

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Webové rozhraní pro zobrazování sensorových dat ze  
sensorové sítě



2017

Martin Chytka

Vedoucí práce: Mgr. Martin  
Trnečka, Ph.D.

Studijní obor: Aplikovaná informatika,  
kombinovaná forma

## **Bibliografické údaje**

Autor: Martin Chytka  
Název práce: Webové rozhraní pro zobrazování sensorových dat ze sensorové sítě  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2017  
Studijní obor: Aplikovaná informatika, kombinovaná forma  
Vedoucí práce: Mgr. Martin Trnečka, Ph.D.  
Počet stran: 49  
Přílohy: 1 CD/DVD  
Jazyk práce: český

## **Bibliographic info**

Author: Martin Chytka  
Title: Web interface for displaying sensor data from sensor network  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2017  
Study field: Applied Computer Science, combined form  
Supervisor: Mgr. Martin Trnečka, Ph.D.  
Page count: 49  
Supplements: 1 CD/DVD  
Thesis language: Czech

## **Anotace**

*V Olomouci bude vytvořena experimentální sensorová síť, pomocí které se bude sledovat kvalita ovzduší. Cílem práce je navrhnout a implementovat rozhraní, kterým se budou zobrazovat naměřená data, tak aby to vyhovovalo požadavkům odborníků na zkoumání znečištění životního prostředí.*

## **Synopsis**

*In Olomouc will be created an experimental sensor network, through which it will monitor air quality. The aim is to design and implement an interface that will display measured data, so as to meet the requirements of specialists to investigate environmental pollution.*

**Klíčová slova:** sensorové sítě, sensorová data, internet věcí, webová rozhraní

**Keywords:** sensor networks, sensor data, Internet of things, the web interface

Děkuji Doc. Ing. Motyčkové, CSc. a Mgr. Martinu Trnečkovi, Ph.D. za vedení této bakalářské práce a za podnětné připomínky k vypracování. Taktéž děkuji Mgr. Hejlové za poskytnutí vstupních dat pro tuto aplikaci.

*Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

Datum odevzdání : květen 2017

Martin Chytka

# Obsah

<b>1</b>	<b>Úvod</b>	<b>9</b>
1.1	Používané zkratky a/nebo registrované značky . . . . .	9
1.2	Zásady pro vypracování . . . . .	10
1.3	Zadání práce . . . . .	11
1.4	Požadavky na aplikaci . . . . .	12
<b>2</b>	<b>Návrh použitých technologií a architektury aplikace</b>	<b>14</b>
2.1	Hardware použité pro sběr dat a datové úložiště . . . . .	14
2.2	Aplikační (webový) server . . . . .	18
2.3	Databázový server . . . . .	19
2.4	Webový klient (webový prohlížeč) . . . . .	19
2.4.1	Historie . . . . .	19
2.4.2	Statistiky rozšířenosti prohlížečů . . . . .	20
2.4.3	Přizpůsobování webových aplikací jednotlivým prohlížečům	20
2.4.4	Skriptování na straně klienta . . . . .	20
2.4.5	Multimédia . . . . .	21
2.5	Monitory, displeje a velikosti zobrazovacích prostorů . . . . .	21
<b>3</b>	<b>Nástroje pro vývoj webové aplikace</b>	<b>23</b>
3.1	Webový a databázový server na lokálním PC . . . . .	23
3.2	Nástroje pro tvorbu grafiky – grafika rastrová (bitmapová) . . . . .	23
3.3	Nástroje pro tvorbu grafiky – grafika vektorová . . . . .	23
3.4	Nástroje pro generování souborů s výstupy . . . . .	23
3.5	Programování PHP, kód HTML, SQL . . . . .	24
3.6	Kód CSS . . . . .	24
3.7	Softwarové nástroje použité pro tuto práci . . . . .	24
3.7.1	Webový a databázový vývojový a testovací server na lo- kálním PC . . . . .	24
3.7.2	Webový a databázový server v produkčním prostředí . . . . .	24
3.7.3	Programování PHP, kód HTML, SQL, CSS . . . . .	25
3.7.4	Nástroje pro tvorbu grafiky . . . . .	25
<b>4</b>	<b>Návrh realizace zobrazování dat v aplikaci</b>	<b>26</b>
4.1	Grafika rastrová . . . . .	26
4.2	Grafika vektorová . . . . .	27
4.3	Přehled webového software pro generování grafů . . . . .	28
4.4	Vlastní řešení zobrazování dat . . . . .	29
4.5	Realizace zobrazování dat . . . . .	29
<b>5</b>	<b>Návrh funkčnosti aplikace</b>	<b>31</b>

<b>6</b>	<b>Popis aplikace</b>	<b>33</b>
6.1	Výčet stran aplikace . . . . .	33
6.1.1	Mapa rozmístění uzlů . . . . .	33
6.1.2	Grafy měření . . . . .	34
6.1.3	Výpočty hodnot . . . . .	35
6.1.4	Počty měření . . . . .	36
6.1.5	Opravy souborů dat . . . . .	36
6.1.6	Přiřazení ID měření . . . . .	36
6.1.7	Parametry sítě . . . . .	36
6.2	Data a databáze . . . . .	37
6.3	Zdrojové kódy aplikace . . . . .	39
6.4	Ukázka SVG kódu grafu část 1 . . . . .	41
6.5	Ukázka SVG kódu grafu část 2 . . . . .	42
<b>7</b>	<b>Cíle práce</b>	<b>43</b>
<b>8</b>	<b>Porovnání s podobnými pracemi</b>	<b>44</b>
	<b>Závěr</b>	<b>45</b>
	<b>Conclusions</b>	<b>46</b>
<b>A</b>	<b>Příloha 1 - zprovoznění webové aplikace na serveru (readme.txt)</b>	<b>47</b>
<b>B</b>	<b>Obsah přiloženého CD/DVD</b>	<b>48</b>
	<b>Bibliografie</b>	<b>49</b>

## Seznam obrázků

1	Brána Meshlium je Linux router, který pracuje jako brána senzorové sítě. Může obsahovat 5 radiových rozhraní: WiFi 2.4GHz, WiFi 5GHz, 3G/GPRS, Bluetooth, ZigBee. Zdroj: prospekty Libelium . . . . .	15
2	Uzly se senzory Waspnote Plug & Sense! zdroj: prospekty Libelium	16
3	Příklad umístění uzlů se senzory Waspnote Plug & Sense! Zobrazeny jsou uzly s externím a interním solárním panelem pro dobíjení baterií. Řešení se solárními panely by bylo vhodné použít v budoucnu. Zdroj: prospekty Libelium . . . . .	17
4	Příklad zobrazení webové stránky na různých zařízeních názorně ukazuje jak rozdílným zobrazovacím prostorům se musí stránka být schopna přizpůsobovat. Zdroj: W3Schools . . . . .	22
5	Použití rastrového obrázku ve formátu JPEG v původním vysokém rozlišení. Zdroj: vlastní řešení . . . . .	26
6	Použití rastrového obrázku ve formátu JPEG v nízkém rozlišení po kompresi ukazuje ztrátu kvality obrazu. Zdroj: vlastní řešení .	27
7	Ukázka bezztrátové kvality SVG grafiky - max zvětšení. Zdroj: vlastní řešení . . . . .	28
8	Ukázka grafu měření vygenerovaného aplikací, strana Grafy měření. zdroj: Webové rozhraní pro zobrazování sensorových dat ze sensorové sítě, která je předmětem této práce. . . . .	34
9	Ukázka grafu měření vygenerovaného aplikací, strana Výpočty hodnot. zdroj: Aplikace Webové rozhraní pro zobrazování sensorových dat ze sensorové sítě, která je předmětem této práce. . . .	35

## Seznam tabulek

1	Parametry bezdrátové sensorové sítě . . . . .	11
2	Vzorečky pro výpočet meteorologických hodnot . . . . .	11
3	Typy uzlů Waspnote a jejich počty k dispozici . . . . .	17
4	Podíly prohlížečů na světovém trhu . . . . .	20
5	Čísla a ID použitých uzlů . . . . .	33
6	Popis adresářů aplikace . . . . .	39
7	Popis HTML souborů aplikace . . . . .	39
8	Popis PHP souborů aplikace . . . . .	40

## Seznam zdrojových kódů

1	vlastnost <b>Media Queries</b> specifikace CSS3 . . . . .	22
---	---	----

2	SQL kód vytvoření tabulky DBS . . . . .	37
3	SVG kód grafu 1 . . . . .	41
4	SVG kód grafu 2 . . . . .	42



# 1 Úvod

## 1.1 Používané zkratky a/nebo registrované značky

Vzhledem k potřebě často vyjmenovávat použité technologie, je zde uveden seznam používaných zkratk, nebo vysvětlivek.

1. IDE - Integrated Development Environment
2. IP - Internet Protocol
3. DTP - technologie, postupy a procesy pro předtiskovou přípravu různých tiskovin
4. CSV - Comma Separated Value
5. XLS – soubor tabulkového editoru Excel
6. IoT - Internet of Things
7. JSON - JavaScript Object Notation
8. URL - Uniform Resources Locator
9. URI - Uniform Resources Identifier
10. webový server - pokud nebude řečeno jinak, myslí se zde software server
11. LAMP - serverová SW technologie Linux, Apache, MySQL, PHP
12. WAMP - serverová SW technologie Windows, Apache, MySQL, PHP
13. HTML – hypertext markup language, značkovací jazyk pro popis struktury webu
14. HTML5 - poslední uvolněná specifikace značkovacího jazyka HTML
15. HTTP - Hypertext Transfer Protocol
16. HTTPD (httpd) - HTTP Deamon (Apache HTTP Server)
17. HTTPS - Hypertext Transfer Protocol Secure (HTTP over SSL)
18. XHTML - specifikace HTML jako podmnožiny v syntaxi XML
19. XML - rozšiřitelný značkovací jazyk
20. PHP - interpretovaný jazyk na straně serveru
21. JavaScript - interpretovaný jazyk na straně klienta (webového prohlížeče)

22. Ajax - Asynchronous JavaScript and XML, technologie kombinující JS s načítáním XML dat ze serveru, což umožňuje načítání dat do webové stránky bez jejího kompletního znovunačtení
23. CSS - Cascading style sheets - značkovací jazyk pro formátování HTML dokumentů
24. SQL - Structured Query Language - neprocedurální dotazovací databázový jazyk
25. MySQL - webová databázová aplikace (MySQL AB, Sun, od r. 2012 Oracle)
26. MariaDB – fork MySQL, pokračování open source vývoje MySQL
27. PhpMyAdmin - freeware webová aplikace pro správu DB MySQL
28. Apache - webový server (SW)
29. JPG, JPEG, GIF, PNG - grafické rastrové (bitmapové) formáty
30. SVG, AI - grafické vektorové formáty
31. W3C – konsorcium pro webové standardy
32. www – world wide web

## 1.2 Zásady pro vypracování

- Navrhněte a implementujte webové rozhraní pro zobrazování meteorologických dat
- Data budou snímána v reálném čase senzorovou sítí a ukládána v databázovém úložišti
- Způsob zobrazování naměřených dat bude sloužit k monitorování znečištění životního prostředí

### 1.3 Zadání práce

V následujících dvou podkapitolách je uveden přehled požadavků na aplikaci od Mgr. Hejlové, jakožto řešitelky plánovaného projektu Měření hodnot vzdušných polutantů v centru Olomouce.

#### Aplikace pro zobrazování dat z uzlů Wasmote Plug & Sense!

Tabulka 1: Parametry bezdrátové sensorové sítě

Parametry bezdrátové sensorové sítě	
Počet uzlů	13
Plocha pokrytí	cca 0,5 km <sup>2</sup>
Počet zaznamenaných prvků (senzorů na každém uzlu)	6
Interval záznamu	10 minut <sup>1</sup>
Shromaždiště dat	lokální databáze v bráně Meshlium <sup>2</sup>
Topologie sítě	stromová
Počet zaznamenaných hodnot v jednom okamžiku	78
Počet zaznamenaných hodnot denně	11232

Tabulka 2: Vzorečky pro výpočet meteorologických hodnot

Vzorečky pro výpočet meteorologických hodnot	
Hodinová průměrná teplota	suma teplot za hodinu / počet hodnot
Maximální denní teplota (T <sub>max</sub> )	nejvyšší teplota dne
Minimální denní teplota (T <sub>min</sub> )	nejnižší teplota dne
Teplotní denní amplituda	$(T_{\max} - T_{\min})/2$
Denní průměrná teplota z klimatologických termínů měření	$(t_7 + t_{14} + t_{21} + t_{21}) / 4$ - číslo u t značí čas zaznamenání teplotní hodnoty.
Denní průměrná teplota ze všech měření	suma všech teplot dne / počet záznamů měření
Hodinová průměrná vlhkost	suma vlhkostí za hodinu/počet hodnot
Maximální denní vlhkost	nejvyšší vlhkost dne
Minimální denní vlhkost	nejnižší vlhkost dne
Denní relativní vlhkost z klimatologických termínů měření	$(rv_7 + rv_{14} + rv_{21} + rv_{21}) / 4$ - číslo u rv značí čas zaznamenání teplotní hodnoty
Denní průměrná vlhkost ze všech měření	suma všech vlhkostí dne / počet záznamů měření

Kromě teploty a relativní vlhkosti budou měřeny: NO<sub>2</sub>, CO<sub>2</sub>, VOC a další vzdušné polutanty jako CO a kyseliny, které jsou dopočítávány na posledním čidle hromadě. U těchto polutantů by bylo vhodné zařadit vyhledání nejnižších a nejvyšších denních hodnot a výpočet průměrné denní hodnoty polutantu.

## 1.4 Požadavky na aplikaci

- Aplikace bude sloužit jako grafické rozhraní k zobrazení dat z uzlů Waspote Plug & Sense!, které posílají data na bránu Meshlium.
- Úvodní stránka bude zobrazovat přehledovou mapu, kde bude vymezena zájmová oblast, která se nachází v centru města Olomouce, k dispozici je statická mapa oblasti (JPG). Kromě přehledové mapy se na této stránce bude nacházet mapa zobrazující místa, kde jsou umístěny senzory, které budou popsány, aby bylo zřejmé, ze kterého místa pocházejí v grafech zobrazovaná data. Mapa by byla ideální dynamická, API na Atlase, Seznamu či Googlu<sup>3</sup>.
- Po kliknutí na bod, kde se nachází uzel, by se měl zobrazit graf průběhu hodnot v daném místě.
- Další stránka bude obsahovat popis měřených prvků se vzorečky výpočtu a grafy, které budou zobrazovat data okamžitě po zaslání v desetiminutovém intervalu<sup>4</sup>.
- Bylo by vhodné vykreslovat 14 grafů. Jeden graf bude souhrnný, a bude v něm možné zobrazit všechna data ze všech stanic za posledních 48 hodin. Jelikož se bude jednat o hodně hodnot, tak bych volila zaškrtačací políčka, ať si uživatel může zvolit, že chce zobrazit pouze teploty na všech stanicích nebo teploty a relativní vlhkosti na dvou vybraných stanicích. 13 grafů bude zobrazovat hodnoty naměřené na jednotlivých uzlech.

---

<sup>1</sup>V případě nasazení aplikace online. Hardwarové řešení ovšem v době této práce nebylo k dispozici, tudíž jsou využita data z experimentálních měření.

<sup>2</sup>Brána Meshlium má vlastní databázi MySQL a může sloužit jako prvotní úložiště dat ze senzorů.

<sup>3</sup>Vzhledem k neměnné oblasti umístění senzorů netřeba, data z měření byla získána ze stabilního umístění senzorů, které bylo později pro jiná s touto prací nesouvisející měření a kvůli rekonstrukci rozária botanické zahrady přesunuta jinam.

<sup>4</sup>Vzhledem k tomu, že podpůrné práce na zprovoznění senzorů a na databázovém úložišti nebyly v době psaní této práce dokončeny a připojení nebylo k dispozici, nakonec rozhraní používá jen data získaná z měření několika dní v dubnu 2014.

- Každý uzel může mít zvolenu jednu barvu, aby byly uzly jednoznačně odlišeny<sup>5</sup> a bylo zřejmé, ke kterému uzlu se vztahují naměřená data. Při najetí na bod grafu se v desetiminutových intervalech zobrazí naměřená data<sup>6</sup>.
- V grafech by se mělo dát zoomovat. Kromě grafů bude možnost zobrazení tabulky naměřených dat, kterou si uživatel bude moct stáhnout. Jako doplňující funkce budou výpočty výše uvedených základních parametrů jednotlivých meteorologických prvků. Tyto vypočtené hodnoty budou vypočítávány v tabulce a bylo by vhodné je také zařadit do výběracích polí pro zobrazení v úvodním grafu nebo udělat jeden prázdný graf, který bude umožňovat zobrazit tyto dopočtené hodnoty.

---

<sup>5</sup>Jelikož vykreslovaných veličin mělo být ve finále cca 7 a uzlů může být v reálném nasazení více, nebude asi možno mít pro takovéto rozlišení dostatek vhodných barev. Barvy bude potřeba na rozlišení křivek veličin, uzly mají svá 9-ti místná identifikační čísla, takže rozlišení bude asi lépe navrhnout číslem

<sup>6</sup>Toto by bylo možné, pokud by aplikace fungovala online, ovšem v návaznosti na výše uvedené toto nemá smysl. Po najetí myší nad uzel grafu by bylo možno zobrazovat třeba hodnoty uzlu.

## 2 Návrh použitých technologií a architektury aplikace

Z požadavků v zadání vyplývá, že se bude jednat o webovou aplikaci s využitím databáze, tudíž jsou některé způsoby řešení a navazující technologie předem dané.

*Architektura webové databázové aplikace se může skládat z několika vrstev, přičemž implementace může být takto po jednotlivých vrstvách realizována a fyzické rozdělení může kopírovat uvedené vrstvy, nebo některé, nebo i všechny vrstvy mohou být sloučeny hardwarově i softwarově a všechny potřebný software může běžet fyzicky na jednom stroji <sup>[1]</sup>*

Obecně nejběžněji používaná řešení jsou 2, podle toho, na jakém operačním systému běží server aplikace. Podrobněji bude problematika rozebrána níže, ale ve zkratce se jedná o kombinace:

- Linux, Apache, MySQL, PHP (LAMP)
- Windows, IIS Windows Server, .NET, MSSQL

Je možno také použít software různě kombinovat, např. software Apache, PHP, MySQL může běžet i na operační systém Windows, nebo na jednotlivých vrstvách může být použito jiné software, třeba jiná databáze.

*Rozdělení webové databázové aplikace do vrstev je většinou tříúrovňové <sup>[1]</sup> :*

1. *aplikační (webový) server (API pro připojení k databázi a podporující nějaký serverový skriptovací jazyk, logika aplikace, skripty)*
2. *databázový server (databáze, ke které přistupují aplikace z aplikačního serveru)*
3. *webový klient (webový prohlížeč)*

### 2.1 Hardware použité pro sběr dat a datové úložiště

Hardware použité pro sběr dat a datové úložiště, pro které je tato aplikace vytvořena nejsou předmětem této práce, uvedeme proto jen stručný popis, ke kterému použijeme citace z některých kapitol příspěvku Mgr. Hejlové<sup>7</sup> na konferenci GIS Ostrava 2015 „BEZDRÁTOVÁ SENZOROVÁ SÍŤ SLEDUJÍCÍ ZNEČIŠTĚNÍ OVZDUŠÍ“.

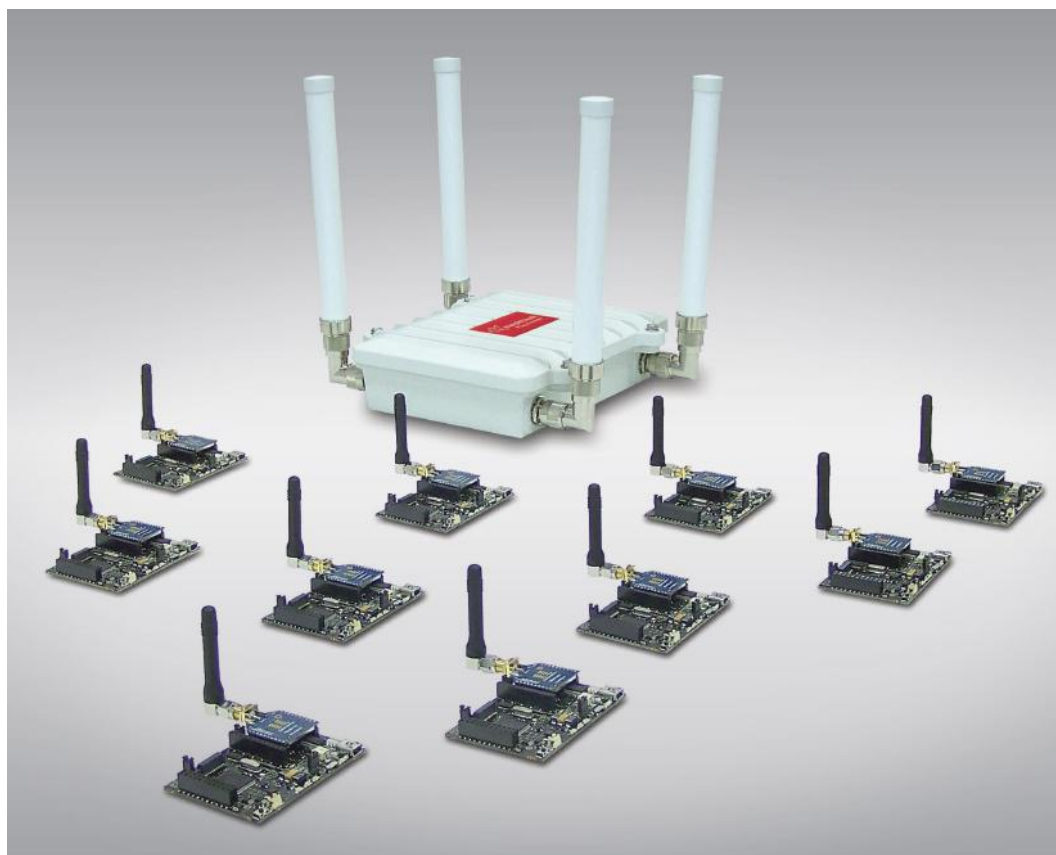
---

<sup>7</sup>Mgr. Hejlová je autorkou práce o bezdrátové sensorové síti pro kterou byla tato aplikace napsána

## BEZDRÁTOVÁ SENZOROVÁ SÍŤ

Bezdrátová senzorová síť je novou technologií sběru dat. Každá bezdrátová senzorová síť se skládá ze tří základních částí, mezi které patří: uzly, brána a server. Uzly jsou malé jednotky vybavené komunikační jednotkou, mikroprocesorem, pamětí, baterií a rozhraním pro napojení senzorů nebo senzorových desek (Büttrich, 2010, Murthy, 2004). Na uzly je možné napojit rozličné množství senzorů a rozšířit tak spektrum aplikací, kde je tato technologie použita. Uzly sbírají a posílají data na druhou část bezdrátové senzorové sítě, kterou je brána. Některé uzly shromažďují data z více na ně napojených uzlů a přeposílají je na bránu. Tyto uzly jsou označovány jako směrovací uzly (routery). Schéma komunikačního toku dat mezi uzly až k cíli, k bráně, je definováno topologií bezdrátové senzorové sítě. V bráně jsou informace dočasně uloženy, připraveny k dalšímu zpracování nebo jsou pomocí ethernetové komunikace posílány na server, kde jsou trvale uloženy. Komunikace v bezdrátové senzorové síti probíhá na principu rádiového vlnění, kterého využívá protokol operující ve vybraných frekvencích (Podhorský, 2012, Ma, 2008).

### POUŽITÉ UZLY



Obrázek 1: Brána Meshlium je Linux router, který pracuje jako brána senzorové sítě. Může obsahovat 5 radiových rozhraní: WiFi 2.4GHz, WiFi 5GHz, 3G/GPRS, Bluetooth, ZigBee. Zdroj: prospekty Libelium

Uzly, které byly použity k monitoringu, se nazývají Wasmote Plug & Sense! Tato komponenta je produkována španělskou společností Libelium Comunicaciones Distribuidas S.L. Uzly Wasmote Plug & Sense! jsou vyráběny v několika řadách a ke každé řadě je možné připojit jiné senzory. Používané uzly, na které je možné napojit senzory pro monitorování vzdušných polutantů, pocházejí z řady Wasmote Plug & Sense! Smart Environment a Wasmote Plug & Sense! Smart Agriculture. Na uzly z řady Smart Environment je možné napojit senzory sledující polutanty produkované dopravou (CO<sub>2</sub>, VOC, NO<sub>2</sub>) a vybrané meteorologické prvky (teplotu, relativní vlhkost a tlak vzduchu). Uzly z řady Smart Agriculture umožňují napojit malou meteorologickou stanicí sledující srážky, směr a rychlost větru a senzory monitorující teplotu, relativní vlhkost a tlak vzduchu. Komunikace mezi uzly byla uskutečňována v pásmu 3,4 GHz a využívala protokolu 802.15.4. Každý uzel i brána má svou MAC adresu a jednoznačné ID, aby došlo při komunikaci k jednoznačné identifikaci jednotlivých uzlů a brány. Všechny řady Wasmote Plug & Sense! jsou již zapouzdřeny, a tak jsou vhodné k přímému venkovnímu použití. Data z uzlů vybavených senzory jsou zasílána na bránu – Meshlium, která je od stejného výrobce. Tato brána je také určena do venkovních podmínek, je schopna uložit až 2 GB naměřených hodnot v lokální MySQL databázi a komunikovat se vzdáleným serverovým úložištěm (Wasmote Plug & Sense!, 2013).



Obrázek 2: Uzly se senzory Wasmote Plug & Sense! zdroj: prospekty Libelium



K výše uvedeným citacím lze dodat, že od r. 2014 proběhlo několik měření v areálu rozária botanické zahrady na třídě 17. listopadu v Olomouci. Jednalo se o experimentální nasazení, při kterém data nebyla ukládána na databázovém serveru, ale byla pouze ručně stahována z databáze v bráně Meshlium a vy-exportována do souborů formátu CSV. Tato aplikace tudíž není připojována k datovému úložišti, které v té době nebylo k dispozici, ale používá data z měření v r. 2014 stažená jako soubor.

Původně bylo v plánu použití celkem až 15 uzlů s multisenzory Waspnote fy Libelium CD S.L., s možností měřit 4 - 10 veličin na jednotlivých uzlech.

Tabulka 3: Typy uzlů Waspnote a jejich počty k dispozici

2x	Smart Agriculture (4 senzory)
2x	Smart Agriculture PRO (4 senzory)
11x	Smart Environment (vyplněny všechny konektory na senzory (6), možnost měřit cca 10 veličin)



Obrázek 3: Příklad umístění uzlů se senzory Waspnote Plug & Sense! Zobrazeny jsou uzly s externím a interním solárním panelem pro dobíjení baterií. Řešení se solárními panely by bylo vhodné použít v budoucnu. Zdroj: prospekty Libelium

Postupem času se plán měření měnil a v době zadání této práce plán sítě počítal s měřením celkem 7-mi veličin, kterými byly: teplota, relativní vlhkost vzduchu, NO<sub>2</sub>, CO<sub>2</sub>, CO, VOC a kyseliny. Z těchto veličin byly nakonec v dubnu 2014 měřeny jen 2. Teplota a relativní vlhkost vzduchu.

Při použití četnosti měření 6 za hodinu v 10-ti minutových intervalech, měl 1 měřený den obsahovat 144 záznamů pro každou veličinu. Tolik záznamů neměl na konci měření zaznamenáno žádný uzel. Žádná data z žádného uzlu nebyla

kompletní, je však k dispozici celkem 6 měření po 143 záznamech za den a nejvíce jich bylo ve 2 dnech po sobě, 12.4. a 13.4. na uzlu č.0 - 382537776.

Od počátku bylo měření provázáno potíží s datovými výstupy a všechna takto získaná data jsou poškozená. Příčinou byly většinou problémy s bateriovým napájením senzorů, kteréžto nebylo schopno udržet senzorové uzly v provozu bez výpadků. Tento problém bude v budoucnu pro spolehlivé nasazení senzorových sítí stěžejní a zabývá se jím samostatná bakalářská práce J. Koníčka (Katedra geoinformatiky UP Olomouc) z r. 2016 <sup>8</sup>. Problémy s napájením senzorových uzlů lze řešit solárními panely pro dobíjení baterií.

Data z exportovaných souborů byla pročištěna od pro tuto práci nepotřebných údajů, chybná měření byla odstraněna, nebo opravena a výsledná data byla naimportována do databázového serveru.

## 2.2 Aplikační (webový) server

*Jako webový server je stále nejčastěji používaným softwarem server Apache často na platformě Linux. Dalšími řešeními může být např. server FHTTPd na unixových systémech, nebo server IIS Windows Server. Apache může běžet nejen na Linuxu, ale také na Unixu, či Windows <sup>[1]</sup>. Dle Wikipedie byl od r. 2011 IIS druhým nejpoužívanějším serverem na světě po prvním Apache HTTP <sup>[9]</sup>, kdy rozšíření IIS na webových serverech dosahovalo 16 %.*

Z hlediska aplikační logiky se serverem Apache souvisí skriptovací jazyk na straně serveru PHP. Na serverech IIS je možno provozovat Microsoft technologii .NET a využívat tak i jiné programovací jazyky, předchůdcem .NET bylo ASP NET.

Pro potřeby této práce se budeme zabývat variantou s jazykem PHP na serveru, předpokládá se že na Apache, ale toto již bude v kompetenci vybraného poskytovatele webhostingu, není zadáním této práce budovat pro aplikaci vlastní webhostingové řešení, proto z tohoto hlediska také nebude podstatné, jestli PHP (Apache) poběží na Linuxu, nebo jiné platformě. Finální webhosting aplikace zatím ještě není znám, ale v současné době je nepřehledné množství poskytovatelů, kteří jsou schopni zajistit kvalitní hosting a to při velmi nízkých uživatelských nákladech. Bylo by jistě možno také aplikaci hostovat na serverech UP, pokud by o to byl zájem a podmínky. Pro potřeby prezentace této práce byl prozatím vybrán běžný komerční poskytovatel webhostingu, jak je uváděno dále v textu.

---

<sup>8</sup>KONÍČEK, Jakub. Optimalizácia energetického zaťaženia uzla waspmote plug & sense s ohľadom na vybrané faktory. 2016. Katedra geoinformatiky UP Olomouc

## 2.3 Databázový server

Nejrozšířenějším databázovým řešením pro webové aplikace v PHP je MySQL. Poté, co práva na MySQL koupil v r. 2008 Sun a v r. 2009 byl Sun koupěn Oraclem, vznikla odnož MySQL která se snaží o zachování open source pokračování MySQL - MariaDB. Dalšími používanými databázemi jsou např. PostgreSQL, nebo databázová řešení Oracle. Pro aplikace na platformě .NET a Windows je pak k dispozici MSSQL. Pro správu MySQL se používají různé nástroje, velmi rozšířená je webová aplikace PhpMyAdmin, „štíhlejší“, ale rychlejší Adminer, další možností je např. desktopová aplikace HeidiSQL. Možná je také správa databáze příkazovým řádkem z konzole.

Pro potřeby této práce bude plně dostačovat MySQL, které splňuje požadavky na aplikaci kladené. Databázové úložiště pro ukládání senzorových dat je předmětem další samostatné kvalifikační práce, ale v době psaní této práce nebyla databáze k dispozici.

Vybrané senzorové řešení, fy Libelium, resp. jeho brána Meshlium router poskytuje také vlastní lokální databázové úložiště MySQL o kapacitě 2GB<sup>9</sup> tudíž by bylo teoreticky možné aplikaci připojovat přímo k němu. Záleží ovšem, na jak dlouho by vystačila kapacita databáze, pokud by aplikace běžela v ostrém provozu třeba i několik let. Řešení se samostatným databázovým serverem nabízí více možností pro další práci s daty.

## 2.4 Webový klient (webový prohlížeč)

### 2.4.1 Historie

Mezi dnešní nejpoužívanější webové prohlížeče patří (v abecedním pořadí) Google Chrome, Mozilla Firefox, Microsoft Edge<sup>10</sup>, Microsoft Internet Explorer<sup>11</sup>, Opera, Safari a další, ve velmi široké řadě verzí.

Podíl rozšíření prohlížečů se za posledních více jak 20 let měnil a některé prohlížeče, které dřív byly oblíbené a hojně rozšířené dnes již neexistují. Jedná se zejména o prohlížeče Mosaic, Netscape (Navigator a další) a Mozilla, které dnes znají asi již jen pamětníci. V první polovině 90 let byl nejrozšířenějším prohlížečem Netscape Navigator, ovšem koncem devadesátých let místo nejrozšířenějšího prohlížeče zaujal Internet Explorer od fy Microsoft. To bylo tehdy způsobeno hlavně tím, že IE byl součástí balíku dodávaného s operačním systémem Windows a při instalaci operačního systému byl nastaven jako výchozí prohlížeč. Mozilla Foundation vydala v r. 2004 prohlížeč Mozilla Firefox, původně zamýšlený jako malý, rychlý a jednoduchý prohlížeč vyvíjený vedle stávajícího velkého balíku Mozilla Suite. Postupem času z Firefoxu jeho vzrůstající obliba udělala stěžejní produkt této organizace. Firefox později začal v procentuelním podílu prohlížečů získávat převahu nad Microsoft Internet Explorer a s nástupem mo-

<sup>9</sup>Dokumentace k senzorům Waspmote a k bráně Meshlium fy Libelium

<sup>10</sup>Microsoft Edge – od Windows 10

<sup>11</sup>Microsoft Internet Explorer – do Windows 8

bilních zařízení na operačním systému Android zase začaly dominovat prohlížeče na jádru Blink a WebKit, tedy Google Chrome a Safari od Apple.

#### 2.4.2 Statistiky rozšířenosti prohlížečů

Statistiky rozšířenosti jednotlivých prohlížečů se různí a jiné výsledky budou pro ČR a jiné pro zbytek světa. Vzhledem k tomu, že v dnešní době je už samozřejmým požadavkem, aby webová aplikace nebyla závislá na žádném konkrétním prohlížeči a hlavně, že dnešní prohlížeče po dlouhých letech nekompatibility již dobře splňují webové standardy to není podstatné.

Ke konci roku 2016 např. na webu organizace StatCounter <sup>[10]</sup> jsou uváděny podíly na světovém trhu:

Tabulka 4: Podíly prohlížečů na světovém trhu

Google Chrome	58 %
Mozilla Firefox	16 %
Apple Safari	9 %
Microsoft Internet Explorer 11	8 %
Microsoft Edge	4 %
ostatní	zbytek

#### 2.4.3 Přizpůsobování webových aplikací jednotlivým prohlížečům

V začátcích webu bylo častou praxí odladit layout webu pro některý konkrétní prohlížeč, protože rozdíly výsledků zobrazení jednoho HTML kódu mezi jednotlivými prohlížeči byly často značné. CSS bylo nové a ne moc dobře podporováno, tudíž do konce 90 let bylo často jediným řešením tabulkové layouty. Tyto byly založeny na použití často i do sebe vnořených tabulek jakožto mřížky, která byla schopna zajistit stejné zobrazení webu jak v prohlížečích Netscape, tak Microsoft Internet Explorer. To ovšem bylo často vykoupeno vysokým počtem prohřešků HTML kódu proti specifikaci.

S postupným prosazováním se webových standardů a hlavně se schopnostmi prohlížečů tyto standardy korektně zobrazovat přišlo konečně období, kdy lze snadno dosáhnout stejného, nebo velmi podobného výsledku při zobrazení layoutů webu v různých prohlížečích.

#### 2.4.4 Skriptování na straně klienta

Jakkoli je situace s dodržováním HTML5 standardu v prohlížečích již vyhovující, stále přetrvávají rozdíly ve zpracování JavaScriptu. Vzhledem k masivnímu nástupu mobilních zařízení<sup>[11]</sup> (v říjnu 2016 poprvé v historii zaznamenal internet více přístupů z mobilních zařízení, než z desktopů) a vzhledem k tomu, že Google Chrome má celosvětově rozšíření přes 50 %, zároveň dobrou podporu

zpracování JavaScriptu a také je vysoce hodnoceným prohlížečem z hlediska dodržování standardů HTML5<sup>[12]</sup>, se často stává, že některé JavaScriptové aplikace jsou funkční jen v Google Chrome a potom např. Ve Firefoxu fungují buď omezeně, nebo vůbec. Pro potřeby této práce není klientské skriptování použito, proto se jím nebudeme dále zabývat.

#### 2.4.5 Multimédia

Dalším druhem obsahu, který může i dnes ještě vyžadovat rozdílné zdrojové kódy pro různé prohlížeče je zobrazování multimédií v prohlížečích. Pro přehrávání multimédií se na webu již od počátku jako v podstatě standard rozšířil Adobe Flash, který v kombinaci se skriptovacím jazykem Action Script nabízel i možnost tvorby dynamických rozhraní a např i her, nebo jiných aplikací. Ovšem poté co firma Apple před několika lety prohlásila, že žádné její zařízení nebude Adobe Flash podporovat, začal se na webu rozšiřovat W3C standard HTML5 video. Což je jistě dobře z hlediska standardu, který by se bez tohoto kroku rozšiřoval pomaleji, nebo vůbec, ale zase se minimálně na přechodné období komplikuje situace pro vývojáře, kteří z hlediska zpětné kompatibility musí nabízet multimédia ve více formátech. Adobe Flash má zatím navíc stále silnou podporu u významného poskytovatele multimediálního obsahu Youtube. Při požadavku na vložení videa z Youtube do jiné webové stránky je tak sice možno použít HTML5 tag video, ale pro starší prohlížeče je vždy lépe vkládat i alternativní zápis tagem OBJECT pro vložení Adobe Flash. V této aplikaci se však s využitím multimédií nepočítá.

### 2.5 Monitory, displeje a velikosti zobrazovacích prostorů

Dalším samostatným problémem dnešní doby jsou rozměry displejů mobilních zařízení. Zatímco na desktopech a noteboocích je při dnešních širokých monitorech dost prostoru pro zobrazení, na mobilních zařízeních s mnohem menšími displeji je nasnadě samozřejmý problém s mnohem menším zobrazovacím prostorem.

Vzhledem k požadavkům na zrychlující se vývoj webů a vzhledem k rozšiřující se množině druhů zařízení tak stejně jako v počátcích webu v podstatě stále přetrvává situace, že často není možno ke všem webovým aplikacím přistupovat z libovolných zařízení.

Také vzhledem k očekávanému nástupu tzv. internetu věcí, jehož jsou weby pro sensorové sítě součástí, se dá předpokládat, že přes všechny snahy o standardizaci se tato situace bude postupem času minimálně komplikovat, ne-li spíše i zhoršovat, pokud výrobci např. netradičních menších zařízení nebudou dodržovat potřebné webové standardy.

Situaci je nutno řešit tzv. responzivním designem webu (viz obr. 4, str. 22). Stránka je při správně napsaném layoutu dobře řízeně degradovatelná a vhodně se přizpůsobuje zobrazovacímu prostoru. Dá se říci, že nutným základem pro splnění požadavku takového „plovoucího“ layoutu je hlavně textový obsah s šířkou stránky nastavenou na jednu určitou konkrétní hodnotu a grafika tvořena malými obrázky, např do 150 px šířky. Další nutností je použití vhodně



Obrázek 4: Příklad zobrazení webové stránky na různých zařízeních názorně ukazuje jak rozdílným zobrazovacím prostorům se musí stránka být schopna přizpůsobovat. Zdroj: W3Schools

voleného předpisu CSS a hlavně vlastnosti Media Queries specifikace CSS3:

- 1 @media <požadovaná hodnota> and (max-width: <požadovaná hodnota>)
- 2 {požadované CSS předpisy pro maximální šířku <požadovaná hodnota>}

Zdrojový kód 1: vlastnost **Media Queries** specifikace CSS3

Tento styl zápisu zajistí, že je možno verzovat vlastnosti zobrazení pro různé šířky monitoru a nejdůležitější je, že lze pro různé velikosti displejů skrývat některý obsah. Je tak možno připravit bohatou podkladovou grafiku pro široké monitory a pro malé displeje např tuto bohatou, ale obsahově bezvýznamnou grafiku vůbec nezobrazovat. Uživatel na malém displeji tak uvidí jen obsahově podstatné věci.

Toto je poměrně v rozporu s požadavky na zobrazování meteorologických dat, protože obrázky (např. grafy) budou muset být větší a s šířkou 150 px by nebylo možno si vystačit.

## 3 Nástroje pro vývoj webové aplikace

Pro tvorbu webových aplikací lze použít různé nástroje, existují komerční nástroje s bohatou výbavou, ale v dnešní době lze dobře použít i širokou řadu freeware nástrojů.

### 3.1 Webový a databázový server na lokálním PC

Pro vývoj webových aplikací na lokálním PC (pro simulaci aplikačních a databázových serverů) se používá různých freeware balíčků distribucí software serveru Apache, skriptovacího jazyka na straně serveru PHP a databáze MySQL, běžících jak na Windows, tak pod jinými operačními systémy. Tyto balíčky lze instalovat pomocí jednoho společného instalátoru a odpadá tak instalace a nastavování konfigurace jednotlivých částí zvláště. Ze starších edicí lze zmínit např. balíčky PHPtriad, EasyPHP, nebo (český) Complex Web Server. Dnes je rozšířeným balíčkem např. XAMPP, nebo VertrigoServ. Distribuce obsahují software server Apache, PHP, MySQL a PhpMyAdmin pro správu databáze.

### 3.2 Nástroje pro tvorbu grafiky – grafika rastrová (bitmapová)

Zřejmě nejznámějším a nejpoužívanějším nástrojem pro ruční tvorbu rastrové grafiky je dnes **Adobe Photoshop**. Z freeware lze použít např. **GIMP** a pro základní rychlé úpravy grafiky, např. pro doostřování obrázků a snížení jejich datové velikosti na minimum, velmi dobře poslouží **IrfanView**. Ačkoliv jsou dnes přenosové rychlosti sítí vysoké, vždycky se vyplatí snaha o snížení velikosti přenášených dat webové aplikace. Pro generování rastrové grafiky grafů se na serveru používá knihovna PHP **GD2**, která je součástí distribuce PHP.

### 3.3 Nástroje pro tvorbu grafiky – grafika vektorová

Pro ruční vytváření souborů v oblasti vektorové grafiky jsou často používanými nástroji **Adobe Illustrator** a **Corel Draw**. V případě potřeby by bylo pro ruční kreslení a editaci vygenerovaného grafu ve vektorovém formátu možno použít např. freeware **Inkscape**. Pro generování vektorové grafiky grafů na serveru je možno použít PHP bez potřeby použití dalších doplňkových knihoven, nebo nadstavbe, jedná se o standardní dynamické generování XML souboru.

### 3.4 Nástroje pro generování souborů s výstupy

Pro exporty do souborů ke stažení existuje řada PHP knihoven. Pro tabulkový editor Excel to je např. **PHPExcel** a pro PDF jsou k dispozici např. **FPDF**, **mPDF**, nebo **TCPDF**. Soubory CSV a XML, tedy i SVG lze v PHP generovat jako běžný textový soubor a není proto potřeba žádná další nadstavba.

## 3.5 Programování PHP, kód HTML, SQL

Pro psaní zdrojových kódů lze použít mnoho různě vybavených editorů, nebo vývojových prostředí. Pro psaní PHP se často používá **PHP Storm**, dříve např. **Zend Development Environment**, nebo další editory dle zvyklosti programátora. Z freeware je možno použít např. český textový editor **PSPad**.

## 3.6 Kód CSS

Pro psaní CSS kódu lze opět použít libovolný textový editor, nebo např. speciálně k tomuto účelu dobře vybavený editor **TopStyle** ve freeware verzi Trial. Ze starších editorů lze stále velmi dobře použít také **Macromedia**<sup>12</sup> **Dreamweaver**

## 3.7 Softwarové nástroje použité pro tuto práci

### 3.7.1 Webový a databázový vývojový a testovací server na lokálním PC

Jako aplikační a databázový server pro vývoj a testování aplikace na lokálním PC byly použity WAMP servery na Windows 7 a další software pro správu databáze a debugování zdrojových kódů:

- VertrigoServ 2.43 (Apache 2.4.20, PHP 5.6.20, MySQL 5.7.12, phpMyAdmin 3.5.4, Xdebug 2.4.0) - vývojový a testovací lokální server
- XAMPP 5.6.24 (Apache 2.4.23, PHP 5.6.24, MariaDB 10.1.16, phpMyAdmin 4.5.1, Xdebug 2.2.5 / 2.5.0) - vývojový a testovací lokální server
- HeidiSQL 9.3.0 - správa DB
- Adminer 4.2.4 - správa DB

### 3.7.2 Webový a databázový server v produkčním prostředí

Pro dostupnost aplikace ze sítě internet byla aplikace umístěna a testována na serverech komerčních poskytovatelů webhostingu Pipni s.r.o. a Onesolution s.r.o. Pipni je více jak 15 let poskytovatelem freehostingu pro domény 2. řádu a Onebit (Onesolution s.r.o.) se již přes 15 let drží na předních pozicích měření dostupnosti hostingů českých poskytovatelů těchto služeb. Důležité parametry serverů těchto poskytovatelů webhostingu jsou podobné jako u použitých lokálních serverů (Apache 2.4, PHP 5.6, MySQL 5) akorát na produkčních serverech běží potřebné software na operačních systémech Linux (LAMP servery), což ovšem pro programátora jako uživatele hostingových služeb nehraje žádnou roli.

Již od r. 2016 poskytovatelé hostingu nabízí i hosting s PHP 7, ovšem na PHP 7 zatím nebyla tato aplikace testována. Nepředpokládá se však, že by bylo

---

<sup>12</sup>Dnes je již vlastníkem značky fa Adobe



potřeba zdrojové kódy na novou verzi jazyka serveru výrazně přepracovávat, neboť aplikace nepoužívá funkce, nebo vlastnosti PHP 5.6, které jsou avizovány, že budou ve verzi 7 zrušeny, nebo nahrazeny jinými. Ale v budoucnu by bylo lépe s PHP7 použít novější připojování k databázi - PDO.

### **3.7.3 Programování PHP, kód HTML, SQL, CSS**

- PhpStorm 7.1.3 - psaní zdrojových kódů PHP a debugging (s využitím Xdebug)
- PSPad 4.6 - psaní zdrojových kódů PHP, HTML a SQL příkazů
- Macromedia Dreamweaver - psaní zdrojových kódů HTML a CSS

### **3.7.4 Nástroje pro tvorbu grafiky**

- Adobe Photoshop, Illustrator CS2 - tvorba a úpravy grafiky aplikace a grafiky textu práce
- IrfanView 4.36 - jednoduché úpravy grafiky

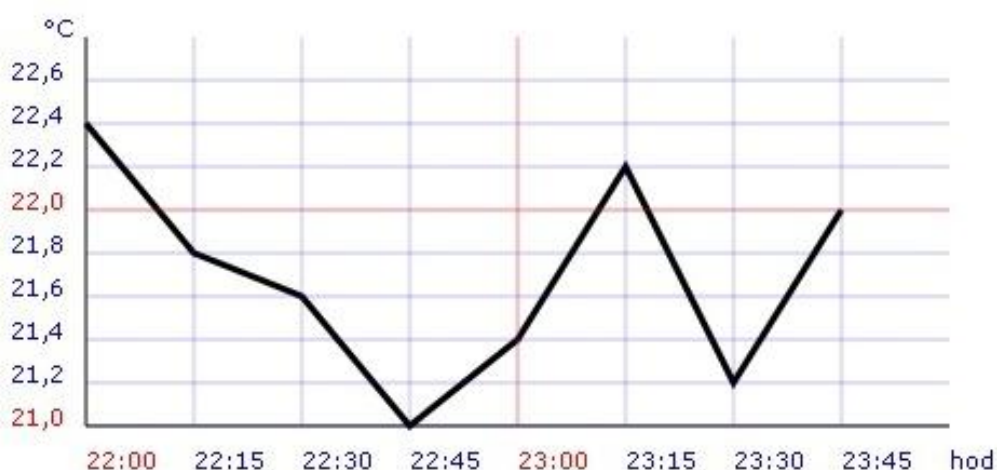
## 4 Návrh realizace zobrazování dat v aplikaci

Z požadavků na aplikaci vyplývá, že data budou zobrazována nejen graficky, ale i textově, nebo je požadováno mít možnost ukládat data jako soubory v různých formátech.

Textová data budou zobrazována v tabulkách, grafické zobrazení bude realizováno grafy. Pro tvorbu grafů lze použít rastrové, nebo vektorové grafiky.

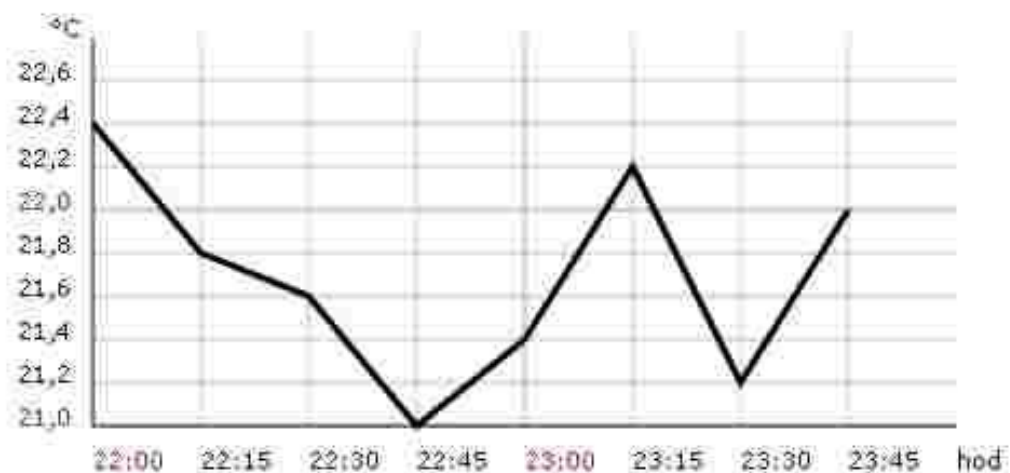
### 4.1 Grafika rastrová

Pro obrázky se na webu používá převážně rastrová grafika. Pro obrázky, které si vystačí s max 256 barvami (animace, grafy, reklamní bannery) lze použít formát GIF, který podporuje také jednoduchou průhlednost, pro fotografie a složitější obrázky je používán většinou formát JPEG. Další rozšířený formát je PNG, který navíc podporuje řízenou průsvitnost. Soubory rastrové grafiky se do HTML kódu stránky vkládají tagem IMG. V HTML5 existuje také možnost 2D vykreslování JavaScriptem do tagu CANVAS, jehož výstupem je rastrový soubor.



Obrázek 5: Použití rastrového obrázku ve formátu JPEG v původním vysokém rozlišení. Zdroj: vlastní řešení

Bylo by možné snížit datovou velikost rastrového obrázku, ale jelikož JPEG je formát se ztrátovou kompresí, na 5 kB by nebylo možné se dostat bez drastického úbytku kvality obrázku. Při kvalitě uložení 10% má JPG 4,7 kB, ale kvalitou obrazu je nepoužitelný.



Obrázek 6: Použití rastrového obrázku ve formátu JPEG v nízkém rozlišení po kompresi ukazuje ztrátu kvality obrazu. Zdroj: vlastní řešení

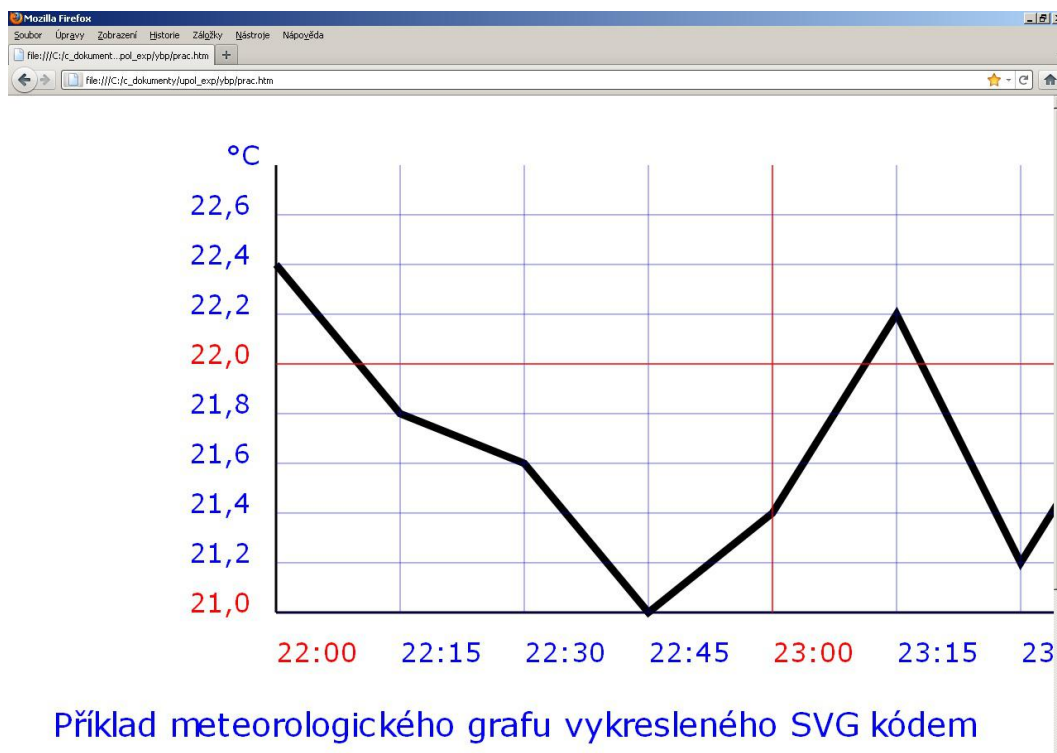
## 4.2 Grafika vektorová

Vektorová grafika je trendem poslední doby a její použití roste. Dříve byl problém se zobrazováním SVG hlavně v prohlížečích Internet Explorer verze 8 a starších, což možná bylo důvodem, proč nebylo SVG na webu příliš rozšířené v dobách jeho masivního rozšíření. SVG je dnes často používáno od různých drobných ikon až po složitější animace.

Formát SVG byl původně navržen primárně pro web (první standard vydán v r. 2001) a nyní je nejpoužívanějšími webovými prohlížeči mladšími 5 let již dobře podporován. Mimo jiné také služba Google chart je na SVG postavena, takže je pravděpodobné, že postupem doby bude možné starší prohlížeče zcela pominout a SVG se stane běžně používaným formátem. Soubory SVG grafiky se do HTML kódu stránky vkládají tagem IMG jako ostatní obrázky, nebo tagem OBJECT.

Pro potřeby této práce, tedy na vykreslování různých grafů je formát SVG velmi vhodný, neboť je to formát na bázi XML a je přiměřeně snadné takovouto grafiku programově generovat. Vektorová grafika lze z principu v prohlížeči zvětšovat bez ztráty kvality obrazu (viz obr 2), přitom dále v textu (viz obr 3), uvedený kód má jako SVG soubor pouhých 4,2 kB a stejný graf uložený jako JPG v kvalitě uložení 95% má 28,7 kB

Je třeba ovšem také dodat, že při použití SVG mohou nastat problémy např. při požadavku na vložení obrázku do DTP programu, nebo do starších verzí kancelářských balíků Microsoft Office a OpenOffice.org na PC uživatele. Aktuální verze kancelářských balíků však SVG již dobře podporují a třeba Adobe Illustrator minimálně od verze CS2 dokáže SVG bez problému načíst, editovat a následně i uložit do PDF, nebo jiných formátů.



Příklad meteorologického grafu vykresleného SVG kódem

Obrázek 7: Ukázka bezztrátové kvality SVG grafiky - max zvětšení. Zdroj: vlastní řešení

### 4.3 Přehled webového software pro generování grafů

Samotné vykreslování grafů ve webovém prohlížeči není úplně triviální záležitostí. Dá se říci, že technologie, na webu používané, jak na straně serveru (PHP), tak i na straně klienta (HTML5 CANVAS a JavaScript) nebo jejich kombinace umožňují generovat grafy jak statické (vygenerované obrázky), tak dynamické měnící se v reálném čase animované JavaScriptem. Ale žádná z technologií nenabízí samostatné speciální uživatelské funkce např. pro tvorbu grafů.

V současné době je na internetu dostupných více hotových řešení. Zřejmě nejznámější je služba Google Chart API <sup>[13]</sup>, která nabízí mnoho typů grafů. Ve stabilní verzi je tato služba k dispozici od r. 2010. Od r. 2007, kdy byla služba spuštěna nabízela generování grafů v rastrovém formátu (PNG), v dnešní době jsou grafy vykreslovány pomocí SVG pro novější prohlížeče, případně VML pro starší Internet Explorer. Grafy jsou také vykreslovány s dynamickými JavaScriptovými doplňky, popisky, různými dalšími animačními prvky, nebo mohou být celé navrženy jako animované. Dostupné jsou ale stále i API pro generování rastrových grafů <sup>[14]</sup>. Dalším nástrojem sloužícím k implementaci do vlastních řešení je např. JavaScriptová knihovna Flot <sup>[15]</sup> pro jQuery.

## 4.4 Vlastní řešení zobrazování dat

Použití výše uvedené služby Google Chart API by požadavek na grafy webového rozhraní v této práci řešilo zřejmě dobře a je to jedna z možných cest. Dále v textu je také uveden příklad využití Google Chart API v jiné závěrečné práci. Pro tuto práci je však použito vlastního řešení na stejném principu. Graf je vykreslován do SVG souboru. Vzhledem k tomu, že dnešní prohlížeče již SVG vykreslují dobře, jeví se toto použití vektorové grafiky jako vhodné. Vlastní řešení také umožní vytvářet grafy více podle specifických potřeb uživatele, než by třeba bylo možno dosáhnout standardizovaným grafem z nabídky Google.

## 4.5 Realizace zobrazování dat

Data uložená v databázi lze prezentovat jako:

1. textová, nebo binární negrafická data - data jsou určena pro další strojové zpracování, nebo pro uživatelské zpracování v tabulkových editorech typu Microsoft Excel, nebo OpenOffice Calc.
  - textové soubory pro další uživatelské, nebo strojové zpracování bez formátování, např. formáty TXT, CSV
  - textové soubory pro další uživatelské, nebo strojové zpracování soubory s daty formátovanými pomocí značkovacích jazyků s požadovanou strukturou - např. (X)HTML a CSS, XML, XML a XSLT
  - binární soubory pro další převážně uživatelské zpracování, např. XLS(X), PDF
2. grafická data - rastrové, nebo vektorové formáty souborů generované jednorázově
  - graf vygenerovaný do rastrového souboru grafiky formátu JPEG / GIF / PNG
  - graf vykreslovaný v prohlížeči kombinací HTML/XHTML a CSS, případně i s pomocí JavaScriptu
  - graf vygenerovaný do vektorového souboru grafiky formátu SVG

PHP v základní výbavě obsahuje knihovnu pro generování rastrových obrázků obsahující několik, dnes již v porovnání s pokročilými funkcemi grafických editorů, základních funkcí. Pro generování grafů neobsahuje žádné speciální nástroje, vše potřebné je potřeba napsat uživatelsky. Taktéž SVG soubory je potřeba generovat uživatelskými skripty jako jiné XML soubory.

Pro tuto práci je použito generování CSV souborů tabulek, generování souborů SVG grafů a generování PDF s vloženým SVG souborem grafu. Všechny tyto exporty tak jdou uložit na PC uživatele.

Generování probíhá pomocí PHP na serveru, CSV a SVG jsou generovány jako běžné textové soubory a pro generování PDF je použita PHP knihovna mPDF.

## 5 Návrh funkčnosti aplikace

Ze zadání vyplývá, že zobrazování dat bude vhodné realizovat několika způsoby. Jelikož se má jednat o aplikaci určenou převážně vědeckým pracovníkům, nebo odborníkům v určitém konkrétním oboru, předpokládá se, že pro toto použití jsou potřebná následující kritéria:

- V případě přímého zobrazení na webu mají být informace jasné, rychle srozumitelné, přehledné bez nadměrně rušivých grafických efektů a nezaměnitelné. Řešení s bohatou efektní grafikou jsou vhodná spíše pro laické uživatele, kteří např. nepotřebují s daty/grafy dále pracovat. Uživatelé dle zadání budou zřejmě preferovat spíše přesné a detailní informace a „tvrdá“ data, než jejich efektní graficky bohatou prezentaci.
- Nejedná se o zobrazování dat pro koncové laické uživatele, ale o zobrazování dat, které je nutno koncipovat jako dále použitelný polotovár pro další strojové, nebo ruční uživatelské zpracování, odborníci jsou spíše v situaci, kdy každá data mohou být pouze podkladem pro nějaké další zpracování, data z webové aplikace nemusí být finálním výstupem práce odborníka, webová aplikace může být jen pracovním nástrojem, nebo prostředkem.
- Předpokládaným častým použitím by mohlo být vygenerování grafu on-line (např. rastrový formát JPEG, nebo vektorový SVG) a jeho ruční kopírování do kancelářské, nebo DTP aplikace/dokumentu off-line, např. Microsoft Word, OpenOffice Writer, nebo další.
- Požadavkem je, aby aplikace mohla zobrazovat nejen surová data ukládaná ze senzorů do databáze, ale i provádět s nimi další výpočty, vytvářet různé sestavy z databáze, generovat souborové výstupy v různých formátech, ať už grafické, nebo textové.
- Webová aplikace by měla být zobrazitelná a funkční v co největší možné množině zařízení. Z tohoto titulu není vhodné stavět aplikaci na málo rozšířených technologiích, na proprietárních technologiích, kterými některá zařízení vůbec nedisponují, nebo v dohledné době disponovat nebudou, zobrazení by mělo být spolehlivé a založeno pokud možno na jednoduchém principu.
- Pro zásadní a důležité grafické prvky bude lépe použít technologie, které lze označit jako dlouhodobě prověřené, spolehlivé a široce rozšířené.
- V případě generování grafů a obrázků lze označit jako spolehlivé generování na serveru pomocí PHP. Bylo možno také použít HTML a CSS sloupcové grafy, ale toto řešení by bylo dále těžko přenositelné, např. Pro ukládání grafu jako souboru.

- Jako podmíněně spolehlivé technologie lze označit JavaScript, Ajax, Flash animace a některé (ne všemi prohlížeči ještě úplně podporované) prvky HTML5. Tyto technologie by měly být použity jen jako doplňující ke stabilně spolehlivým, např. pro zvýšení uživatelského komfortu, nebo k vhodným dynamickým doplňkům. Nikoli však k stěžejní funkcionalitě. JavaScriptová řešení jsou prakticky nepoužitelná jinak dále (bez další softwarové konverze), než jen k zobrazení v prohlížeči. Nelze je nijak dále uchopit, exportovat jinak než jako obrázek sejmutím obrazovky, který musí být dále ručně zpracován. Existují ovšem samozřejmě aplikace, které bez klientského skriptování nejsou jiným způsobem vůbec proveditelné. Charakteristickým zástupcem takovýchto webových aplikací s vysokou uživatelskou hodnotou je např. zobrazování mapových dat. Pak je ovšem potřeba mít k dispozici vhodný moderní prohlížeč a mít na paměti, že se k těmto datům např. ze staršího mobilního telefonu nebude možné dostat.
- Jako nevhodné kvůli malému rozšíření, závislosti na platformě, nebo nízké zaručené spolehlivosti správné interpretace na široké základně prohlížečů lze zatím označit VB Script, Silverlight, WebGL, případně některé další inovativní, avšak ještě ne hojně rozšířené technologie.
- Příkladem možných problémů s technologií zobrazování grafiky může být výše v textu uvedené prohlášení fy Apple, že její produkty nikdy nebudou podporovat technologii Adobe Flash.



## 6 Popis aplikace

Jak je popsáno již v jiných částech tohoto textu, aplikace je navržena jako webová aplikace s použitím databáze MySQL, serverového jazyka PHP, HTML a CSS kódování. Skriptování na straně klienta zde není zatím použito.

### 6.1 Výčet stran aplikace

**Aplikace obsahuje** 3 funkční a 3 pomocné strany a 1 informativní stranu, dostupné z menu aplikace. Pomocné strany vznikly během vývoje aplikace jako podpůrné skripty pro vývoj aplikace. Jedná se o strany, které sloužily pro částečné strojové opravy poškozených surových dat z měření tak, aby bylo možno data načíst do databáze ve formátu a struktuře vhodné pro běh aplikace. Mimo tyto 3 pomocné strany dostupné z menu webu vzniklo ještě několik samostatných pracovních a pomocných PHP skriptů pro opravy dat bez grafického rozhraní, které nejsou dostupné z menu aplikace.

#### 6.1.1 Mapa rozmístění uzlů

Strana **Mapa rozmístění uzlů** je úvodní strana aplikace. Obsahuje statickou mapu plánovaného rozmístění sensorových uzlů v zájmové oblasti. Mapa zobrazuje plánovaných 13 uzlů označených 1 - 13 a bránu označenou jako 0. Síť uzlů v této konfiguraci však nebyla v době psaní této práce k dispozici. Tato aplikace využívá dat z experimentálních měření v dubnu 2014, kdy byly použity pouze 4 sensorové uzly, označené pořadovým číslem a (zřejmě) výrobním číslem, nebo interním ID výrobce uzlu:

Tabulka 5: Čísla a ID použitých uzlů

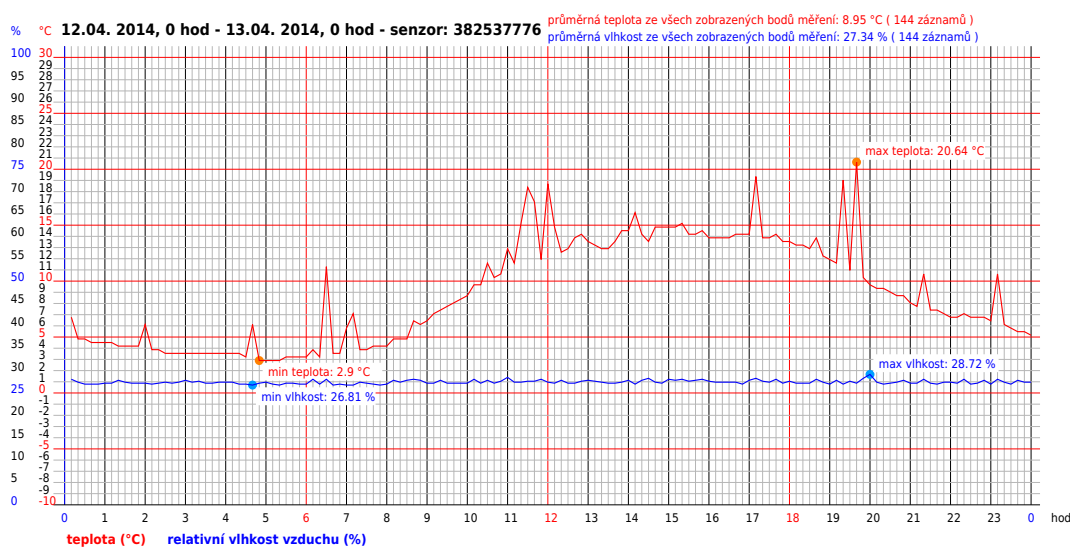
číslo uzlu	ID uzlu
uzel č.0	382537776
uzel č.1	382551615
uzel č.2	382552437
uzel č.3	382554362

## 6.1.2 Grafy měření

Strana **Grafy měření** umožňuje vykreslit graf z hodnot měření za 1, 3, 6, 12, 24, nebo 48 hodin od vybrané hodiny vybraného dne. V grafu se také mimo křivky veličin automaticky zobrazují body s minimální a maximální hodnotou vybrané veličiny a popisem a v čase, kdy toto minimum a maximum nastalo. Dále také výpisy za jaké dny a hodiny měření je a také výpočty průměrných hodnot veličin ze všech zobrazených hodnot měření.

Z důvodů uvedených na jiných místech této práce umožňuje zatím strana vykreslení dat ze 2 dnů měření a 1 sensorového uzlu, což lze vybrat z rozbalovacích nabídek stejně jako hodina začátku vykreslování. Pomocí checkboxů lze vybrat, zda bude v grafu vykreslena křivka teploty a/nebo vlhkosti. Zároveň s tím je možno pomocí dalšího checkboxu volitelně nechat pod grafem vypsat tabulky měření s vybranými veličinami.

Dále strana umožňuje uložit samotný soubor grafu ve formátu SVG, nebo tento SVG soubor vložit do generovaného PDF ke stažení a dále také uložit vypsané tabulky měření do souboru CSV pro další editaci v tabulkovém editoru.



Obrázek 8: Ukázka grafu měření vygenerovaného aplikací, strana Grafy měření. zdroj: Webové rozhraní pro zobrazování sensorových dat ze sensorové sítě, která je předmětem této práce.

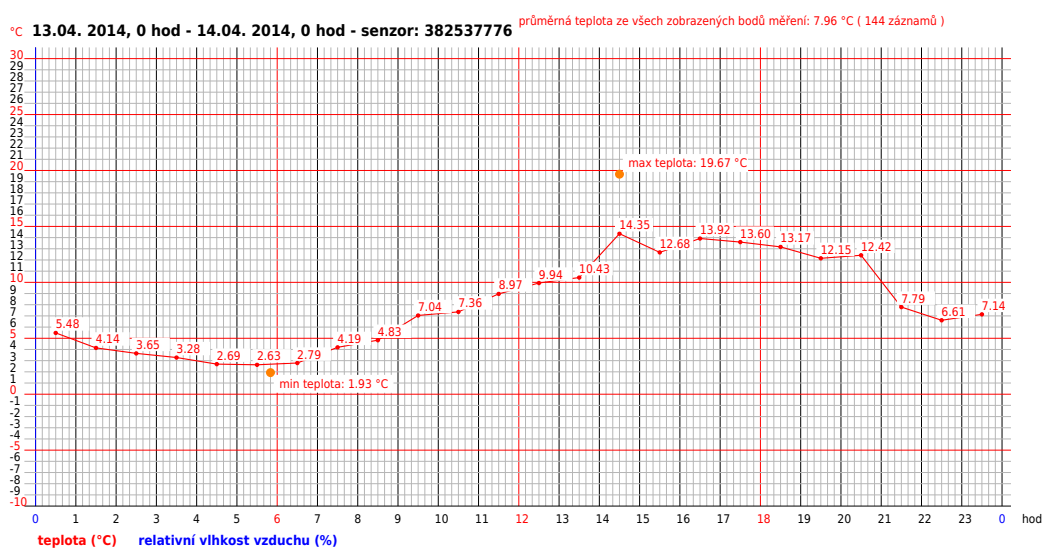
### 6.1.3 Výpočty hodnot

Strana **Výpočty hodnot** umožňuje vypočítávat meteorologické hodnoty z dat měření za 1 vybraný den a senzor a vykreslit graf z vypočítaných hodnot. Vypočítávané meteorologické hodnoty a jejich vzorce jsou uvedeny výše v kapitole s požadavky na aplikaci. Omezení výběru dnů a uzlů je stejné jako na předešlé straně.

Výchozím nastavením pro vykreslování grafu je vykreslení bodů průměrných hodnot. Pomocí checkboxů lze vybrat, zda bude v grafu vykreslena křivka teploty a/nebo vlhkosti, zda bude vůbec vykreslena křivka spojující body. Dále lze nastavit výpis vykreslení číselných popisů hodnot veličin k zobrazeným bodům a zda budou vykresleny body s minimální a maximální hodnotou vybrané veličiny a popisem a v čase, kdy toto minimum a maximum nastalo.

Dole pod grafem se pak automaticky vypisují tabulky výsledků výpočtů k vybraným veličinám.

Strana také umožňuje uložit samotný soubor grafu ve formátu SVG, nebo tento SVG soubor vložit do generovaného PDF ke stažení a dále také uložit vypsané tabulky výpočtů do souboru CSV pro další editaci v tabulkovém editoru.



Obrázek 9: Ukázka grafu měření vygenerovaného aplikací, strana Výpočty hodnot. zdroj: Aplikace Webové rozhraní pro zobrazování sensorových dat ze sensorové sítě, která je předmětem této práce.

#### 6.1.4 Počty měření

Skript informativní strany **Počty měření** generuje seznam všech dostupných měření uložených v databázi za všechny měřené dny z experimentálního měření v dubnu 2014, již po předešlé opravě samostatným PHP skriptem. Původní surová data z měření 3. - 17. 4. 2014 neobsahují ani v jednom měření potřebných 144 záznamů/den (24h x 6 zázrn/hod), tedy záznamy buď nejsou vůbec, nebo byly skriptem **Opravy souborů dat** vyřazeny jako neúplné. Je však k dispozici více (celkem 6) měření po 143 záznamech/den, tak bylo pro senzor č.0 - 382537776 ve dnech 12. - 15. 4. 2014 u každého dne doplněn potřebný počet záznamů, aby byla sada dat kompletní aspoň pro 4 dny. Data vypsaná skriptem **Počty měření** by byla použitelná v aplikaci webového rozhraní, pokud by proběhla další oprava viz skript **Přiřazení id měření**.

#### 6.1.5 Opravy souborů dat

Jelikož byla všechna surová data měření poškozená, bylo nutno se aspoň pokusit některé hodnoty strojově opravit pro možnost importovat data do databáze v přiměřeně konzistentním stavu. Skript **Opravy souborů dat** vygeneruje z načteného souboru dat nový CSV soubor, který k původním datům přidá sloupce s opravovanými daty a různými příznaky, podle kterých je možno data filtrovat. Vygenerovaný soubor žádné (neprázdné) záznamy nemaže, jen je přeskupí do podoby vhodné pro pozdější import do databáze.

#### 6.1.6 Přiřazení ID měření

Skript **Přiřazení ID měření** je přípravou na další fázi oprav dat měření. Tyto opravy nebyly dokončeny. Skript vezme neúplné záznamy bez časové návaznosti a doplní 0 na místa bez záznamu tak, že výsledkem je vždy 144 záznamů na den a data jsou správně rozmístěna vzhledem ke svému času. Skript umí seřadit (a doplnit 0) i libovolně nesetříděné záznamy do formátu 6 záznamů/hod x 24h = 144 celkem na 1 den a senzor.

#### 6.1.7 Parametry sítě

Strana **Parametry sítě** je informativní stranou o plánovaných parametrech a cílech projektu.

## 6.2 Data a databáze

Databáze sice není předmětem této práce, ale pro zobrazování dat je nutným požadavkem mít data uložená v nějaké databázi. Proto bylo potřeba nějakou databázi na nějakém serveru zřídit a jak je zmíněno obecně v jiné části tohoto textu, pro tuto práci je použita databáze MySQL, jako součást služeb provozovatele webhostingu pro prezentaci aplikace.

Pro tuto práci jsou použita data získaná experimentálním měřením v dubnu 2014, kde se měřily 2 veličiny - teplota a vlhkost vzduchu. Na základě dodaného excelovského souboru s daty byla navržena 1 tabulka databáze tak, aby kopírovala strukturu dat v excelovském souboru. To z důvodu, že byl předpoklad, že takto budou data vypadat i nadále a bude pak snazší data do databáze strojově načítat.

V založené MySQL databázi byla vytvořena tabulka DBS se sloupci viz SQL kód níže.

```
1 CREATE TABLE `dbs` (  
2   `id` INT(10) NOT NULL,  
3   `datum` DATE NOT NULL,  
4   `cas` TIME NOT NULL,  
5   `sync` VARCHAR(3) NOT NULL,  
6   `i` INT(10) NOT NULL,  
7   `idm_dne` INT(10) NOT NULL,  
8   `n` INT(10) NOT NULL,  
9   `ca` FLOAT NOT NULL,  
10  `cb` FLOAT NOT NULL,  
11  PRIMARY KEY (`id`)  
12 )  
13 COLLATE='utf8_general_ci'  
14 ENGINE=InnoDB;
```

Zdrojový kód 2: SQL kód vytvoření tabulky DBS

Už při letmém pohledu na data uložená v XLS bylo jasné, že data nejen že jsou na bráně Meshlium ukládána nevhodně, ale obsahují i chyby. Takto uložená data obsahují zbytečné sloupce pouze s hodnotou NULL, nebo 0, čísla jsou uložena jako řetězce v nepotřebných uvozovkách a všechna zásadní data obsahující měřené veličiny, číslo senzoru, a např. pro tuto práci nepotřebný stav nabití baterie, jsou uložena v jednom sloupci, čímž nespĺňují ani 1. normální databázovou formu. Z toho mimo jiné vyplývá, že pokud jsou data prvotně uložena na bráně, je otázkou, do jaké míry lze ovlivňovat jak bude vypadat tabulka MySQL databáze v bráně. Pak před uložením do databáze na serveru by musela stejně proběhnout transformace dat do jakékoli podoby vhodné pro zobrazení v aplikaci, respektive do struktury tabulek databáze na serveru.

Data z měření byla ukládána v bráně a odtud ve formátu CSV exportována a následně uložena v XLS. Tímto převodem m.j. následně vznikly některé chyby

uváděné níže. Bylo tedy potřeba do dat zasáhnout napřed ručně a pak i strojově. Soubor XLS byl ručně uložen jako CSV, aby s ním šlo dále pracovat strojově a byly provedeny elementární transformace, které bylo rychlejší provést ručně v kancelářském balíku OpenOffice. Postupem doby bylo napsáno několik podpůrných PHP skriptů, které pomocí regulárních výrazů transformovaly data do potřebného výsledku.

Data vykazovala několik různých typů chyb, které je v podstatě možno rozdělit na 2 základní druhy. Chyby měření senzorů a chyby uložení dat na bráně Meshlium.

Zřejmě již na senzorech docházelo k fatálním chybám v měření. Zatímco teploty vycházely na duben očekávané, vlhkosti vzduchu se pohybovaly kolem 127%, což je evidentně hodnota vadná.

**Z chyb které mohly vzniknout chybným ukládáním na bráně se pak jedná o chyby:**

- Rozdělení 1 záznamu na 2 - neúplné záznamy a úplně vadné hodnoty.
- 1 rozdělený záznam na 2 - data jednoho záznamu jsou v libovolném místě rozdělena a zbytek se zapsal do dalšího řádku. Toto jsou data, z kterých by za určitých okolností šla data zrekonstruovat, ale bylo by to poměrně pracné. Jelikož záznamy byly rozdělené libovolně kdekoliv v záznamu, byl by problém podle čeho data zpět spojovat. Pak po spojení dat do jednoho řádku by bylo ještě nutno smazat řádek s neúplným zbytkem záznamu.
- Neúplné záznamy - záznamy, které byly rozdělené, ale zbytek záznamu se ani neuložil do dalšího řádku.
- Úplně vadné hodnoty - v některých záznamech se vyskytují např. startovní hlášení brány „Hello, this is Waspnote Plug & Sense“, nebo rozdělené řádky kde Excel z několika číslic sestavil datum a převedl ho třeba i na římské číslice.

## 6.3 Zdrojové kódy aplikace

Vlastní zdrojový kód aplikace v jazyce PHP je z převážné části napsán procedurálně. Aplikace je rozdělena do několika souborů, podle funkčnosti a do několika adresářů, viz tabulky níže. Jediným vlastním objektovým kódem je jednoduchá knihovna pro generování odkazů a menu - web\_atoms.php.

Tabulka 6: Popis adresářů aplikace

název adresáře	popis
3_01drv	obsahuje soubory menu.html a umwa.php
3_05bck	obsahuje soubory s grafikou aplikace
3_06dwn	obsahuje PDF soubory popisující zadání a senzory
5_01css	obsahuje CSS soubory web_atoms.css, menu.css, menu2.css
grafy	adresář pro ukládání vygenerovaných SVG souborů grafů
mpdf60	adresář s knihovnou mPDF
soubory	adresář pro ukládání vygenerovaných CSV a PDF souborů

Tabulka 7: Popis HTML souborů aplikace

název souboru	popis
404.htm	obsah strany 404
x03.html	obsah strany PARAMETRY SÍTĚ
x03_a.html	pravý sloupec strany PARAMETRY SÍTĚ
x22_popis.htm	obsah popisu strany x22

Tabulka 8: Popis PHP souborů aplikace

název souboru	popis
404.php	skript strany 404
dbconn.php	soubor s nastavením pro připojení k databázi
fce1.php	1. soubor s funkcemi načítanými do skriptů
fce2.php	2. soubor s funkcemi načítanými do skriptů
fce3.php	3. soubor s funkcemi načítanými do skriptů
fce4.php	4. soubor s funkcemi načítanými do skriptů
index.php	soubor hlavní strany webu s přesměrováním
stroj.php	hlavní vykreslovací engine, do kterého se načítají obsahy podstran
web_atoms.php	jednoduchá knihovna pro generování menu a odkazů
x00.php	strana GRAFY MĚŘENÍ
x02.php	strana MAPA ROZMÍSTĚNÍ UZLŮ
x11.php	skript vykreslování grafů měření
x11_form.php	vstupní formulář skriptu vykreslování grafů měření
x13.php	skript pro generování PDF
x21.php	strana grafů výpočtů hodnot
x21_form.php	vstupní formulář skriptu vykreslování grafů výpočtů hodnot
x21_graf.php	skript vykreslování grafů výpočtů hodnot
x21_tab.php	skript výpisů a generování CSV tabulek
x21_vyp.php	skript výpisů hodnot
x22.php	strana OPRAVY SOUBORŮ DAT
x22w1.php	1. část skriptu strany OPRAVY SOUBORŮ DAT
x22w2.php	2. část skriptu strany OPRAVY SOUBORŮ DAT
x22w3.php	3. část skriptu strany OPRAVY SOUBORŮ DAT
x24.php	strana POČTY MĚŘENÍ
x27.php	strana PŘÍRAZENÍ ID MĚŘENÍ
x27_form.php	vstupní formulář strany PŘÍRAZENÍ ID MĚŘENÍ



## 6.4 Ukázka SVG kódu grafu část 1

---

```
<!-- hlavicka souboru -->
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/
  Graphics/SVG/1.1/DTD/svg11.dtd">

<!-- startovaci tag souboru -->
<svg width="600px" height="400px" version="1.1" xmlns="http://www.w3
  .org/2000/svg">

<!-- popis souboru grafu -->
  <desc>Příklad SVG kodu meteorologickeho grafu</desc>

<!-- vykresleni krivky grafu -->
  <path d="M100 60 L150 120 200 140 250 200 300 160 350 80 400 180
    450 100" fill="none" stroke="black" stroke-width="3"/>

<!-- vykresleni cary svisle stupnice grafu -->
  <line x1="100" y1="200" x2="500" y2="200" fill="none" stroke="black
    " stroke-width="1"/>

<!-- vykresleni cary vodorovne stupnice grafu -->
  <line x1="100" y1="200" x2="100" y2="20" fill="none" stroke="black"
    stroke-width="1"/>

<!-- vykresleni popisu veliciny grafu -->
  <text x="80px" y="20px" font-size="8pt" fill="blue">st.C</
    text>

<!-- vypsani hodnot svisle stupnice grafu -->
  <text x="65px" y="40px" font-size="8pt" fill="blue">22,6</text
    >
  <text x="65px" y="60px" font-size="8pt" fill="blue">22,4</text
    >
<!-- ... atd -->

<!-- vypsani hodnot vodorovne stupnice grafu -->
  <text x="100px" y="220px" font-family="Verdana" font-size="8pt
    " fill="red">22:00</text>
  <text x="150px" y="220px" font-family="Verdana" font-size="8pt
    " fill="blue">22:15</text>
  <text x="500px" y="220px" font-family="Verdana" font-size="8pt
    " fill="blue">hod</text>
<!-- ... atd -->
```

---

Zdrojový kód 3: SVG kód grafu 1

## 6.5 Ukázka SVG kódu grafu část 2

---

```
<!-- vykresleni svislych car stupnice grafu -->
<line x1="150" y1="200" x2="150" y2="20" fill="none" stroke="blue"
stroke-width="0.2"/>
<line x1="200" y1="200" x2="200" y2="20" fill="none" stroke="blue"
stroke-width="0.2"/>
<line x1="250" y1="200" x2="250" y2="20" fill="none" stroke="blue"
stroke-width="0.2"/>
<line x1="300" y1="200" x2="300" y2="20" fill="none" stroke="red"
stroke-width="0.5"/>
\textcolor{blue}{... atd}

<!-- vykresleni vodorovnych car stupnice grafu -->
<line x1="100" y1="40" x2="500" y2="40" fill="none" stroke="blue"
stroke-width="0.2"/>
<line x1="100" y1="60" x2="500" y2="60" fill="none" stroke="blue"
stroke-width="0.2"/>
<line x1="100" y1="80" x2="500" y2="80" fill="none" stroke="blue"
stroke-width="0.2"/>
<line x1="100" y1="100" x2="500" y2="100" fill="none" stroke="red"
stroke-width="0.5"/>
<!-- ... atd}

<!-- vypsani popisneho textu grafu -->
<text x="10px" y="250px" font-family="Verdana" font-size="10pt"
fill="blue">
  Příklad meteorologickeho grafu vykresleneho SVG kodem
</text>

<!-- uzaviraci tag souboru -->
</svg>
```

---

Zdrojový kód 4: SVG kód grafu 2

## 7 Cíle práce

Tato bakalářská práce byla vypsána jako jedna z prací na podporu disertační práce *Experimental wireless sensor network in Olomouc* Mgr. Hejlové z katedry Geoinformatiky UP Olomouc. Cílem této a některých dalších podpůrných prací bylo vytvořit zázemí pro realizaci komplikované a rozsáhlé problematiky fungování bezdrátových sensorových sítí. Na Katedře informatiky byly k tomuto tematu vypsány celkem 3 práce:

1. *Sběr dat v sensorové síti*
2. *Databáze pro ukládání dat naměřených v sensorové síti*
3. a tato práce - *Webové rozhraní pro zobrazování sensorových dat*

Práce *Sběr dat v sensorové síti* a *Databáze pro ukládání dat naměřených v sensorové síti* zatím nebyly realizovány. Proto se tato aplikace nepřipojuje k databázi z práce č.2 a nepoužívá data online sbíraná v práci č.1, ale využívá sadu dat z experimentálních měření. Jak je zmíněno v jiné části tohoto textu, data z experimentálních měření byla všechna poškozená a nekompletní a bylo je nutno před přesunem do databáze ručně a strojově opravit.

Také na Katedře geoinformatiky vzniklo k tématu sensorových sítí více prací např.:

- **JIRÁSEK, Ondřej.** *Senzorové sítě v environmentálních studiích.* 2013. Katedra geoinformatiky UP Olomouc
- **DOSTÁL, Radek.** *Měření základních meteorologických prvků pomocí bezdrátových sensorových sítí.* 2015. Katedra geoinformatiky UP Olomouc
- **KONÍČEK, Jakub.** *Optimalizácia energetického zaťaženia uzla wasp-mote plug & sense s ohľadom na vybrané faktory.* 2016. Katedra geoinformatiky UP Olomouc
- **BALUN, Marek.** *Analytické zpracování dat v prostředí sensorového webu.* 2016. Katedra geoinformatiky UP Olomouc
- **FEDRZEL, Filip.** *Párování dat DPZ se sensorovými daty pro oblast precizního zemědělství.* 2016. Katedra geoinformatiky UP Olomouc

## 8 Porovnání s podobnými pracemi

Podobným problémem jako tato práce se částečně zabývá práce M. Baluna *Analytické zpracování dat v prostředí sensorového webu*, která se také zabývá m.j. i webovým rozhraním pro výpis sensorových dat do tabulek a vykreslováním grafů. Autor jde při zobrazování grafů ovšem jinou cestou, využívá službu společnosti Google <sup>[13]</sup>.

Grafy Google jsou vytvářeny pomocí Google Chart API. O této cestě pro tuto práci bylo také uvažováno, ale jako možný problém se jevílo, že je závislá na cizí službě a také je plně závislá na JavaScriptu. Další nevýhodou by mohlo být, že grafy vykreslené pomocí Google Chart jdou uživatelsky ukládat jako soubory poměrně složitě. Pokud by uživatel chtěl grafy ukládat u sebe v PC pro případné další použití. Lze uložit celou webovou stránku s grafem a HTML kód stránky obsahuje vložený SVG kód, který lze ručně vypreparovat a udělat z něj SVG soubor. Další nevýhodou by mohlo být, že lze použít jen předdefinované prvky a zobrazení grafů je tedy uživatelsky těžko přizpůsobitelné, pokud by případný uživatel chtěl diametrálně odlišný způsob zobrazování grafů.

Výhodou na druhou stranu pak je, že se používá hotového řešení vyvinutého stabilní firmou se silným zázemím a grafy mají implementováno dynamické zobrazování hodnot při najetí myši nad uzlem grafu. Což je jinak při generování pouhého SVG souboru grafu nutno řešit samostatnou JavaScriptovou obsluhou jako práce navíc.

Práce F. Fedrzela *Párování dat DPZ se sensorovými daty pro oblast precizního zemědělství* zase např. řeší opravy chybových dat z uzlů, ovšem jde cestou skriptů v jazyce Python s použitím cca 6-ti nativně implementovaných knihoven pro různé standardizované účely. Jelikož v této práci je jako skriptovací jazyk na serveru použito PHP, které takové knihovny nemá, tato práce používá pro opravy dat regulárních výrazů. Tento přístup přináší širší možnosti práce s daty, zachovává původní soubor s daty nezměněn pro případnou archivaci zdrojového souboru a data pro import do DB připravuje výběrem potřebných údajů z původního souboru a vytvořením nového souboru s opravenými daty.

## Závěr

Výsledkem práce *Webové rozhraní pro zobrazování sensorových dat ze sensorové sítě* je webová aplikace, která umožňuje data různými požadovanými způsoby zobrazovat, dopočítávat z nich další hodnoty a ukládat zpracovaná data do souborů. Aplikace umožňuje zobrazovat výstupy z dat měření v grafech a v tabulkách, lze vytvářet různé sestavy jak časové, tak veličinové, lze ukládat tabulky dat do souborů formátů CSV a lze generovat a ukládat grafy do formátů SVG a PDF.

## Conclusions

The result of the work Web interface for displaying sensor data from sensor network is a web application that allows the data to be displayed in various desired ways, counting other values and storing the processed data into the files. The application allows display outputs from measurement data in charts and tables, create different time and value assemblies, save datasets to CSV files and generate and save charts to SVG and PDF formats.

## A Příloha 1 - zprovoznění webové aplikace na serveru (readme.txt)

- Pro provoz webové aplikace je třeba mít k dispozici server se softwarovým vybavením podrobněji popsáním v jiných částech tohoto textu. V dnešní době prakticky všechny nabídky služeb komerčních subjektů poskytujících webhosting jsou pro provoz této aplikace dostačující a není potřeba ani nic speciálně nastavovat. Až v případném ostrém produkčním provozu by vplynuly nějaké další konkrétní požadavky, které zatím při experimentálních měřeních nejsou známy.
- Pokud by byl požadavek provozovat aplikaci na vlastním serveru, je potřeba mít server s konfigurací softwaru posledních verzí, tedy nejlépe Apache 2.4+, PHP 5.6+ a MySQL 5.5+, na vhodném operačním systému, nejběžněji na Linuxu.
- pro připojení aplikace k databázi je potřeba v souboru *dbconn.php* nastavit aktuální hodnoty konstant, které toto připojení zajišťují:
  - SQL\_HOST - adresa databázového serveru (u komerčních provozovatelů často ve tvaru 127.0.0.1, neboť z hlediska webhostingu se jedná o připojení k vlastní lokální databázi uživatele)
  - SQL\_DBNAME - jméno databáze na serveru
  - SQL\_USERNAME - název uživatele
  - SQL\_PASSWORD - heslo pro připojení k databázi
- Je potřeba založit databázi, toto se provádí v administračním prostředí webhostingu u jednotlivých provozovatelů dle jejich administrace, v případě provozu aplikace např. na serveru UPOL dle pokynů správce serveru.
- Ve vytvořené databázi je také potřeba vytvořit potřebné tabulky a naplnit je daty. Toto je možno udělat importem přiloženého souboru *feed.sql*, který obsahuje SQL jak příkazy pro kompletní vytvoření a nastavení tabulek, tak i již transformovaná data z experimentálních měření.
- Poté by již aplikace měla být funkční z adresy dle adresářové struktury vytvořené na serveru, většinou spouštěné z adresy *url/cesta/index.php*. Jelikož v případě této aplikace je pro některé vybrané strany webu jako alternativa k plné adrese s GET parametry použito funkčnosti *mod\_rewrite* modulu HTTP serveru Apache pro přepisování adres, stačí zadat adresu pouze ve tvaru *url/cesta/ybp*.

Aplikace k prezentaci se nachází na adrese:

<http://www.webylevne.cz/ybp>

## B Obsah příloženého CD/DVD

Příložené CD/DVD obsahuje adresáře a soubory:

### **bin/**

Kompletní adresářová struktura webové aplikace WEBOVÉ ROZHRANÍ PRO ZOBRAZOVÁNÍ SENZOROVÝCH DAT ZE SENZOROVÉ SÍTĚ (v ZIP archivu) včetně všech zdrojových kódů pro zkopírování na webový server. Adresář obsahuje i všechny další soubory potřebné pro bezproblémový provoz webové aplikace na webovém serveru.

### **doc/**

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

### **data/**

Adresář obsahuje soubory se zdrojovými daty obdržnými z experimentálních měření a také soubory s transformovanými daty pro import do databáze.

### **readme.txt**

Instrukce pro nasazení webové aplikace WEBOVÉ ROZHRANÍ PRO ZOBRAZOVÁNÍ SENZOROVÝCH DAT ZE SENZOROVÉ SÍTĚ na webový server, včetně všech požadavků pro její bezproblémový provoz, a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce. Text souboru *readme.txt* je také obsahem přílohy 1.



## Bibliografie

- [1] CASTAGNETTO, Jesus; RAWAT, Harish; SCHUMANN, Sascha; SCOLLO, Chris; VELIATH, Deepak. PHP programujeme profesionálně. Wrox Press Ltd., 1999 / Praha: Computer Press, 2. opravené a aktualizované vydání 2002. xxii, 656 str. ISBN 80-7226-310-2. <http://www.cpress.cz>
- [2] MIKLE, Pavol. Rozšířené dynamické HTML - referenční příručka. Brno: Zoner Press, 2004. xv, 206 str. ISBN 80-86815-01-3. <http://www.zonerpress.cz>
- [3] MIKLE, Pavol. XCSS - CSS1, CSS2, CSS2.1 - úplná přesná referenční příručka. Brno: Zoner Press, 2004. viii, 231 str. ISBN 80-86815-13-7. <http://www.zonerpress.cz>
- [4] THE PHP GROUP. PHP Documentation [online], 2017. <http://php.net>
- [5] W3C. W3C [online], 2017. <https://www.w3.org>
- [6] REFSNES DATA. W3Schools [online], 2017. <https://www.w3schools.com>
- [7] STACK EXCHANGE INC. Stack Overflow [online], 2017. <http://www.stackoverflow.com>
- [8] JANOVSKEJ, Dušan. Jak psát web [online]. ISSN 1801-0458, 2017. <https://www.jakpsatweb.cz>
- [9] WIKIPEDIA, [online]. <https://cs.wikipedia.org/wiki/IIS>
- [10] STATCOUNTER, [online]. <http://gs.statcounter.com>
- [11] STATCOUNTER, [online]. <http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide>
- [12] HTML5TEST, [online]. <https://html5test.com>
- [13] GOOGLE, [online]. <https://developers.google.com/chart/interactive/docs/gallery>
- [14] GOOGLE, [online]. <https://code.google.com/archive/p/gchartphp/>
- [15] FLOTCHARTS, [online]. <http://www.flotcharts.org>