, controller).

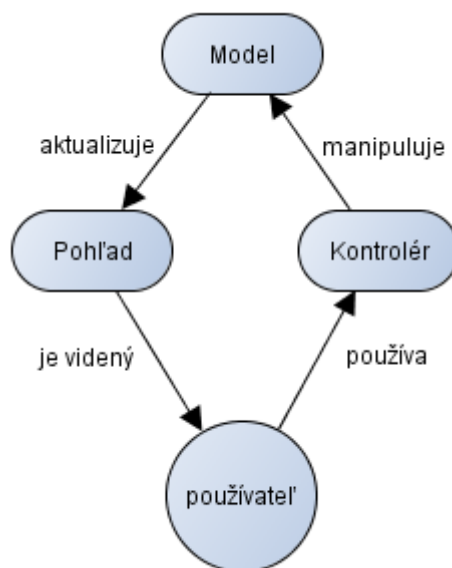
### 5.1 Model, view, controller

Je typ softwarovej architektúry, kde je od seba oddelený dátový model, užívateľské rozhranie a riadiaca logika do troch nezávislých komponent tak, že modifikácia niektorej z nich má len minimálny vplyv na ostatné [7].

- kontrolér (controller) – reaguje na udalosti od užívateľa (stlačenie tlačidiel a podobne). Hlavnou úlohou je získať potrebné zdroje na danú operáciu z modelu a zaistiť príslušné zmeny v pohľade.
- model (model) – predstavuje samotné dáta a pravidlá, ktoré sa na ne vzťahujú. Kontroléru poskytuje model údaje, ktoré si užívateľ vyžiada. Pokiaľ tieto údaje len zobrazujeme, tak je model nemenný a menia sa len spôsoby ako dáta zobrazujeme. Pokiaľ má užívateľ možnosť editovať dáta, mení sa aj model. V našom prípade sú modelom dáta požadované z externého servera, jedná sa teda o uzly spolu s ich bližšou špecifikáciou, ktorú reprezentujeme podľa akcií užívateľa.
- pohľad (view) – vytvára viacero možností ako prezentovať dáta z modelu. Je na kontroléri, aby vybral správny pohľad. V našom prípade to môže byť zobrazenie správnej záložky po stlačení tlačidla alebo zobrazovanie korektných údajov pri značkách podľa toho, aké si užívateľ vyberie.

Výhodou MVC je, že podporuje užívateľov, ktorí môžu mať rôzne typy zariadení (resp. rôzne typy prehliadačov, rozlíšení), keďže kontrolér vyberá vhodné typy pohľadov na zobrazenie na týchto zariadeniach.

Ďalšou výhodou je, že MVC napomáha zjednodušiť komplexnosť veľkých aplikácií (akou OutdoorView určite je). Samotný kód je oveľa viac štruktúrovaný a teda aj jednoduchší na udržiavanie, testovanie a znovu použitie.



Obrázek 5.1: MVC v praxi

## 5.2 Použité technológie na strane servera (Cosm/Pachube)

Dáta zo senzorov je samozrejme nutné čítať a ukladať. Ako takéto úložisko bol zvolený portál Cosm (predtým pachube), keďže funguje ako celosvetové online úložisko pre dáta z rôznych druhov senzorov a zariadení. Pre aplikáciu poskytuje vhodné aplikačné rozhranie a bol doteraz využívaný aj v rámci aplikácie IndoorView. Navyše vďaka tomu, že je využívaný množstvom ľudí a organizácií môže mať aplikácia OutdoorView do budúcnosti oveľa širšie využitie.

### 5.2.1 Komunikácia so službou Cosm

Aplikácia komunikuje so servermi pomocou technológie AJAX, kde je pre každú žiadosť vytvorená HTTP požiadavka. Na túto žiadosť server odpovedá vrátením dát vo formáte JSON. Všetky tieto požiadavky sú asynchrónne, teda aplikácia nemusí čakať na túto odpoveď, ale spracuje ju až v dobe, keď je prijatá. To umožňuje pracovať s webovou stránkou bez nutnosti jej znovu načítania. Súčasťou každej HTTP hlavičky odoslanej na Cosm musí byť aj cosm kľúč, ktorý zadal užívateľ do príslušného textového poľa.

Metódy na komunikáciu so servermi Cosm, ktoré bolo treba implementovať:

1. vyhľadanie všetkých uzlov podľa zadaných podmienok – odpoveďou na túto žiadosť boli všetky uzly, ktoré spĺňali podmienky (zadané užívateľské meno, full textové vyhľadanie alebo tag) spolu s aktuálnymi hodnotami dátových tokov na každom z týchto uzlov. Tento typ žiadosti bol využitý pri prvotnom vyhľadaní a taktiež periodicky každých  $n$  sekúnd v režime now.
2. prihlásenie sa na odber k uzlu – tento typ žiadosti bol využitý v režime now (socket), kde po vyhľadaní sa každý uzol prihlásil na odber a prijal nové dáta vždy, keď prišlo

k zmene na senzore. S tým súvisela nutnosť implementovať metódu na odhlásenie sa od odberu, ak bol zmenený režim na now alebo history.

3. história konkrétneho uzlu – táto žiadosť sa odosiela vždy pri tvorení nového grafu. Parametrami bol časový rozsah (určený z kalendára) a interval medzi hodnotami z uzlu. Keďže je možné získať naraz len 1000 hodnôt z konkrétneho uzlu bolo nutné tento interval určiť podľa tabuľky 5.1.

interval	popis	max. rozsah získaných dát
0	každá zaznamenaná hodnota	6 hodín
30	hodnota v každých 30 sekundách	12 hodín
60	hodnota v každej minúte	24 hodín
300	hodnota v každých 5 minútach	5 dní
900	hodnota v každých 15 minútach	14 dní
3600	hodnota v každej hodine	31 dní
10800	hodnota za každé tri hodiny	90 dní
86400	hodnota za deň	1 rok

Tabuľka 5.1: Zvolenie intervalu medzi získavanými hodnotami na základe užívateľom zvoleného časového rozsahu

4. vytvorenie nového uzlu – v prípade vytvorenia nového uzlu alebo budovy
5. aktualizácia konkrétneho uzlu – vykonala sa vždy, ak užívateľ editoval uzol, menil jeho polohu
6. vymazanie konkrétneho uzlu
7. zistenie práv na kľúč – viď 4.4.4
8. zistenie práv na konkrétny úkon – napríklad povolenie na presun uzlu. Nebolo možné použiť informácie o kľúči, pretože kľúč môže mať povolenie upravovať len isté uzly (nemusi mať toto povolenie pre všetky vyhľadane)

Všetky dáta sú prenášané vo formáte JSON<sup>1</sup>, čo je to štandard pre výmenu dát s nízkou redundanciou. Tieto dáta reprezentujú dátové štruktúry alebo asociatívne polia. Výhodou je, že JSON bol priamo navrhnutý pre jazyk JavaScript, na ktorom je postavená celá aplikácia, čiže bolo jednoduché spracovať takéto dáta. Formát prijatých dát je znázornený na obr. 5.2. Súčasťou tohoto asociatívneho poľa je ďalšie asociatívne pole location (obr. 5.3) Na základe týchto údajov dochádzalo k primárnemu deleniu uzlov podľa tabuľky 5.2. Všetky ostatné kombinácie boli filtrované a zahadzované, keďže sa jednalo o interiérové uzly, pre ktoré táto aplikácia nie je určená.

<sup>1</sup>JavaScript Object Notation (JavaScriptový objektový zápis)

```

totalResults      5
itemsPerPage     50
startIndex       0
[-] results
  [-] 0
    id            96862
    title        "My room"
    private      "false"
    [+] tags     [ "arduino", "enviroment", "humidity",      ]
    [+] description "Temperature and Humidit...ino and Sparkfun SHT15."
    feed         "https://api.cosm.com/v2/feeds/96862.json"
    status       "live"
    updated      "2013-04-06T10:47:52.296009Z"
    created      "2013-01-09T21:04:22.191042Z"
    creator      "https://cosm.com/users/qball12"
    version      "1.0.0"
    [+] datastreams [ Object { id="Humidity", current_value="28.78", at="2013-04-06T10:47:52.209886Z",
    [+] location  Object { disposition="fixed", name="Wien, Vienna", exposure="indoor",      }
  [+] 1          Object { id=97307, title="Automatisierung Wien", private="false",      }
  [+] 2          Object { id=43263, title="meinersterfeed", private="false",      }
  [+] 3          Object { id=27177, title="Testfeed", private="false",      }
  [+] 4          Object { id=17623, title="weather station HTL Wien 10", private="false",      }

```

Obrázek 5.2: formát prijatých dát

```

[-] location
  disposition    "fixed"
  name           "Berlin Kreuzberg"
  exposure       "outdoor"
  domain         "physical"
  ele            "55m"
  lat            52.4949282559702
  lon            13.4342622756958

```

Obrázek 5.3: asociatívne pole location

## 5.3 Použitie technológie na klientskej strane

OutdoorView je webová aplikácia zameraná predovšetkým na klientskú stranu, využíva teda všetky technológie na to určené, konkrétne HTML5, CSS3, javascript. Pri implementácii a návrhu dizajnu boli ako priority stanovené jednoduchosť a intuitívnosť ovládania.

### 5.3.1 HTML5,CSS

HTML5 je nová špecifikácia jazyka HTML a je nástupcom jazyka HTML 4.0 s mnohými vylepšeniami. Keďže však ešte stále nepodporujú všetky prehliadače plne HTML5 [3], bola aplikácia napísaná tak, aby splňovala štandardy HTML5. Avšak žiadne kľúčové funkcie nie sú napísané v HTML5, aby nebola v súčasnosti pre niektorých používateľov obmedzená funkčnosť.

Samotný jazyk html je využitý pri tvorbe kostry stránky, ktorá je tvorená viacerými blo-

location	exposure	latitude & longitude	feed title	typ uzlu
undefined	-	-	-	uzol bez polohy
defined	undefined	-	-	uzol bez polohy
defined	outdoor	undefined	-	uzol bez polohy
defined	defined	zadaná poloha	Building:x	budova
defined	outdoor	zadaná poloha	-	vonkajší uzol
defined	indoor	zadaná poloha	-	vonkajší uzol

Tabulka 5.2: Spôsob filtrovania uzlov podľa polohy

kovými elementami (div), ktoré reflektujú kapitolu 4. Tzn. existuje blok pre horný panel, mapu, pravý panel a graf. Takisto existuje takýto blokový element pre každý prvok užívateľského rozhrania (záložky, tabuľky, rolovacie zoznamy, ...). Samotné blokové elementy však neposkytujú podporu pre takéto prvky, preto sú rozšírené pomocou jazyka javascript (doplňa ich funkcionality, priraduje im štýly a pod.).

Dominantná časť dizajnu je vytvorená pomocou kaskádových štýlov CSS verzie 2, hoci už existuje CSS3. Rozhodol som sa tak preto, že ešte nie je úplne zjednotený štandard CSS3 a prehliadače ho podporujú rôzne. V aplikácií existujú aj isté prvky, ktoré využívajú CSS3, avšak v prípade, že nie sú podporované prehliadačom, použije sa ich alternatíva. Jedná sa napríklad o animáciu zväčšujúcu informačné okno pre používateľa.

### 5.3.2 Javascript

OutdoorView je aplikácia, kde zdrojom dát je externý server Cosm a vizualizácia dát je vykonávaná výhradne na klientskej strane. V súčasnej dobe, jediný jazyk, ktorý podporuje všetky požadované funkcie na spracovanie týchto dát a ich zobrazenie je javascript, preto je väčšina projektu implementovaná práve v ňom. Podobne ako HTML5 alebo CSS3 je aj javascript v rôznych prehliadačoch interpretovaný rôzne. Z tohto dôvodu bolo hlavne pri tvorbe požiadaviek na server (ajax) a prístupe k objektovému modelu stránky (DOM) využitá knižnica jQuery<sup>2</sup>, ktorá zapúzdruje všetky tieto funkcie pre rôzne typy prehliadačov.

Javascript je využitý na tvorbu užívateľského rozhrania pomocou frameworku jqwidgets<sup>3</sup>. Tento framework je založený na knižnici jQuery, tým pádom poskytuje podporu pre všetky typy moderných prehliadačov, nevynímajúc prehliadače na mobilných zariadeniach. Ďalším dôvodom, prečo som sa rozhodol práve pre tento framework je veľmi široké aplikačné rozhranie, ponúka možnosti vytvoriť všetky potrebné prvky GUI, ako sú tlačidlá, tabuľky, zoznamy a s tým spojené súvisiace metódy na aktualizáciu tabuliek, rolovacích zoznamov apod. Nespornou výhodou je aj jeho bezplatné použitie v prípade nekomerčných projektov.

Grafy boli vytvorené pomocou knižnice amCharts<sup>4</sup>, ktorá je taktiež v prípade nekomerčného použitia bezplatná. Táto knižnica bola vybraná z toho dôvodu, že poskytuje všetky možnosti, ktoré boli požadované pre prácu s grafmi (popísane v sekcii 4.5 a takisto je celá implementovaná v jazyku javascript, takže nebol problém ju integrovať do aplikácie.

<sup>2</sup><http://api.jquery.com/>

<sup>3</sup><http://www.jqwidgets.com/>

<sup>4</sup><http://www.amcharts.com/stock-chart/>

## 5.4 Nástroje a aplikácie využívané pri implementácií

Na vývoj aplikácie bol využitý program Microsoft Visual Studio 2010 s doplnkami na vývoj webových aplikácií (pomôcky pre organizáciu JavaScriptového kódu, html a css), na tvorbu niektorých prvkov užívateľského rozhrania a ikon som využíval program Adobe Photoshop. Testovanie prebiehalo hlavne pomocou prehliadačov Firefox a Chrome s doplnkom Firebug. Funkčnosť bola otestovaná aj na prehliadačoch Opera a Safari.

## 5.5 Optimalizácie

Ako už bolo spomenuté jedná sa o klientskú aplikáciu, ktorá pracuje s veľkými množstvami dát a taktiež je požadovaná aj rýchla doba odozvy. Z tohto dôvodu museli byť preto všetky náročné operácie optimalizované:

### 5.5.1 Funkcia `setTimeout`

JavaScript štandardne nepodporuje paralelné procesy, tým pádom celá aplikácia využíva jedno vlákno [2], čo je vo väčšine prípadov postačujúce. Pokiaľ však príde k operácii, ktorá je časovo a výpočetne náročná, môže prísť k zahlteniu tohto vlákna. V tomto prípade prehliadač prestane odpovedať a užívateľ má možnosť prerušiť výpočet alebo čakať ďalej. Takouto operáciou v aplikácii je vyhľadanie veľkého množstva uzlov, kde pri každom sa musí vykonať rozhodovanie o aký typ uzlu ide, jeho vykreslenie, spracovanie aktuálnych hodnôt z jeho dátových tokov a podobne.

Aby sa predišlo tomuto nežiaducemu správaniu je využitá natívna javascriptová funkcia `setTimeout`, ktorá je ekvivalentom funkcie `sleep` známej z iných programovacích jazykov. Vždy po určitom množstve uzlov (štandardne je nastavených 50 uzlov) sa aplikácia na krátku dobu uspí, aby mal procesor čas spracovať všetky uzly. Toto sa cyklicky opakuje, až kým nie sú spracované všetky uzly.

### 5.5.2 Kompresia zdrojových kódov

Ako bolo spomenuté, javascript je interpretovaný jazyk, ktorý pracuje na klientskej strane. Z toho dôvodu musia byť všetky zdrojové kódy stiahnuté prehliadačom pri prvotnom zobrazení stránky. Aby bolo toto množstvo dát čo najviac minimalizované, bola použitá minifikácia zdrojových kódov. Odstránili sa všetky biele znaky, znaky nového riadku, komentáre, názvy premenných boli nahradené za kratšie názvy a podobne.

Význam minifikácie nespočíva len v znížení dát potrebných na prenesenie, ale aj na skomplikovanie kódu<sup>5</sup> pre čítanie druhými ľuďmi, aby zabránili (alebo aspoň sťažili) kopírovanie zdrojových kódov.

V tomto projekte bol na tento účel použitý nástroj google closure compiler<sup>6</sup>, ktorý zredukoval veľkosť zdrojových kódov o 40%.

### 5.5.3 Optimalizácia JavaScript kódu

Ďalšie optimalizácie vychádzali najmä z článku [4]. Spoza všetkých spomeniem tie, ktoré najviac prispeli k zrýchleniu aplikácie:

<sup>5</sup>angl. obfuscate - urobenie nejasným

<sup>6</sup><https://developers.google.com/closure/compiler/>

- preferovanie lokálnych premenných všade, kde je to možné
- nepoužívanie príkazu `with`
- využívanie jednoduchých premenných je rýchlejšie ako pristupovať k atribútom objektu alebo k poľu
- obmedziť používanie cyklu `for .. in`
- redukcia operácií s objektovým modelom stránky (DOM)

Z tabuľky 5.3 je vidieť prínos optimalizácií hlavne pri vyhľadani rôzneho počtu uzlov. Prejavili sa najmä pri spracovaní veľkého počtu uzlov, príkladom je spracovanie 287 uzlov, kde sa dosiahla časová úspora vyše 30 sekúnd. Na meranie som použil nástroj console<sup>7</sup>, ktorý je súčasťou doplnku Firebug pre prehliadač Firefox.

počet uzlov	čas odozvy od servera	celková doba spracovania (vrátane odozvy od servera)	celková doba spracovania s optimalizáciami
5	610ms	2900ms	1815ms
36	938ms	7542ms	2245ms
287	2002ms	38 754ms	3800ms

Tabuľka 5.3: Čas spracovania rôzneho počtu uzlov s optimalizáciami a bez

#### 5.5.4 Systémové nároky

Rýchlosť samotnej aplikácie je závislá od

1. počítača - na aplikáciu postačuje súčasný bežný počítač (aplikácia bola testovaná na počítači s procesorom Intel i7 so 4GB systémovej pamäte). Samozrejme je aplikácia funkčná aj na starších počítačoch, ale napr. doba vykreslenia, resp. práca s grafom s veľkým množstvom dát sa môže spomaliť.
2. typu pripojenia na internet - rýchlosť a najmä doba odozvy použitého internetového pripojenia ovplyvňuje rýchlosť získavania dát z externého servera a tým pádom aj rýchlosť vykreslenia a zobrazenia uzlov.
3. prehliadača - hoci bola aplikácia vyvíjaná a testovaná vo všetkých moderných prehliadačoch (Internet Explorer, Firefox, Chrome, Opera, Safari), odporúča sa predovšetkým použitie prehliadača Chrome alebo Firefox, keďže v nich dosahuje rýchlosť vykonávania javascriptového kódu najlepšie časy vďaka interným optimalizáciám týchto prehliadačov.

Samotné vykreslenie a zobrazenie uzlov je taktiež závislé aj na množstve vyhľadaných uzlov.

<sup>7</sup>[http://getfirebug.com/wiki/index.php/Console\\_API](http://getfirebug.com/wiki/index.php/Console_API)

# Kapitola 6

## Testovanie

Testovanie pozostávalo z dvoch hlavných častí. Prvou bolo testovanie správnej funkčnosti aplikácie, kedy som sa zamerlal na overenie validity výsledkov. Druhou časťou bolo testovanie navrhnutého užívateľského rozhrania, či zodpovedá požiadavkám, je dostatočne intuitívne apod. V tejto časti sa zameriam aj na špecifikáciu koncových užívateľov tejto aplikácie a možný ďalší vývoj do budúcnosti.

### 6.1 1. fáza testovania

1. fáza testovania prebiehala už od začiatku práce na projekte až po jeho koniec, pretože už počas implementácie bolo nutné overovať validitu dát a správny vývoj GUI. Keďže toto testovanie prebiehalo ešte počas práce na projekte, tak sa na testovaní podieľal len obmedzený počet ľudí.

#### 6.1.1 Kontrola správnosti dát

Kontrolu správnosti dát som overoval z viacerých dôvodov:

- Zo serverov Cosm sa prenášalo veľké množstvo dát, ktoré bolo nutné rozdeliť do niekoľkých častí. Preto sa mohlo ľahko stať, že niektorá z týchto požiadaviek mohla byť serverom chybné zamietnutá alebo vypršal čas na jej prijatie. V takom prípade by zobrazované informácie nezodpovedali realite (počet nájdených uzlov by bol menší ako reálny počet uzlov) a aplikácia by nefungovala správne bez toho, aby o tom užívateľ vedel. V súčasnosti je na takúto chybu užívateľ upozornený s tým, že má voľbu opakovať neskôr. Takéto správanie nastávalo najčastejšie ak bol server preťažený.
- Bolo nutné správne filtrovať uzly, ktoré sú pre aplikáciu určené. Hlavne z toho dôvodu, že nastavenie lokality uzlu je ponechané na užívateľovi, ktorý nemusí vedieť správne nastaviť lokalitu a tým pádom nie je úplne jednoznačné, ktoré uzly sú interiérové a ktoré exteriérové. Často dochádzalo k nesprávnemu filtrovaniu, teda sa mohli niektoré uzly zobrazovať navyše alebo niektoré chýbali. Konečná implementácia filtrovania je navrhnutá podľa tabuľky 5.2.
- Chybné zobrazenie údajov v čiarovom grafe - vplyvom interpolácie dochádzalo k skresľovaniu údajov. Táto chyba bola zistená testovaním s dátami, ktoré neboli v pravidelných časových intervaloch alebo boli dlhšie časové obdobia bez výstupu zo senzorov.

### 6.1.2 Testovanie UI

Zmeny a testovanie užívateľského rozhrania bolo nutné vykonávať už počas vývoja. Konkrétne v nadväznosti na IndoorView prišlo k preusporiadaniu prvkov – vyhľadávanie bolo presunuté do pravého panela, horný panel bol venovaný len nastaveniam a zmenám režimu. Okrem toho, som sa pri implementácii snažil o to, aby som splnil nasledujúce body (vychádzajúce z Schneidermanových zlatých pravidiel pre design prostredia):

- o konzistenciu v sekvenciách akcií
- ponúknuť informatívnu spätnú väzbu pre každú akciu používateľa
- uistenie, aby užívateľ vedel, že systém vykonal príkaz, tzn. informovať o tom, že sa podarilo vytvoriť uzol, že nemá povolenie presúvať uzly apod.
- predchádzať chybám alebo zabezpečiť ich jednoduché riešenie, ideálne tak, aby užívateľ nemal možnosť urobiť chybu a ak by ju urobil, tak systém ponúkne jednoduché riešenie.
- zaistiť jednoduché vrátenie akcií, aby sa umožnilo novým užívateľom experimentovať s aplikáciou
- pridať ovládacie prvky, ktoré by užívateľovi umožňovali manipulovať s aplikáciou

## 6.2 2. fáza testovania

Táto fáza testovania bola zameraná pre širšiu vzorku užívateľov a prebiehala až po dokončení väčšiny práce na aplikácií. Testovanie prebiehalo hlavne s dvoma vzorkami užívateľov.

Prvým predtým ešte nikdy nepracovali s bezdrôtovými sieťami, takže som im len vysvetlil základný princíp, a následne som ich nechal s aplikáciou pracovať. Spočiatku som len sledoval, ako sa s aplikáciou zoznamujú, čo im je nejasné. Neskôr som im zadal jednoduché úlohy, ako napríklad vyhľadať všetky senzory v Berlíne a sledovať teplotu alebo vytvoriť budovu s 5 podlažiami. Z tohto sledovania mi vzišli nasledujúce poznámky:

- spočiatku boli zmätení všetkými možnosťami v pravom paneli (prázdna stromová komponenta, zoznam označených uzlov,...)
- privítali by prívitejší formulár na vytvorenie uzlu aj s vysvetlivkami, čo jednotlivé položky znamenajú
- nejasnosť, kam zadať vyhľadávacie heslo, teda ktoré z troch možných polí na hľadanie zvolíť

Druhou skupinou, ktorá testovala OutdoorView boli užívatelia, ktorí už mali skúsenosti s aplikáciami pre prácu so sensorovými sieťami, konkrétne s aplikáciou IndoorView. Získané výsledky a pripomienky:

- pridať možnosť na vypnutie zobrazovania tabuliek s hodnotami dátových tokov
- pridať možnosť uzamknutia značiek, aby sa s nimi nedalo omylom pohnúť
- možnosť zväčšovať panel s grafom a tým zväčšovať aj samotný graf

- pridať tlačidlá na výber, resp odznačenie všetkých uzlov
- limitovať nájdené uzly podľa geografickej polohy

### 6.3 Ďalší vývoj

Hoci aplikácia splňa požadované parametre, stále tu je veľký priestor na možné inovovanie a rozšírenie do budúcnosti.

Jednou z takýchto možností by bola náhrada google máp vlastnými, napríklad katastrálnymi. Táto možnosť spolu s využitím nových technológií z HTML5 by umožňovali aplikácií čiastočne pracovať v offline móde a tým minimalizovať nároky na kvalitné internetové pripojenie. To by so sebou prinieslo možnosť aplikáciu efektívne používať aj v sťažených podmienkach.

Keď by sa aplikácia osvedčila, tak ďalšou možnou inováciou by bola podpora pre stálych užívateľov napríklad v podobe zavedenia skratiek na najčastejšie používané operácie alebo možnosť preusporiadania grafického rozhrania podľa osobných preferencií.

# Kapitola 7

## Záver

Hlavným cieľom tejto bakalárskej práce bolo vytvoriť webový portál, ktorý bude poskytovať vizualizačné rozhranie pre užívateľov bezdrôtových sensorových sietí s možnosťou lokalizovať uzly, sledovať aktuálne aj historické hodnoty na senzoch, vytvárať rozhranie pre nové uzly a budovy. Všetky tieto body zadania boli plnené s ohľadom na to, aby boli nielen dostatočne užívateľsky príjemné, ale zároveň aby poskytovali širšie množstvo nástrojov a rozšírení aj pre náročnejších užívateľov. Aplikácia je nasadená online, preto som sa snažil využiť všetky možnosti, ktoré toto prostredie ponúka, teda snaha o čo najväčšiu prenositeľnosť, multiplatformnosť a previazanie s ostatnými webovými službami ako Google Maps a Cosm. Medzi hlavné úspechy zaraďujem to, že už teraz si aplikácia našla prvých používateľov prinášajúcich prvé námety a pripomienky.

OutdoorView vzniklo v nadväznosti na projekt IndoorView vyvíjaný na Ústave Telekomunikácií Fakulty elektrotechniky a komunikačných technológií VUT v Brne. V spojení s touto aplikáciou vytvárajú plnohodnotný nástroj na sledovanie verejných ale aj súkromných uzlov v exteriéri aj interiéri. Súbor týchto nástrojov je dostupný na adrese <http://www.wsnapp.wislab.cz/>.

Senzorové siete sú rýchlo sa rozvíjajúca oblasť informatiky so stále väčším počtom užívateľov a rôznych využití, takže vzniká aj potenciál pre ďalšie rozširovanie aplikácie, dopĺňanie funkcionality podľa požiadaviek užívateľov a taktiež ďalší vývoj spomenutý v kapitole 6.3.

# Literatura

- [1] Akyildiz, I.: Wireless sensor networks: a survey. *Computer Networks*, ročník 38, 2002: s. 393–422, ISSN 1389-1286.
- [2] David Rousset: Introduction to HTML5 Web Workers: The JavaScript Multi-threading Approach. 2012.  
URL <http://msdn.microsoft.com/en-us/hh549259.aspx>
- [3] Jim Morrison: Browser support for html5 and css3. 2012.  
URL [http://deepbluesky.com/blog/-/browser-support-for-css3-and-html5\\_72/](http://deepbluesky.com/blog/-/browser-support-for-css3-and-html5_72/)
- [4] Kostiainen, A.: JavaScript Performance Best Practices. August 2012.  
URL [http://www.developer.nokia.com/Community/Wiki/JavaScript\\_Performance\\_Best\\_Practices](http://www.developer.nokia.com/Community/Wiki/JavaScript_Performance_Best_Practices)
- [5] Kreibich, O.: Bezdrátové senzorové sítě. *MM Průmyslové spektrum*, 2011: str. 38, ISSN 1212- 2572.
- [6] Matt Cronin: Modal Windows In Modern Web Design. 2009.  
URL <http://www.smashingmagazine.com/2009/05/27/modal-windows-in-modern-web-design/>
- [7] Pecinovský, R.: *Návrhové vzory*. Brno: Computer Press, 2013, ISBN 978-80-251-1582-4.
- [8] Shafranovich, Y.: RFC 4180: Common Format and MIME Type for Comma-Separated Values (CSV) Files. Říjen 2005.  
URL <http://tools.ietf.org/html/rfc4180#section-1>
- [9] Sohraby, K.; Minoli, D.: *Wireless sensor networks:Technology, Protocols, and Applications*. Hoboken: A John Wiley & Sons, 2007, ISBN 978-0-471-74300-2.
- [10] Wikipedia: HSL and HSV. April 2013.  
URL [http://en.wikipedia.org/wiki/HSL\\_and\\_HSV](http://en.wikipedia.org/wiki/HSL_and_HSV)

# Příloha A

## Obsah DVD

Na priloženom DVD sa nachádza:

- zložka *latex* – obsahuje zdrojové súbory technickej správy
- zložka *src* – obsahuje zdrojové kódy vrátane spustiteľnej webovej stránky
- zložka *video* – obsahuje video súbor s ukážkou práce a využitím OutdoorView
- súbor *technicka sprava.pdf* – táto technická správa