Univerzita Palackého v Olomouci
Přírodovědecká fakulta
Katedra geoinformatiky

ANALYTICKÉ ZPRACOVÁNÍ DAT V PROSTŘEDÍ SENZOROVÉHO WEBU

Magisterská práce

Marek BALUN

Vedoucí práce doc. RNDr. Vilém Pechanec, Ph.D.

Olomouc 2016
Geoinformatika

ANOTACE

Tato diplomová práce se zabývá vývojem prototypu Dohledového centra senzorové sítě. Dohledové centrum, vyvinuté v rámci diplomové práce je informační systém pro správu a jednoduché analýzy senzorových dat uložených v databázovém systému PostgreSQL. Podporuje také export dat do různých formátů např. CSV, XML a import dat z CSV a databází.

KLÍČOVÁ SLOVA

senzorový web; senzorová data; IoT; PHP; PostgreSQL

Počet stran práce: 45

Počet příloh: 3 (z toho 2 volné)

ANOTATION

The aim of this diploma thesis is development of protoype Control centre of sensors network. Control centre developed in context diploma thesis is web based information system that supports management of sensor data stored in PostgreSQL databases. Control centre also supports simple analys of sensor data, export data to several file format e.g. CSV, XML and import data from CSV file and from databases.

KEYWORDS

sensor web; sensor data; IoT; PHP; PostgreSQL

Number of pages: 45
Number of appendixes: 3

Prohlašuji, že

- diplomovou práci včetně příloh, jsem vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu.

- jsem si vědom, že na moji diplomovou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,

- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou práci užívat (§ 35 odst. 3),

- souhlasím, aby jeden výtisk diplomové práce byl uložen v Knihovně UP k prezenčnímu nahlédnutí,

- souhlasím, že údaje o mé diplomové práci budou zveřejněny ve Studijním informačním systému UP,

- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užít výsledky a výstupy mé diplomové práce v rozsahu § 12 odst. 4 autorského zákona,

- použít výsledky a výstupy mé diplomové práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Olomouci dne Marek Balun

Na tomto místě bych chtěl poděkovat vedoucím práce doc. Vilémovi Pechancovi za odborné vedení, cenné rady a konzultace.

UNIVERZITA PALACKÉHO V OLOMOUCI
Přírodovědecká fakulta
Akademický rok: **2013/2014**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Marek BALUN**
Osobní číslo:	**R130115**
Studijní program:	**N1301 Geografie**
Studijní obor:	**Geoinformatika**
Název tématu:	**Analytické zpracování dat v prostředí senzorového webu**
Zadávající katedra:	**Katedra geoinformatiky**

Zásady pro vypracování:

Cílem diplomové práce je tvorba prototypu kontrolního panelu dohledového centra, který bude obsahovat prostředí pro základní parametry sítě, umožní částečnou vzdálenou konfiguraci jednotlivých prvků KSS a zabezpečí import dat ze senzoru/primárního úložiště. V této fázi vyjde z již existujících dílčích řešení na KGI a standardů OGC SWE. Progresivní částí systému bude model pro analytické operace senzorových dat. Operace umožní základní principy proudového zpracování dat, vzdálenou selekci a filtraci, zabezpečí popisnou statistiku a vytvoří rozhraní pro prostorové operace podle OGC FE.
Student vyplní údaje o všech datových sadách, které vytvořil nebo získal v rámci práce, do Metainformačního systému katedry geoinformatiky a současně zálohu údajů ve formě validovaného XML souboru. Celá práce (text, přílohy, výstupy, zdrojová a vytvořená data, XML soubor) se odevzdá v digitální podobě na CD (DVD) a text práce s vybranými přílohami bude odevzdán ve dvou svázaných výtiscích na sekretariát katedry. O diplomové práci student vytvoří webovou stránku v souladu s pravidly dostupnými na stránkách katedry. Práce bude zpracována podle zásad Voženílek (2002) a závazné šablony pro diplomové práce na KGI (2010).

OBSAH

SEZNAM POUŽITÝCH ZKRATEK

Zkratka	Význam
AES	Advanced Encryption Standard
AMQP	Advanced Message Queuing Protocol
API	Application Programing Interface
AWS	Amazon Web Services
CSS	Cascading Style Sheets
CSV	Comma Separated Value
FQDN	Fully Qualified Domain Name
GIS	Geographical Information System
GML	Geography Markup Language
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPD (httpd)	HTTP Deamon (Apache HTTP Server)
HTTPS	Hypertext Transfer Protocol Secure (HTTP over SSL)
IDE	Integrated Development Environment
IoT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
MQTT	MQ Telemetry Transport
OGC	Open Geospatial Consortium
O&M	Observations & Measurements
OSM	Open Street Map
PHP	Hypertext Preprocessor (dříve Personal Home Page)
REST	Representational State Transfer
SAS	Sensor Alert Service
SensorML	Sensor Model Language
SES	Sensor Event Service
SHP	Esri Shapefile
SOAP	Simple Object Access Protocol
SOS	Sensor Observation Service
SPS	Sensor Planning Service
SQL	Structured Query Language
SWE	Sensor Web Enablement
TLS	Transport Layer Security
URL	Uniform Resources Locator
XML	eXtensible Markup Language
WAN	Wide Area Network
WNS	Web Notification Service

ÚVOD

Senzorové technologie se aktuálně stávají velkým trendem. Předpokládá se, že v následujících letech to bude jeden z nejrychleji rostoucích segmentů trhu s informačními technologiemi. Stále větší množství nejrůznějších zařízení je schopno sbírat a přenášet data. Škála těchto zařízení je opravdu široká - může se jednat jak o specializované senzorové systémy pro sběr meteorologických dat, senzory využívané v průmyslové automatizaci, tak o zařízení z kategorie spotřební elektroniky např. náramky na měření tepové frekvence. S využitím těchto technologií počítají třeba koncepce inteligentních budov [3], inteligentních měst (*smart city*) [21], inteligentních sítí (*smart grid*) [22] a také jsou nezbytným předpokladem pro další rozvoj robotiky. Všechna tato zařízení se dnes často souhrnně označují jako internet věcí. Toto slovní spojení nedávno ještě neznámé proniká stále rychleji do obecného povědomí. Aplikace zpracovávající data z těchto zařízení jsou označovány jako web věcí [7]. Senzorový web pak lze považovat za podmnožinu webu věcí – tedy webové aplikace zaměřené na zpracování dat pouze ze senzorů. Diplomová práce se bude zabývat vývojem právě takovéto aplikace.

1 CÍLE PRÁCE

Cílem diplomové práce je tvorba prototypu kontrolního panelu Dohledového centra senzorové sítě. Dohledové centrum bude webová aplikace, která bude schopná dynamicky připojovat databáze se senzorovými daty a bude umožňovat uživateli definovat vlastní dotazy nad těmito daty bez zásahu do zdrojového kódu aplikace a znalosti jazyka SQL. Aplikace Dohledového centra bude dále umožňovat data vizualizovat v tabelární podobě a v podobě grafů. Bude také podporován exportovat do vybraných prostorových (GML a Esri Shapefile) a neprostorových formátů (XML a CSV) a import ze souborů CSV a z jiných databází. Aplikace bude také zobrazovat jednoduchou popisnou statistiku dat a bude obsahovat nástroje na analýzu senzorových dat.

2 METODY A POSTUPY ZPRACOVÁNÍ

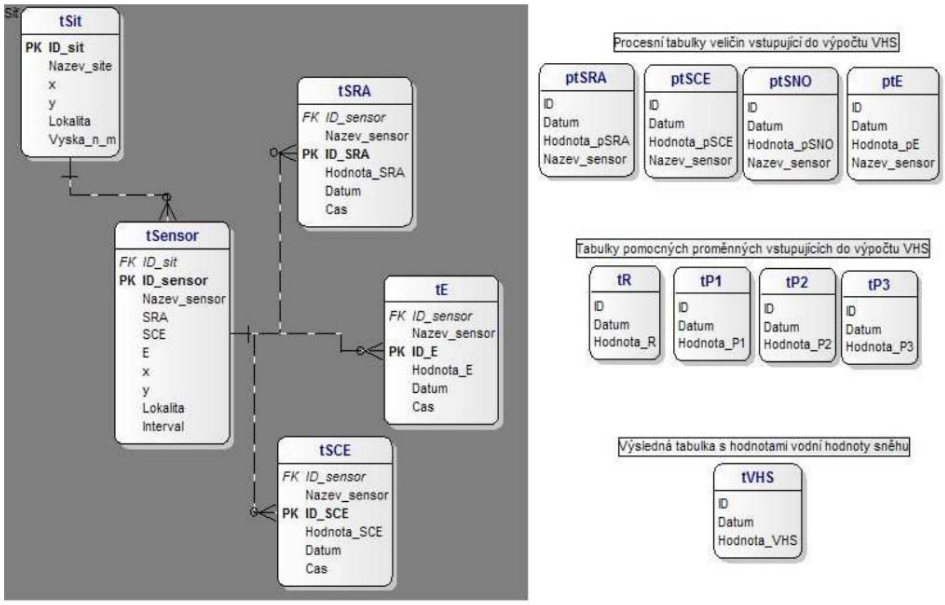
Použité metody

V této práci nebyla sbírána ani analyzována žádná data, práce se striktně zaměřuje na vývoj aplikace Dohledového centra senzorové sítě. Soustředí se na technické řešení samotné aplikace a práci s databází nikoliv na zpracování dat a jejich následnou interpretaci. Hlavní náplní byla tedy vývojářská práce zahrnující programování, debugging, ladění, testování aplikace a návrh servisní databáze. Kromě toho samozřejmě studium různých informačních zdrojů a vyhledávání informací na webu.

Použitá data

V rámci zpracování diplomové práce nebyla pořízena žádná vlastní data. Jako testovací a ukázková databáze byla použita databáze *vodni_hodnota_snehu* pořízená Liborem Kimplem při zpracování jeho diplomové práce *Standardy pro dohledové centrum senzorové sítě (2012).* Databáze obsahuje meteorologická data naměřená senzorovou sítí provozovanou Katedrou geoinformatiky u obce Vysoké Pole (okres Zlín). Na lokalitě byly monitorovány tři veličiny – celková výška sněhu, denní úhrn srážek a průměrný denní tlak vodní páry. Data pocházejí z roku 2011. Jedná se o časovou řadu měření od 23. 11. 2011 do 24. 12. 2011. Časové intervaly mezi jednotlivými měřeními závisejí na konkrétním senzoru. Nejkratší interval je 1x za 10 min, nejdelší 1x denně. Databáze obsahuje data z devíti senzorů (tři senzory na každou měřenou veličinu). Původní databáze byla pro potřeby aplikace mírně upravena, např. velká písmena v názvech tabulek a v názvech sloupců byla změněna na mínusky. Velká písmena v názvech způsobovala, že SQL dotazy vracely chybu sloupec nenalezen, neexistující objekt.

Pro potřeby aplikace byla vytvořena servisní databáze *ControlCenterMgmtDB*. Databáze neobsahuje žádná naměřená data, slouží pouze k ukládání konfigurace aplikace. Více podrobností o této databázi je uvedeno v části Vlastní řešení.



Obr. 1 Schéma původní databáze *vodni_hodnota_snehu* [10]

Použité programy

- NetBeans IDE 8.1 (NetBeans IDE 8), verze pro PHP, od společnosti Oracle – sloužilo jako primární vývojové prostředí, pro jazyk PHP a HTML
- PSPad 4.6 – pro editaci souborů
- GitHub – sloužil jako repozitář zdrojových kódů a verzovací systém (aplikace byla vyvíjena na více zařízeních a zdrojový kód byl mezi nimi synchronizován pomocí GitHub)
- XAMPP 1.8.2 – balík obsahující webový server (httpd), PHP a databázový systém MySQL a klienta phpMyAdmin
- PostgreSQL 9.5 – databázový server
- pgAdminIII – desktopový klient pro PostgreSQL
- Windows Server 2012 R2, Windows 10 – vývojové a testovací prostředí bylo hostováno na platformě Windows
- Hyper-V – virtualizační platforma od společnosti Microsoft, testování probíhalo na virtualizovaných serverech, byla využita jak klientská verze (Windows 10 Pro) tak serverová verze (Windows Server 2012 R2 Datacenter) Hyper-V
- QGIS 2.10 – open source GIS software, pro vizualizaci prostorových dat uložených v PostgreSQL (PostGIS) při testování, úprava datových vrstev
- Toad Data modeler 5.4 Freeware – vytvoření schématu databáze
- XMind 7.5 – vytvoření mapy webových stránek
- Microsoft Visio 2013 – tvorba grafických schémat
- Microsoft Word 2016 – sepsání textové části práce
- Adobe Photoshop CS 5 – tvorba posteru

Postup zpracování

Pro vývoj aplikace bylo zvoleno vývojové prostředí NetBeans IDE. Při výběru IDE bylo požadováno, aby umělo zvýrazňovat syntaxi, zvýrazňovat syntaktické chyby, doplňovat zdrojový kód a obsahovalo nápovědu k funkcím a klíčovým slovům jazyka. Textové editory jako PSPad, Notepad++, Sublime Text apod. byly hned zavrženy. Co se týče vývojových prostředí s pokročilejšími funkcemi, je na výběr více produktů, ale většina z nich je proprietárních a placených např. PhpStorm, Komodo IDE, Zend Studio. Ze světa open source jsou k dispozici dva produkty - NetBeans IDE a Eclipse. Z nich bylo vybráno NetBeans kvůli přehlednému uživatelskému rozhraní a nativní podpoře PHP (verze pro PHP, Eclipse vyžaduje pro vývoj v PHP plugin PHP Development Tools).

Jako platforma pro hostování aplikace byl vybrán Windows Server. Windows byl upřednostněn před GNU/Linux, protože autor se profesně zabývá serverovými technologiemi společnosti Microsoft.

Vývoj aplikace probíhal iterativně po jednotlivých funkcích. Vždy byla vyvíjena jedna komponenta, která byla zároveň testována. Na začátku bylo vytvořeno jádro aplikace, které tvořil mechanismus pro dynamické připojování databází a jednoduché prohlížení tabulek bez možností filtrování a řazení záznamů. Na to byly postupně nabalovány další funkce. Následovaly funkce pro export a import dat, podpora grafů, volitelné řazení výsledků dotazů, implementace uživatelských účtů a přihlašování atd. Na závěr se řešila validace HTML kódu vygenerovaného pomocí PHP a nápověda.

Často se během vývoje stávalo, že se některé koncepce ukázaly jako nevyhovující, proto se vracelo zpět a některé funkce a komponenty se kompletně předělávaly. Např. editace tabulek byla zcela přepsána, po zjištění, že je funkční jen u tabulek s menším počtem záznamů, ale u velkých tabulek narazí na limit proměnných, které lze v PHP

vytvořit v rámci jedné *session*. Autor se také zároveň učil pracovat s jazykem PHP, jak se zlepšovaly jeho schopnosti v tomto jazyce, docházel k závěru, že některé věci lze naprogramovat lépe. Aplikace byla proto několikrát kompletně přepsaná a z původních zdrojových kódů nezbylo téměř nic. Celá aplikace je programována dle procedurálního (imperativního) paradigmatu, objektových prvků je v aplikaci zastoupeno jen minimum – zejména se jedná o fragmenty JavaScriptu.

3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

3.1 Základní pojmy

Senzor

Senzor je hardwarové zařízení, čidlo, které měří určitou fyzikální veličinu v prostředí. Tato veličina je nejčastěji přenášena v elektrické podobě označované jako signál. V obecnějším smyslu lze za senzor považovat např. i člověka pokud se podílí na sběru a přenosu dat. Senzory lze dělit dle různých kritérií – dle měřené veličiny, fyzikálního principu, na němž jsou založeny, podle styku s prostředím (kontaktní a bezkontaktní), na analogové a digitální. Senzory mohou být jednoduché, zaznamenávající pouze jednu fyzikální veličinu, nebo komplexní schopné sledovat celé spektrum veličin. Mezi základní parametry senzorů patří citlivost, práh citlivosti, dynamický rozsah, reprodukovatelnost a rozšiřitelnost [20].

Systém senzorů

Systém senzorů je označení pro větší množství různých senzorů, které společně tvoří jeden systém a obvykle jsou i spojeny do fyzického celku a instalovány na jednu platformu. Např. satelit nebo meteorologická stanice [7]. Senzorový systém umožňuje adresovat každý senzor samostatně.

Senzorová síť

Senzorová síť je soubor senzorů propojených komunikační infrastrukturou, která zajišťuje přenos dat (signálu) ze senzorů a jejich uložení na perzistentní úložiště – buď lokálně do dataloggeru nebo do cloudu, pokud má síť konektivitu do WAN. Základní dělení senzorových sítí je dle způsobu přenosu dat – na drátové senzorové sítě a bezdrátové senzorové sítě. Kromě kabelových spojů a éteru může senzorová síť k přenosu využívat i agentů. Agent je nějaká autonomní entita, která zajišťuje přenos dat z primárního úložiště. Příkladem agenta může být člověk nebo robot [7].

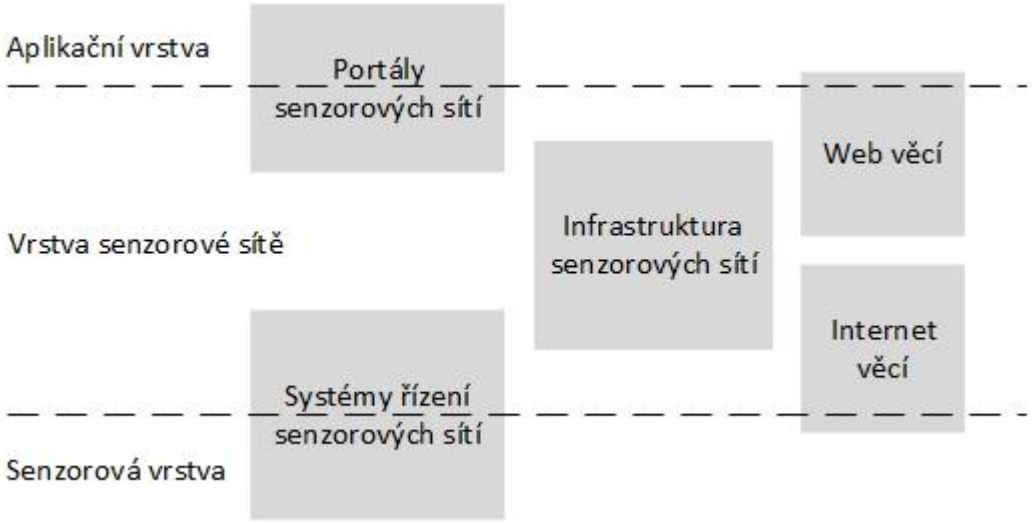
Internet věcí

Jako internet věcí (z anglického *Internet of Things* – zkr. *IoT*) se označují různá zařízení (obvykle jednoduchá, bez vlastního výpočetního výkonu) která je možno připojit k internetu (mají síťová rozhraní, kterým je možno přidělit IP adresu), za účelem přenosu dat a komunikaci s těmito zařízeními. Typickým zařízením *IoT* je právě senzor pro sběr dat, internet věcí je proto velmi úzce spjat se senzorovými systémy a tyto pojmy jsou dnes často používány jako synonyma. Pojem internet věcí je také často označován jako *buzzword*.

Senzorový web

Senzorový web je webové rozhraní, které umožňuje přístup k senzorovým datům i k senzorům samotným využívající standardizované protokoly a API [8]. Na senzorový web lze pohlížet jako na prostředníka mezi webovými aplikacemi a samotnými senzory. Senzorový web je specifický typ webu věcí (web věcí je chápán obecnější pojem), který je úzce zaměřen na zpracování dat ze senzorů. Architekturu senzorového webu lze rozdělit do tří logických vrstev. Nejníže leží vrstva senzorů, která reprezentuje hardwarové zařízení, tedy nejrůznější senzory. Nad ní se nachází vrstva senzorové sítě, která zajišťuje

komunikaci mezi senzory jakožto zdroji dat a koncovými aplikacemi. Nejvýše se nachází aplikační vrstva tvořena aplikacemi určenými pro koncové uživatele [7].



Obr. 2 Architektura senzorových systémů [7]

3.2 Standardy OGC

Open Geospatial Consorcium (OGC)

Open Geospatial Consorcium je mezinárodní nezisková organizace zabývající se standardizací v oblasti geoprostorových dat. Sdružuje jak komerční subjekty, tak i neziskové organizace, vládní instituce, výzkumné organizace a univerzity. Mezi významné členy patří např. NASA, USGS, NOAA, Google, Oracle, Esri, Intergraph Corporation, Lockhed Martin, BAE Systems a další. V současnosti má OGC více jak 500 členů [19].

Zájem OGC se soustředí na několik tematických oblastí – letectví, geovědy a environmentalistiku, business inteligence, obranu, krizový management, energetiku, mobilní internet, polohové služby a senzorový web [19], kterým se bude tato práce podrobněji zabývat.

Sensor Web Enablement (SWE)

Sensor Web Enablemet zastřešuje všechny ostatní standardy zabývající se senzorovým webem a senzorovými sítěmi. Jedná se o iniciativu OGC definující rozhraní služeb, které umožňují interoperabilní využití senzorových zdrojů. SWE je definováno dvěma standardy – jednak SWE Common Data Model Encoding Standard a SWE Service Model Implementation Standard.

SWE Common Data Model Encoding Standard

Standard definuje nízkoúrovňové modely pro výměnu dat mezi uzly sítě postavené na standardech OGC SWE. Tyto modely umožňují aplikacím a serverům kódovat a přenášet senzorová data a těmto datům přidávat sémantický význam. Přesněji SWE Common Data

Model definuje datové reprezentace, kódování, kvalitu, strukturu a vlastnosti dat (*nature of data*) – tyto vlastnosti slouží k základnímu popisu datového proudu.

Datové reprezentace určují, jaké datové typy budou použity pro jednotlivé sledované charakteristiky např. booleovský datový typ, kategoriální, číselný, textový řetězec atd. Datové reprezentace také definují omezení např. rozsah nebo výčet hodnot, kterých může daná veličina nabývat.

Nature of data jsou definovány, jako informace nutné k porozumění tomu jakou vlastnost daná hodnota reprezentuje. Může být zajištěna pomocí externích zdrojů v podobě řízených slovníků nebo ontologií.

Kvalita dat je informace pro spotřebitele dané služby. Je asociována s konkrétní měřenou veličinou a závisí na datovém typu.

Struktura dat udává, jak jsou jednotlivá data v rámci datového proudu organizována do skupin, řazena a v jakých intervalech jsou doručována. Data mohou být agregována do záznamu (*record*) – sestává se seznamu polí (*fields*) která jsou definována svým datovým typem - do pole (*array*) – sestává ze skupiny hodnot stejného datového typu - nebo jako výčtový typ (*choice*) – sestává ze seznamu alternativních hodnot, kde každá může mít vlastní datový typ.

Kódování určuje, jak jsou data řazena do bloků za účelem efektivního přenosu a ukládání s využitím různých formátů a protokolů.

SWE Common language je XML implementace SWE datového modelu a je využívána dalšími standardy jako SensorML SOS, SPS, O&M atd. [17].

SWE Service Model Implementation Standard

Tento standard specifikuje datové typy a rozhraní společné pro služby senzorového webu, proto slouží jako základ pro vývoj dalších služeb.

- je aplikovatelný na všechny služby, které poskytují data ze senzorů
- specifikuje přístup k popisu senzorů, a jak může být tento popis spravován
- stanovuje prostředky pro vkládání a mazání senzorů přes servisní rozhraní
- specifikuje funkce pro publikování/odběr dat založené na definici rozpoznatelných typů událostí
- specifikuje informace potřebné k využití SOAP

Standard v současnosti definuje osm balíčků s datovými typy pro běžné užití napříč SWE službami. Jedná se o následující balíčky:

Contents – balíček specifikuje datové typy, které mohou být použity ke specifikaci služeb, které poskytují přístup k senzorům.

Notifications – balíček definující datové typy pro podporu poskytování metadat

Common – datové typy společné pro všechny balíčky

Common Codes – balíček definuje běžně používané seznamy kódů se speciální sémantikou

DescribeSensor – definuje datové typy pro požadavek/odpověď operace pro získání metadat ze senzoru

UpdateSensorDesription - definuje datové typy pro požadavek/odpověď operace používané pro popis senzoru

InsertSensor – definuje datové typy pro požadavek/odpověď používané pro operace vložení sensoru

DeleteSensor - definuje datové typy pro požadavek/odpověď používané pro operace smazání sensoru [18]

SensorML

SensorML je zkratka pro Sensor Model Language. Tento standard popisuje vlastností senzoru, definuje modely a schéma jazyka XML. Schéma má ve jmenném prostoru prefix *sml*. Standard zahrnuje jak senzory používané při dálkovém průzkumu Země tak senzory pro měření in-situ. SensorML je určen pro:

- popis senzorů a senzorových systémů
- procesing a popis analytických nástrojů pro jednotlivá pozorování
- geolokaci výsledků měření
- strojově čitelný popis rozhraní a proudů dat
- popis procesů, pomocí kterých jsou získány výsledky měření
- poskytování procesů pro získávání nových dat

Základem SensorML je *AbstractProcess*, z něj jsou odvozeny ostatní typy procesů, ty lze rozdělit na fyzické a nefyzické procesy. *AbstractProcess* je definován následujícími vlastnostmi – název, popis, parametry, metedata a vstupy (pro fyzické procesy může být vstupem pozorovaný jev, pro nefyzický proces jsou vstupem příchozí data) a výstupy. Aktuální verze standardu je 2.0 [14].

Observation & Measurments (O&M)

Tento standard poskytuje model pro popis výsledků senzorových měření. Měření je zde chápáno jako událost, jež probíhá v určitém přesně definovaném čase a je pomocí něj získána hodnota pro sledovaný jev v daném čase. Kromě naměřené hodnoty a času může O&M popisovat i další vlastnosti jako například proces, jímž je generována naměřená hodnota. Tento standart definuje XML schéma pro pozorování. Schéma používá ve jmenném prostoru prefix *om*. Standart se skládá ze dvou částí, první popisuje samotné měření a druhá popisuje vzorkování výsledků. Měření musí být vždy vázáno na proceduru, která představuje proces, kterým bylo měření provedeno – může se jednat např. o fyzický senzor nebo simulaci. Standart definuje následující vlastnosti:

featureOfInterst – skutečný objekt reálného světa – předmět pozorování

observedProperty – pozorovaná vlastnost tohoto objektu

procedure – popis procesu jež generuje výsledek

result – hodnota vygenerovaná procedurou [13]

Sensor Observation Service (SOS)

Tento standard definuje jak přistupovat standardizovanou cestou k popisu senzorů, pozorování a počítačové reprezentaci pozorovaných vlastností. Dále standard definuje jak zaregistrovat nové senzory a odstraňovat ty existující. Také specifikuje jak vkládat nové pozorování a jak získávat výsledné naměřené hodnoty. Aktuální verze standardu je 2.0. Standard definuje následující operace:

GetObservation – poskytuje přístup k výsledkům měření, prostorové a časové filtrování výsledků

GetCababilites – přístup k metadatům, a informace o operacích poskytovaných SOS serverem

DescribeSensor – umožňuje získat popis senzorů ze SOS serveru [15]

Sensor Planning Service (SPS)

Standart SPS definuje rozhraní, pomocí kterých je možné vyvářet a plánovat úlohy pro zařízení jako senzory, aktuátory a senzorové systémy. Standard je stejně jako předešlé postaven na XML schématu a aktuální verze je 2.0. Rozhraní definované ve standardu umožňuje:

- získat metadata služeb
- popsat nastavení dostupné na jednotlivých sensorech
- zkontrolovat jestli může služba vykonat požadovanou úlohu
- rezervovat si zdroje v určitou dobu, aby mohli vykonat plánovanou úlohu
- instruovat službu aby vykonala plánovanou úlohu
- upravit naplánovanou úlohu
- zrušit plánovanou úlohu
- získat informace jak přistupovat k naměřeným datům

Standart SPS definuje následující povinné operace, které musí rozhraní nabízet:

GetCapabilites – umožňuje získat metadata

DescribeTasking – získání informací které jsou potřeba pro naplánování úlohy

Submit – tato operace popisuje přijetí úlohy

GetStatus – získání informací o naplánované úloze

GetTask – poskytuje kompletní informace o naplánované úloze

DescribeResultAccess – říká klientovi jak přistupovat k naměřeným datům [16]

3.3 Cloudová řešení

S rostoucím zájmem o IoT a senzorové technologie se pomalu začínají objevovat řešení nabízející nástroje pro příjem, skladování a zpracování senzorových dat. Nejpropracovanější a nejrobustnější řešení nabízejí provozovatelé velkých cloudových platforem jako Amazon (AWS), Microsoft (Microsoft Azure), Google (Google Cloud Platform), IBM (IBM Cloud), Oracle (Oracle Cloud) a další. Žádné z těchto řešení neimplementuje standardy OGC SWE a jsou postaveny převážně na proprietárních službách provozovaných v těchto cloudech, ale vzhledem k tomu, že jejich provozovatelé mají na trhu s informačními technologiemi dominantní postavení, lze předpokládat, že budou do budoucna určovat trendy v této oblasti.

Amazon Web Services IoT (AWS IoT)

AWS IoT je velká platforma pro ukládání a zpracování senzorových dat od společnosti Amazon. AWS IoT je hostován v cloudu AWS a koncovým spotřebitelům je poskytován formou služby. AWS IoT poskytuje zabezpečenou obousměrnou komunikaci mezi zařízeními připojenými k internetu (senzory, aktuátory a jiná „chytrá“ zařízení) a cloudem AWS pro sběr telemetrických dat. AWS IoT se skládá z následujících komponent:

Device Gateway – Zajišťuje obousměrnou komunikaci mezi zařízeními a AWS cloudem. Podporuje připojení zařízení přes protokoly HTTP (HTTPS), WebSockets a MQTT (MQ Telemetry Transport) [4].

Message Broker – Poskytuje zabezpečený mechanismus pro výměnu zpráv mezi koncovými zařízeními a aplikacemi. Lze použít protokol MQTT nebo HTTP REST.

Rules Engine – Zajišťuje zpracování zpráv a integraci se službami AWS. Pro výběr dat z těla zpráv lze použít jazyk založený na SQL. Data lze pak přeposílat službám jako

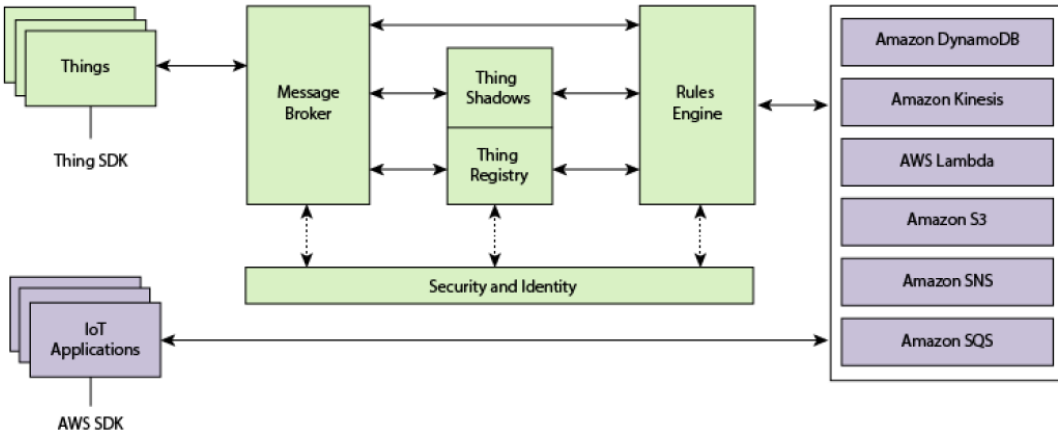
Amazon S3, *Amazon DynamoDB* nebo *AWS Lambda*, případně jiným službám poskytovaným třetí stranou.

Thing Registry – Umožňuje organizovat zdroje asociované s jednotlivými zařízeními. S každým registrovaným zařízením umožňuje asociovat až tři vlastní vlastnosti.

Thing Shadow – Jedná se o virtuální reprezentace zařízení. Je tvořena JSON dokumentem který uchovává aktuální informace o stavu zařízení.

Thing Shadow service – Představuje perzistentní reprezentace zařízení v AWS cloudu. Zařízení IoT z důvodu energetické nenáročnosti nejsou nepřetržitě online, ale připojují se jen na krátkou dobu v určitých intervalech. Koncové aplikace k nim tedy nemohou mít nepřetržitě přístup, z toho důvodu je pro ně stav zařízení reprezentován službou *thing Shadow service*. V okamžiku kdy je zařízení online, informace o jeho stavu se synchronizuje s *thing shadow*.

Security and Identity Service – Zajišťuje zabezpečenou komunikaci s využitím šifrování a certifikátů. Podporuje také *AWS Identity Access Management* (*AWS IAM*) což je služba AWS pro řízení přístupu ke zdrojům [2].



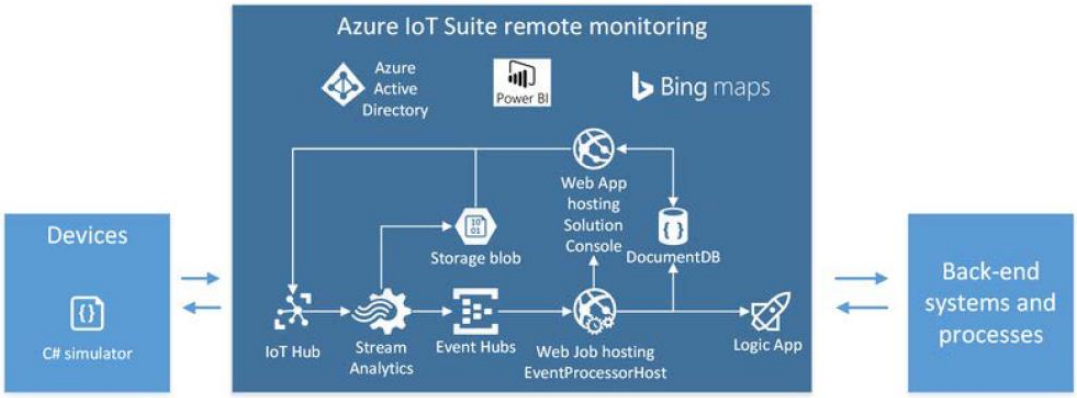
Obr. 3 Schéma AWS IoT [2]

Microsoft Azure IoT Suite & Azure IoT Hub

Azure IoT Suite je další řešení pro zpracování telemetrických dat, tentokrát od konkurenční společnosti Microsoft. Jedná se o kolekci aplikací a služeb, které mají na starosti ukládání dat, analytiku nad proudovými daty, business logiku, transformaci dat a jejich vizualizaci. IoT Suite obsahuje set předpřipravených řešení, které zákazníci nejčastěji požadují. V současnosti jsou k dispozici šablony pro dva scénáře – vzdálený monitoring (*remote monitoring*) a prediktivní údržba (*predictive maintanance*). Oba scénáře obsahují následující funkcionalitu:

- Přijímání dat
- Správa identity připojených zařízení
- Odesílání zpráv připojeným zařízením a plánování akcí těchto zařízení
- Správa pravidel

Jak už název napovídá IoT Suite je hostován v cloudu Microsoft Azure a využívá dalších služeb provozovaných v tomto cloudu. Služba byla spuštěna v září 2015 [12].



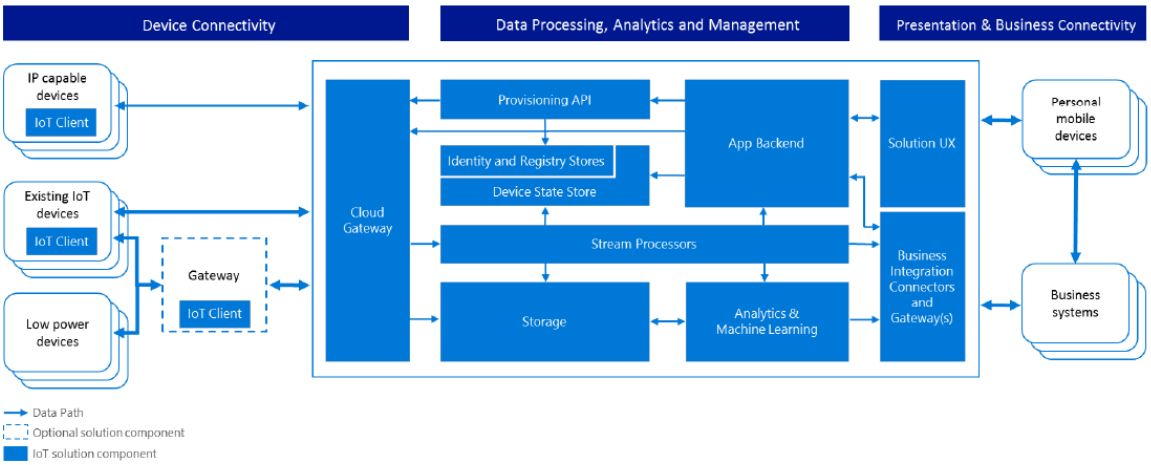
Obr. 4 Architektura Microsoft Azure IoT Suite [12]

O měsíc později – v říjnu 2015 Microsoft spouští další službu zaměřenou na internet věcí – Azure IoT Hub. Jedná se o službu pro směrování správ. IoT Hub má na starosti zabezpečenou obousměrnou komunikaci mezi cloudem a připojenými zařízeními. Tato služba je přímou konkurencí a reakcí na AWS IoT. IoT Hub podporuje komunikaci pomocí protokolů HTTP, AMQP a MQTT. Veškerá komunikace musí být zabezpečena použitím protokolu TLS. IoT Hub tvoří tři hlavní komponenty [12].

Identity store – Spravuje identitu zařízení. Obsahuje databázi kryptografických otisků zařízení, na jejichž základě identifikuje příchozí komunikaci od klientů. Může být realizován pomocí *Azure DocumentDB*, *Azure Tables*, SQL databází nebo pomocí jiného řešení třetí strany.

Registry store – Tento registr obsahuje metadata jako např. geografické informace nebo identifikační údaje zařízení a umožňuje uživateli se na tyto metadata dotazovat za účelem nalezení konkrétního zařízení. Může být realizován pomocí *Azure DocumentDB*, SQL databází nebo pomocí jiného řešení třetí strany.

Device state store - Jedná se o ekvivalent *Thing Shadow* z AWS IoT. Obsahuje rychle se měnící operační data jako poslední naměřené hodnoty a informace o stavu zařízení. Může být postaven na celé řadě nejrůznějších technologií jak od Microsoftu tak jiné třetí strany např. *Azure Data Lake*, *Azure Blob Storage*, *Azure Tables*, SQL databází atd. [11].



Obr. 5 Architektura Microsoft Azure IoT Hub [12]

3.4 Další řešení senzorového webu

Řešení, která se snaží o implementaci standardů OGC SWE jsou většinou menší a komunitní projekty. Asi nejznámější projektem je implementace sdružení 52°North. Mezi další projekty patří open source projekt Deegree nebo projekt Foodie od českého sdružení WIRELESSINFO a implementací standardu SOS se zabýval také ve své diplomové práci Michal Kepka ze ZČU [9]. Vlastní řešení senzorového webu nabízejí i některé české firmy zabývající se senzorovými systémy jako Fiedler, Ekotechnika nebo EMS Brno. Tyto řešení se nedrží standardů OGC SWE.

52°North

Open source iniciativa 52°North je mezinárodní síť partnerských organizací v oblasti výzkumu, průmyslu a veřejné správy. Jejím hlavním cílem je podpora inovací v oblasti geoinformací. Uvnitř sdružení funguje komunita zabývající se vývojem senzorového webu, která stojí za vývojem asi nejznámější implementací standardů OGC. Implementace sdružení se striktně drží OGC SWE a snaží se maximální interoperabilitu. 52°North implementuje následující standardy – SOS, SAS, SES, SPS a WNS. Navíc k těmto standardům poskytují vlastní klienty např. pro ArcGIS nebo webového klienta vytvořeného v JavaScriptu [1].

Fiedler

Fiedler je česká společnost zabývající se vývojem a prodejem vlastních telemetrických sestav. Zaměřují se především na sběr hydrologických dat. Pro přístup k datům nabízejí vlastní řešení – Webový prohlížeč měřených dat. Aplikace je určená provozovatelům telemetrických stanic, kteří využívají datahosting této společnosti a umožňuje grafickou i tabelární vizualizaci dat. Společnosti Fiedler ještě provozuje webový portál Hladiny.cz na kterém jsou pro veřejnost přístupné vodočty z hlásných profilů ČHMÚ a podniků Povodí s.p. [6].

Ekotechnika

Společnost působící na českém, která nabízí měřící a vzorkovací přístroje celé řady světových výrobců. Zaměřuje se na monitoring v oblasti geologie, pedologie, hydrologie, meteorologie a monitoring životního prostředí. Pro správu naměřených dat poskytuje aplikaci EnviroDATA. Jedná se o webovou aplikaci, která nabízí zobrazení dat v grafech a v tabelární podobě, předzpracování dat, statistické funkce a export dat pro další zpracování [5].

4 VLASTNÍ ŘEŠENÍ

Cílem praktické části práce bylo vytvořit Dohledové centrum senzorové sítě a naprogramovat funkce pro analytické zpracování dat. Dohledové centrum senzorové sítě vytvořené v rámci této diplomové práce je webové rozhraní, pomocí kterého je možno přistupovat k senzorovým datům uloženým v databázích, tato data zobrazovat v tabelární podobě, v podobě grafů a provádět nad nimi jednoduché analytické operace. Kromě samotných dat dále Dohledové centrum obsahuje informace o jednotlivých senzorech a sítích použitých pro sběr dat, o lokalitách na kterých probíhá měření a o měřených veličinách.

4.1 Požadavky na aplikaci

- Hlavní požadavek byl, aby se jednalo o webovou aplikaci. Výhody webové aplikace oproti desktopové dnes již snad asi není potřeba vysvětlovat.
- Aplikace bude obsluhovat databázový systém PostgreSQL, mimo jiné kvůli již existujícím datům uloženým v této databázi. Tento požadavek předurčil výběr databázového systému, na kterém bude aplikace postavena.
- Databáze nebude k aplikaci „napevno“ připojená, ale bude možné ve webovém rozhraní interaktivně připojovat a odpojovat jednotlivé databáze bez zásahu do zdrojového kódu aplikace.
- Aplikace bude podporovat export dat z databáze do následujících formátů – XML, CSV, dBase, JSON, GeoJSON, GML a SQL.
- Aplikace bude podporovat import dat ze soborů (CSV) a z databázových systémů PostgreSQL (PostgreSQL -> PostgreSQL) a MySQL (MySQL -> PostgreSQL).
- Data uložená v databázích bude možno vizualizovat v tabelární podobě a v podobě grafů.
- Uživatel bude moci interaktivně definovat ve webovém rozhraní vlastní dotazy nad daty bez nutnosti zásahu do zdrojového kódu.
- Vizualizace rozmístění senzorů v terénu.
- Zobrazení jednoduché popisné statistiky dat.

4.2 Použité technologie

Pro tvorbu webového rozhraní byl použit značkovací jazyk HTML5. Pro definici stylů byl použit stylovací jazyk CSS3. Pro skriptování na straně klienta JavaScript a jako technologie pro skriptování na straně serveru slouží jazyk PHP. PHP je zde dominantním jazykem, tvoří převážnou část zdrojových kódů aplikace. Při vývoji byla snaha většinu funkcionality ošetřit na straně serveru a na klientské straně řešit pouze nezbytné minimum, proto JavaScript naopak tvoří jen minoritní část zdrojového kódu. Jako webserver slouží Apache HTTP Server verze 2.4, konkrétně byl využit balík XAMPP. Jako databázový systém slouží PostgreSQL s extenzí PostGIS pro práci s prostorovými daty. Pro potřeby importu dat měl být ještě navíc podporován databázový systém MySQL, ten se ale do finální verze nakonec nedostal. Celý *stack* je potom hostován na platformě Windows Server. Jedná se tedy o *protocol stack* WAPP (OS – Windows Server, webserver – Apache HTTP Server, skriptovací jazyk - PHP, databázový systém - PostgreSQL).

Úplný seznam použitých technologií:

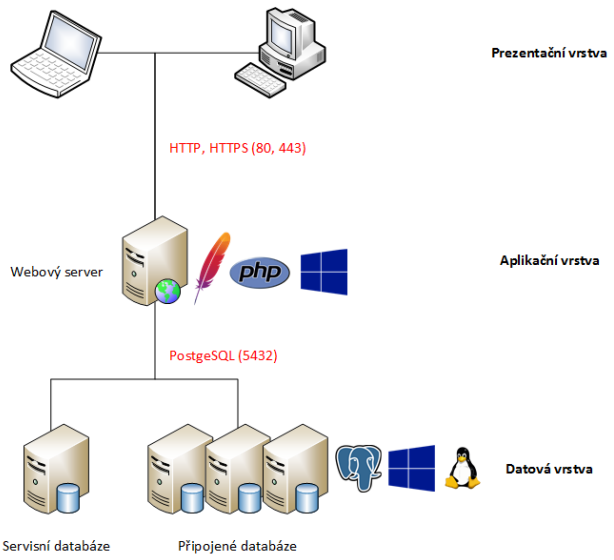
- PHP 5.4.31 (XAMPP 1.8.2)
- HTML 5
- CSS3

- JavaScript
- jQuery 2.1.4
- L.Control.MousePosition.js – JS knihovna, autor Christoffer Rosen
- Leaflet API 0.7
- Esri Leaflet
- Google Charts API
- GDAL/OGR 1.9
- PostgreSQL 9.5
- PostGIS 2.0
- MySQL 5.6 – ve finální verzi není nakonec použit
- Apache HTTP Server 2.4 (XAMPP 1.8.2)
- Windows Server 2012 R2 Standart

Horizontální navigace s rozbalovacími podnabídkami vznikla úpravou CSS šablony stažené ze stránek *http://cssmenumaker.com*. Na tabulky byla použita CSS šablona stažená ze stránek *https://www.freshdesignweb.com/free-css-table.*

4.3 Architektura aplikace

Dohledové centrum senzorové sítě se skládá z vlastní aplikace běžící na webovém serveru (httpd), servisní databáze (PostgreSQL) a libovolného počtu dalších připojených databází (PostgreSQL). Servisní databáze je mandatorní, bez ní je aplikace nefunkční. V případě, že je servisní databáze nedostupná, aplikace se uzamkne, aby se tím předešlo chybovým stavům způsobeným nemožností načíst aplikační data a konfigurační údaje. Další databáze obsahující aplikační data nejsou povinné – aplikace je bez nich funkční, ale její používání bez těchto dat ztrácí smysl. Samotná aplikace musí být hostována na serveru s operačním systémem Windows Server, databáze může být na témže serveru nebo může být připojena ze vzdáleného serveru s libovolným operačním systémem - Windows, GNU/Linux, FreeBSD, Solaris atd.



Obr. 6 Schéma architektury aplikace – *protocol stack* (zdroj: vlastní zpracování)

Skládání stránek

Aplikace se skládá z indexu a dalších šedesáti-tří souborů s příponou *.php*. Hlavičku s horizontální navigací a patičku obsahuje pouze stránka index. Ostatní soubory obsahují pouze tělo stránky s funkčním obsahem. Do stránky index se vkládají pomocí funkce *include()*, která přijímá jako parametr PHP soubor. Parametr je této funkci předáván pomocí metody GET v URL pomocí tzv. *query string*. Uživatel tedy de facto celou dobu pracuje pouze na jedné stránce, která je jen modifikována.

```
index.php?page=dbproperties&database=vodni_hodnota_snehu
```

Příklad URL pro stránku *dbproperties*, kde jsou v *query string* za otazníkem dva parametry oddělené ampersandem – *page*, který říká, která stránka se má načíst a parametr *database*, který je již vztažen k této konkrétní stránce (zde *dbproperties*) a říká, jaká databáze se načte.

Struktura aplikace

Jednotlivé soubory aplikace jsou logicky uspořádány do následujících složek:

conf – obsahuje konfigurační soubor *conf.xml*

export – tato složka slouží jako dočasné úložiště pro exportované soubory, soubor je vygenerován v této složce a následně je odeslán uživateli, po odeslání je soubor ze složky smazán

gdal – obsahuje knihovny GDAL/OGR 1.9

img – obsahuje obrázky – loga a ikony nacházející se na horní liště a screenshoty ze sekce *Nápověda*

import – dočasná složka pro importované soubory CSV, soubor je nejdříve nahrán do této složky a následně je zpracován pomocí SQL příkazu *COPY*, po skončení importu je soubor ze složky smazán

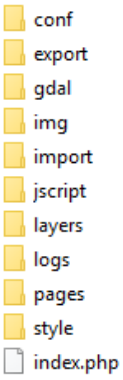
jscript – složka obsahuje JavaScriptové knihovny jQuery a L.Control.MousePosition.js

layers – obsahuje tematické mapové vrstvy ve formátu JSON

logs – obsahuje podsložky *access_log*, *error_log* a *transact_log* do kterých se ukládají jednotlivé logy

pages – obsahuje PHP soubory – části jednotlivých stránek a skripty

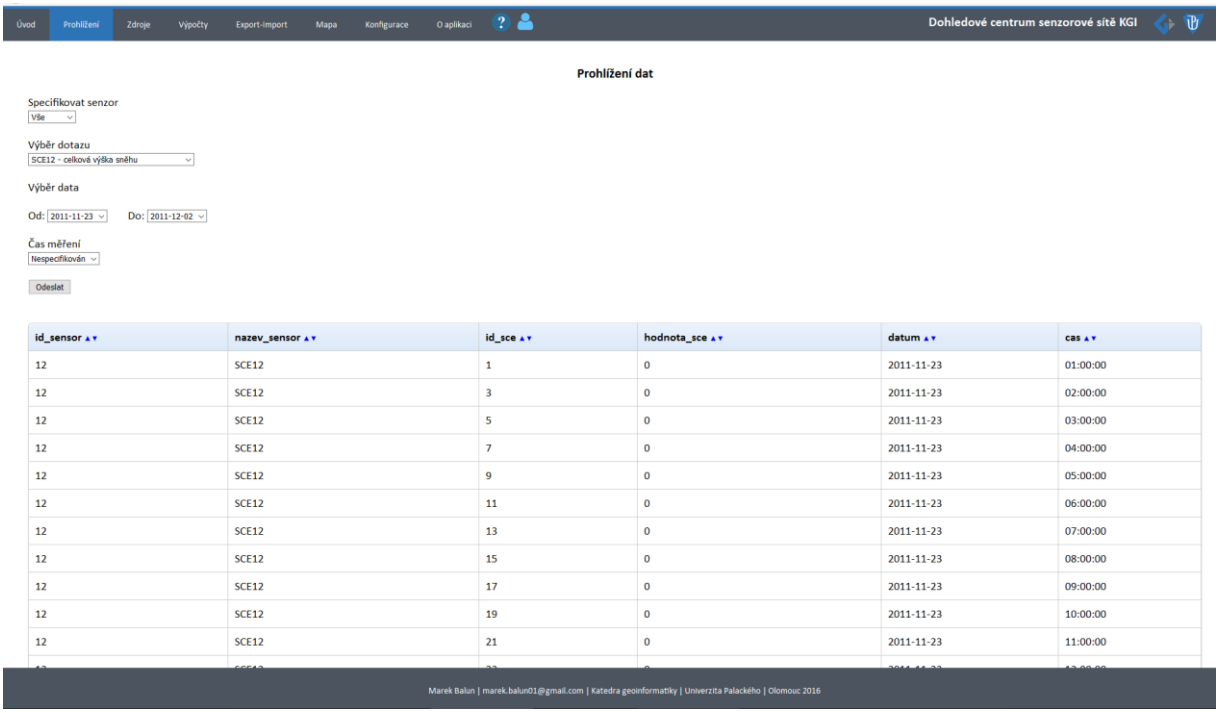
style – obsahuje CSS soubory s nadefinovanými kaskádovými styly



Obr. 7 Struktura kořenového adresáře aplikace (zdroj: vlastní zpracování)

4.4 Popis aplikace

Jak již bylo uvedeno Dohledové centrum senzorové sítě je webová aplikace. Pro její prohlížení postačuje webový browser, žádné další doplňky nejsou vyžadovány. Podporované prohlížeče jsou Microsoft Edge, Google Chrome, Mozilla Firefox, Opera, Vivaldi a Maxthon (v těchto prohlížečích byla aplikace testována a bez problému fungovala). Internet Explorer podporován není a jednotlivé stránky se v něm nemusejí zobrazovat korektně.



Obr. 8 Ukázka layoutu aplikace (zdroj: vlastní zpracování)

Layout aplikace

Layout aplikace se skládá z těla, hlavičky a patičky. Hlavička a patička je tvořena průběžnými lištami. Lišty mají tmavě šedou barvu a popisky na nich jsou bílé, naproti tomu tělo má bílé pozadí s černým písmem, barevné schéma lišt a těla je tedy inverzní. Nad horní lištou je ještě tenký modrý pruh a také ikony a loga na liště jsou laděny do modré barvy. Na horní liště je umístěna horizontální navigace, která obsahuje skryté rozbalovací podnabídky, jež jsou k dispozici po najetí kurzorem. Dále jsou na liště loga Univerzity Palackého a Katedry geoinformatiky, které slouží jako hypertextové odkazy na stránky těchto institucí, nadpis, ikona nápovědy a ikona která slouží jako indikátor toho jestli je uživatel přihlášený (zobrazuje se pouze přihlášenému uživateli – po najetí kurzorem se objeví popisek se jménem aktuálně přihlášeného uživatele). Na spodní liště je pouze tiráž – jméno autora, email autora, název instituce a rok. Horizontální navigace má několik záložek obsahujících podzáložky v rozbalovacích nabídkách. Tyto záložky jsou rozděleny do několika tematických sekcí.

Úvod – Obecné informace o aplikaci a diplomové práci.

Prohlížení – V této sekci je možné si prohlížet uživatelsky definované dotazy nad daty. Jednak v tabelární podobě - záložka *Prohlížení dat* a jednak v podobě spojnicových grafů – záložka *Prohlížení grafů*. V poslední záložce *Vlastní dotaz* lze zadat libovolný SQL dotaz, dotaz lze také uložit.

Zdroje – V této sekci je přehled všech připojených zdrojů k aplikaci. Jedná se o připojené databáze, registrované senzory, sítě, lokality a měřené veličiny. Přístupná pouze pro přihlášeného uživatele.

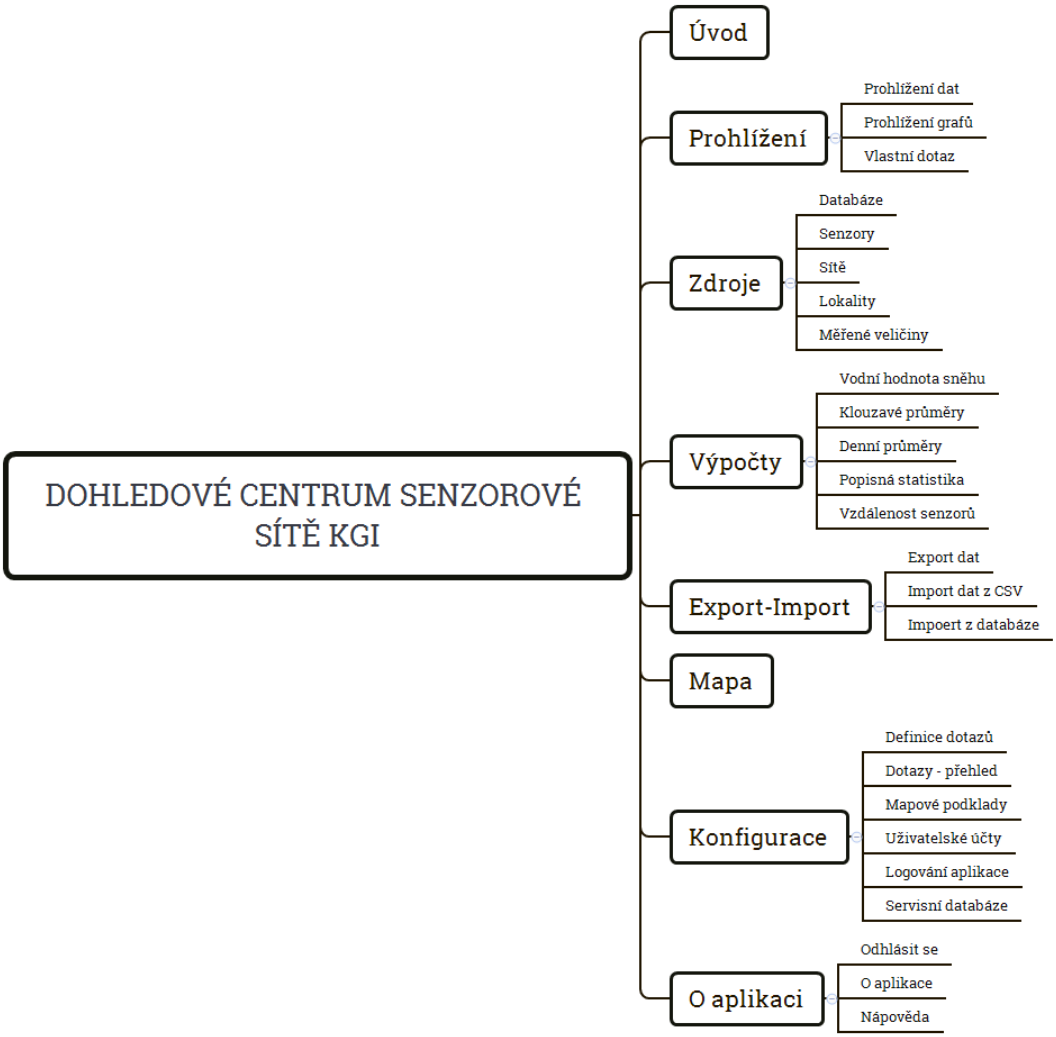
Výpočty – Tato sekce zahrnuje výpočet vodní hodnoty sněhu, výpočty denních průměrů měření, klouzavých průměrů za zvolené období, výpočet vzdálenosti senzorů a popisnou statistiku tabulek. Přístupná pouze pro přihlášeného uživatele.

Export-Import – Jak už název napovídá, v této sekci se nacházejí nástroje pro import a export dat. Jsou zde tři záložky, jedna pro *Export dat*, další dvě pro import, který je rozdělený na *Import dat z CSV* (soubor -> databáze) a *Import z databáze* (databáze -> databáze). Přístupná pouze pro přihlášeného uživatele.

Mapa – Vizualizace rozmístění jednotlivých senzorů a sítí v podobě mapového náhledu. Umožňuje také přidávat uživateli vlastní tematické vrstvy ve formátu JSON.

Konfigurace – Obsahuje všechna nastavení aplikace – *Uživatelské účty*, *Logování aplikace*, *Mapové podklady*, *Servisní databáze* a nachází se zde sekce pro definování uživatelských dotazů. Přístupná pouze pro přihlášeného uživatele.

O aplikaci – Zde představuje hlavní sekci *Nápověda*, dále je zde stránka pro odhlášení (pouze přihlášený uživatel, na liště nepřihlášeného uživatele je samostatná záložka *Přihlásit se*) a tiráž – informace o autorovi, kontaktní údaje.



Obr. 9 Mapa webových stránek aplikace (zdroj: vlastní zpracování)

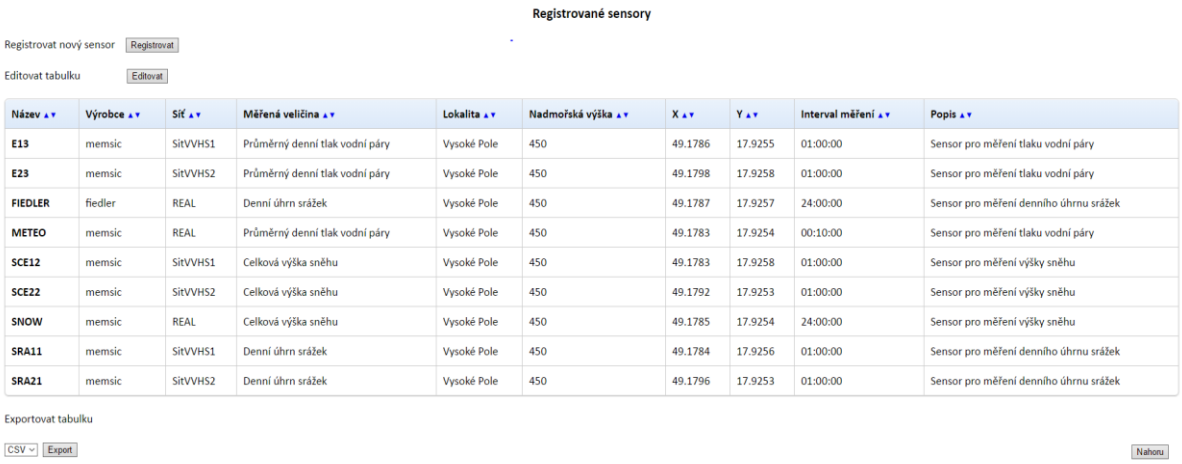
Zdroje

Zdroje aplikace tvoří připojené databáze (servisní databáze se ve výchozím stavu nezobrazuje mezi připojenými databázemi, pro uživatele je transparentní, pokud si ji explicitně nepřipojí), registr senzorů, registr sítí, seznam lokalit a seznam měřených veličin. Databáze reprezentují reálné aktuálně připojené datové úložiště. Ostatní zdroje jsou reprezentovány pouze entitami v servisní databázi. Slouží zde k vytváření přehledů, pro filtrování dotazů a výpočtů, se kterými jsou asociovány. Aplikace nepodporuje přímé připojení senzorů ani sítí. Všechny položky v sekci *Zdroje* může uživatel přidávat, odebírat nebo upravovat. Pro přidání nových položek jsou k dispozici vkládací formuláře. Úpravy a odstranění položek se dělají pomocí editace přehledové tabulky. Přidávání a úpravy zdrojů nejsou úplně volné, ale řídí se některými omezeními. Tyto omezení jsou vynuceny jednak integritními omezeními na úrovni databáze a jednak omezeními na aplikační úrovni pomocí omezených množin hodnot, které může uživatel vybrat nebo např. kontrolou jestli odstraněním položky nevzniknou osiřelé záznamy. Ve vztahu rodič – potomek je *Lokalita* rodičem pro *Sítě* a *Sítě* jsou rodičem pro *Senzor*. *Měřená veličina* je rodičem pro *Senzor*. Pro přidání senzoru, je potřeba nejdříve vytvořit objekt *Lokalita*, na které probíhá měření, pak je možné vytvořit objekt *Síť*, který lze asociovat s touto

lokalitou. Dále musí být vytvořený objekt *Měřené veličiny*, kterou senzor zaznamenává. Jsou-li tyto podmínky splněny, je možné přidat senzor, který se při vkládání asociuje s danou měřenou veličinou a sítí do které připojený. Přes objekt *Sítě* se senzor automaticky asociuje s objektem *Lokalita*, na které probíhá měření. S objektem *Senzor* jsou pak asociovány s dotazy. Při smazání senzoru jsou tyto dotazy rovněž automaticky smazány. Asociace mezi *Senzory* a dotazy již není zajištěna na úrovni databáze vazbami mezi tabulkami, ale pouze na aplikační úrovni.

Senzory

Senzory jsou v aplikaci reprezentovány pouze jako záznamy v databázi. Registr senzorů slouží jako přehled senzorů asociovaných s naměřenými daty. Senzory lze libovolně přidávat a mazat jsou-li splněny podmínky vyplívající z integritních omezení v databázi. Nový senzor lze přidat kliknutím na odkaz *Registrovat nový senzor* – který uživatele přesměruje na vkládací formulář. Jednotlivé pole ve formuláři mají omezené množiny hodnot, které lze zadat. Např. souřadnice X může nabývat hodnoty od -90 do 90. Smazání senzoru se provádí přes editaci tabulky. Po zaregistrování senzoru se v Mapě vytvoří špendlík na zadaných souřadnicích, zároveň je zaregistrovaný senzor možné asociovat s uživatelsky definovanými dotazy pro jejich filtrování. Senzor musí být asociován s konkrétní sítí a měřenou veličinou. Lokalita je mu automaticky přiřazena dle lokality, do které náleží síť. Měnit údaje o registrovaných senzorech je možné pomocí editace tabulky. Údaje o senzorech je možné exportovat do souborů CSV nebo do XML. Základní údaje o registrovaných senzorech jsou uloženy v tabulce *sensors,* další jsou načítány z tabulek *networks*, *locality* a *measurand* s kterými je tabulka *sensors* spojená přes *join.* Tabulka zobrazená v záložce *Senzory* je tvořená spojením právě těchto čtyř tabulek.



Název	Výrobce	Síť	Měřená veličina	Lokalita	Nadmořská výška	X	Y	Interval měření	Popis
E13	memsic	SitVVHS1	Průměrný denní tlak vodní páry	Vysoké Pole	450	49.1786	17.9255	01:00:00	Sensor pro měření tlaku vodní páry
E23	memsic	SitVVHS2	Průměrný denní tlak vodní páry	Vysoké Pole	450	49.1798	17.9258	01:00:00	Sensor pro měření tlaku vodní páry
FIEDLER	fiedler	REAL	Denní úhrn srážek	Vysoké Pole	450	49.1787	17.9257	24:00:00	Sensor pro měření denního úhrnu srážek
METEO	memsic	REAL	Průměrný denní tlak vodní páry	Vysoké Pole	450	49.1783	17.9254	00:10:00	Sensor pro měření tlaku vodní páry
SCE12	memsic	SitVVHS1	Celková výška sněhu	Vysoké Pole	450	49.1783	17.9258	01:00:00	Sensor pro měření výšky sněhu
SCE22	memsic	SitVVHS2	Celková výška sněhu	Vysoké Pole	450	49.1792	17.9253	01:00:00	Sensor pro měření výšky sněhu
SNOW	memsic	REAL	Celková výška sněhu	Vysoké Pole	450	49.1785	17.9254	24:00:00	Sensor pro měření výšky sněhu
SRA11	memsic	SitVVHS1	Denní úhrn srážek	Vysoké Pole	450	49.1784	17.9256	01:00:00	Sensor pro měření denního úhrnu srážek
SRA21	memsic	SitVVHS2	Denní úhrn srážek	Vysoké Pole	450	49.1796	17.9253	01:00:00	Sensor pro měření denního úhrnu srážek

Obr. 10 Registr senzorů (zdroj: vlastní zpracování)

Měřené veličiny

Tato záložka obsahuje tabulku měřených veličin. V tabulce jsou registrovány všechny měřené veličiny včetně použitých jednotek. Zároveň zde uživatel vidí všechny senzory měřící danou veličinu. Při registraci senzoru musí uživatel nový senzor asociovat s některou veličinou z tohoto seznamu. Pokud na seznamu daná veličina chybí je potřeba ji nejdříve doplnit, a pak až zaregistrovat senzor zaznamenávající tuto veličinu. Měřenou veličinu lze smazat, jen pokud s ní není asociován žádný senzor. Měřené veličiny jsou

uloženy v tabulce *measurand*. Tabulky *measurand* a *sensors* jsou propojeny cizím klíčem, vztah 1:N. Měřené veličiny jsou využívány v grafech a výpočtech, ke kterým se na základě asociace senzoru s měřenou veličinou automaticky přiřadí jednotky a názvy těchto měřených veličin.

Lokality

Záložka obsahuje přehled lokalit, na kterých jsou umístěny jednotlivé sítě a senzory. Při přidávání nové sítě je potřeba tuto síť asociovat s lokalitou, pokud lokalita zatím není evidována, je potřeba ji nejdříve přidat. Senzory, které budou následně přidány do sítě se automaticky asociují s lokalitou, která je spojena s danou sítí. Lokality jsou jsou využívány při výpočtech obdobně jako měřené veličiny.

Databáze

Od počátku bylo snahou udělat univerzální aplikaci, která bude moci obsluhovat jakoukoli PostgreSQL databázi. Databáze lze proto v aplikaci libovolně připojovat a odpojovat. Pro připojení nové databáze je k dispozici jednoduchý formulář, kde se zadávají přihlašovací údaje, název připojení (ten je pouze pro potřeby aplikace, slouží jako identifikátor připojení a musí být proto unikátní) a volitelně popis připojení. Lze ještě vybrat jestli databáze bude pouze pro import. S databází pro import nelze destruktivně manipulovat, nelze nad ní dělat žádné výpočty ani exportovat data. Je možné pouze prohlížet a tabulky importovat z ní data do jiné databáze. Připojená databáze může být hostovaná jak na *localhostu* (na *localhostu* z pohledu aplikace tj. na serveru který hostuje aplikaci) tak na vzdáleném serveru. V případě *localhostu* s připojením databáze není problém. Pokud chceme připojit databázi ze vzdáleného serveru, musíme toto připojení nejdříve povolit v konfiguračním souboru *pg_hba.conf* a také povolit port na kterém databáze naslouchá na *firewallu* nebo v *iptables* (jsou-li aktivní).

Obr. 11 Vyplněný formulář pro připojení databáze (zdroj: vlastní zpracování)

V záložce *Databáze* v sekci *Zdroje* je přehled a rozcestník připojených databází. Připojené databáze jsou zobrazeny v tabulce, ve které jsou parametry připojení k těmto databázím, popis, informaci jestli databázi lze editovat a seznam tabulek. V seznamu tabulek jsou vidět tabulky ze všech schémat v databázi (kromě systémových tabulek, jež jsou součástí katalogů – mají prefix *pg_**, a tabulky obsahující souřadnicové referenční systémy *spatial_ref_sys*), ale aplikace umí pracovat pouze s výchozím schématem *Public*. Seznam aktuálně připojených databází lze exportovat do XML a CSV. Po kliknutí na název databáze se zobrazí detailnější informace o dané databázi, seznam tabulek s názvy jednotlivých sloupců a datovými typy. Na této stránce lze změnit popis k databázi, odpojit databázi nebo smazat tabulku.

```
function dbConnect($dbName)
{
    include('pages/connections.php');

    $queryDB = pg_query($dbControlCenterMgmt, 'select "hostname", "database", "user", "port", convert_from(decrypt(password, \'cypherKey\', \'aes\'), \'utf-8\') as pwd from "databases" where connectionname = \''.$dbName.'\'');

    $row = pg_fetch_assoc($queryDB);

    $hostname = $row['hostname'];
    $database = $row['database'];
    $user = $row['user'];
    $port = $row['port'];
    $password = $row['pwd'];

    $dbSession = pg_connect("host=$hostname dbname=$database port=$port user=$user password=$password");

   if($dbSession)
   {
       return $dbSession;
   }
    else
   {
      header('Location:./index.php?page=unreachabledb&dbName='.$dbName);
   }
}
```

Funkce, která po předání názvu databáze (názvu připojení) vrací připojení k této databázi. Pokud se připojení nezdaří, je uživatel přesměrován na rozcestník, kde musí rozhodnout jak naložit s nedostupnou databází – přesunout na seznam ignorovaných nebo smazat přihlašovací údaje k databázi.

Ignorované databáze

V případě že se aplikaci nepodaří připojit k databázi, jejíž přihlašovací údaje jsou v aplikaci již uloženy, je uživatel vyzván, aby zvolil další postup. Na výběr jsou dvě možnosti - přihlašovací údaje permanentně odstranit (při odstranění databáze jsou trvale odstraněny i dotazy asociované s touto databází) nebo přesunout databázi do seznamu ignorovaných např. v případě že je nedostupnost pouze dočasná. K ignorované databázi se již aplikace nepokouší znovu připojit. S touto databází není možné žádným způsobem manipulovat ani zobrazovat data, ale přihlašovací údaje a dotazy asociované s databází zůstanou uloženy. Seznam ignorovaných databází je realizován parametrem *ignored* v tabulce *databases.* Parametr nabývá dvou hodnot (0,1) – 0 funkční databáze, 1 ignorovaná databáze. Pokud je databáze opět dostupná lze ji ze seznamu vyjmout. Při vyjmutí se seznamu se aplikace pokusí k databázi znovu připojit, pokud je pokus úspěšný je změněna hodnota parametru *ignored* na 0, pokud ne, je uživateli ještě nabídnuta možnost změnit přihlašovací údaje. Změna se provádí v detailním přehledu databáze, do kterého se lze dostat z *Prohlížení/Databáze* po kliknutí na název databáze. Přesunutím databáze na seznam ignorovaných se předchází chybovým stavům, způsobeným neúspěšnými pokusy o připojení.

Servisní databáze

Servisní databáze uchovává konfiguraci aplikace. Obsahuje následující položky:

- Údaje o ostatních připojených databázích včetně hesel
- Uživatelské účty včetně hesel
- Registr senzorů a sítí
- Registr lokalit a měřených veličin
- Údaje o mapových podkladech ve formátu GeoJSON
- Předefinované dotazy

Výchozí servisní databáze se jmenuje *ControlCenterMgmtDB*. Servisní databáze není „napevno“ připojená ve zdrojovém kódu aplikace. Její přihlašovací údaje jsou uloženy v konfiguračním souboru *conf.xml*. V případě že bude databáze umístěna na jiném serveru je možné tyto přihlašovací údaje změnit – buď přímo editací souboru *conf.xml* nebo v grafickém rozhraní v aplikace v záložce *Konfigurace/Servisní databáze*.

Všechna hesla uložená v databázi jsou zašifrovaná pomocí algoritmu AES. Pro šifrování bylo využito rozšíření PostgreSQL *pgcrypto*. Jediné heslo, které je uloženo v *plain textu*, je heslo výchozího administrátorského účtu. Toto heslo je uloženo v souboru *conf.xml*. Důvodem je zajištění funkčnosti tohoto účtu i v případě kdy nebude připojená servisní databáze. V případě odpojení servisní databáze se aplikace uzamkne, přístupný je pouze formulář pro opětovné připojení databáze. Pro autorizaci připojení k servisní databázi jsou právě vyžadovány přihlašovací údaje výchozího administrátora (je potřeba zadat dvoje přihlašovací údaje 1) k servisní databázi – autentizace vůči databázovému serveru 2) výchozího administrátora – autentizace vůči aplikaci).

Servisní databáze se skládá z následujících třinácti tabulek:

charts – Tabulka obsahuje nadefinované parametry pro vykreslování grafů.

customquerys – Tabulka obsahuje uživatelsky definované SQL dotazy. Dotaz je zde uložen jako textový řetězec. Obsahuje-li řetězec apostrofy, jsou při ukládání do

tabulky nahrazeny tildou. Při načítání dotazu jsou tildy opět překódovány na apostrofy. Důvodem je problematické zpracování – hodnoty jsou při zpracování skriptem PHP načítány do proměnných. Pokud se uvnitř proměnné (v dotazu) vyskytuje apostrof, musí být zapsán jako *escape* sekvence, s touto *escape* sekvencí je potom uložen i do databáze. Při opětovném načtení dotazu už ale *escape* sekvence není správně rozpoznána a interpretována jako apostrof. Místo toho je načtena jako posloupnost dvou znaků - *backslash* a apostrofu.

databases – Tato tabulka obsahuje názvy připojení k databázím (v aplikaci mají funkci identifikátorů jednotlivých připojení) a přihlašovací údaje k jednotlivým databázím včetně šifrovaných hesel. Díky tomu, že jsou hesla uložena v databázi, není je potřeba opětovně zadávat. Tabulka dále obsahuje informace, o tom jestli je databáze na seznamu ignorovaných a jestli je databázi možné editovat.

dayaverage – Tabulka obsahuje definice pro výpočty denních průměrů.

layers – Obsahuje informace o připojených datových vrstvách ve formátu GeoJSON.

locality – Tabulka obsahující seznam lokalit a jejich souřadnic. Tabulka má vazbu 1:N k tabulce *networks*.

measurand – Tabulka obsahuje seznam měřených veličin a jednotek. Je propojena vazbou 1:N s tabulkou *sensors*.

movingaverage – Tabulka obsahuje definice pro výpočty klouzavých průměrů.

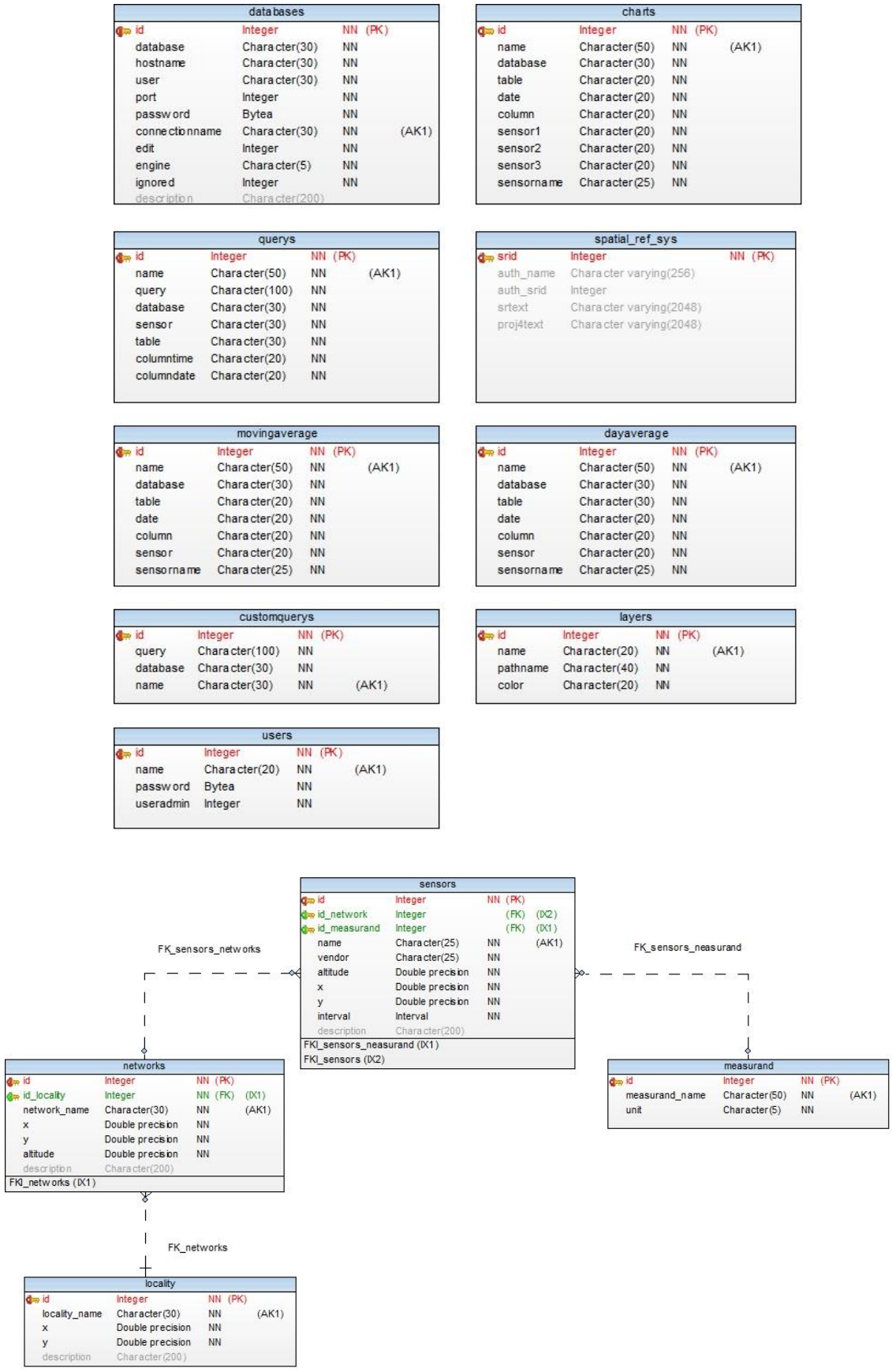
networks – Tabulka obsahující údaje o registrovaných senzorových sítích. Tabulka je přes cizí klíče propojena vazbami 1:N s tabulkou *locality* a s tabulkou *sensors* .

querys – Tabulka obsahuje nadefinované dotazy pro *Prohížení dat*.

sensors – Obsahuje informace o registrovaných senzorech. Tabulka je propojena vazbami 1:N s tabulkou *networks* a s tabulkou *measurand*.

users – V této tabulce jsou evidovány jména uživatelů, šifrovaná hesla a informace jestli má uživatel oprávnění spravovat další uživatelské účty. Nejsou zde údaje výchozího administrátora.

spatial_ref_sys – Tabulka obsahující souřadnicové referenční systémy. Automaticky se v databázi vytvoří po přidání rozšíření *postgis*.



Obr. 12 Schéma servisní databáze . PK – primární klíč, FK – cizí klíč, AK – omezení jedinečností, NN – nesmí být NULL (zdroj: vlastní zpracování)

Prohlížení tabulek

Po kliknutí na název tabulky v záložce *Databáze* se zobrazí detail vybrané tabulky. Výchozí řazení záznamů v tabulce je vzestupně podle sloupce na pozici číslo jedna. Uživatel může řazení změnit kliknutím na šipky, které se nacházejí v hlavičce tabulky za názvem sloupce. Po kliknutí na šipku se stránka znovu načte s novými parametry předanými metodou GET, které modifikují SQL dotaz na pozadí. Pro vyhledávání v tabulce je možné použít jednoduchý filtr, který umožňuje zadat jeden parametr vyhledávání. Filtr se skládá ze tří polí. První pole je rolovací menu, ve kterém uživatel z nabídky vybere název sloupce. Druhé pole je rovněž rolovací menu, ve kterém uživatel vybere operátor. Pro sloupce s datovým typem *character* jsou na výběr operátory *rovná se*, *nerovná se* a operátor *LIKE*. Pro sloupce s číselnými datovými typy jsou na výběr všechny porovnávací operátory. Poslední pole je textový vstup, kam uživatel napíše hodnotu, se kterou chce porovnávat. Filtr je aktivní dokud uživatel neklikne na tlačítko *Zrušit výběr* nebo neopustí stránku, změnou řazení ani přejitím do režimu editace se výběr nezruší.

Editace tabulek

Aplikace podporuje editaci tabulek u všech připojených databází (vyjma těch u kterých uživatel vybral možnost jen pro import) ze schématu *Public*. Při editaci tabulky je buď možné upravovat jednotlivé záznamy, nebo je mazat. Uživatel si při editace musí být vědom integritních omezení dané databáze a tabulky – pokud se např. bude pokoušet nastavit kolizní hodnoty primárního klíče, editace řádku se nezdaří. Editace neumožňuje přidávání nových záznamů, to je řešeno pomocí importů. V aplikaci, která pracuje s telemetrickými daty ze senzorů ruční vkládání jednotlivých záznamů ani nedává příliš smysl. Pomocí editace tabulek se neupravují jen tabulky z připojených databází, ale i registr senzorů, sítí a další systémové zdroje. V jejich případě se ale často nejedná o jednu tabulku, ale spojení více tabulek (např. tabulka v registru senzorů vznikla spojením 4 tabulek) a na rozdíl od tabulek z připojených databází je možné do nich přidávat nové záznamy prostřednictvím formulářů. Pro editaci tabulky je potřeba se přepnout do režimu editace. V tomto režimu se přidají na začátek tabulky dva sloupce. První sloupec obsahuje tlačítko, které aktivuje editaci daného řádku – v buňkách se objeví pole pro vstup textu, ve kterých je možné přepsat původní hodnoty (u editace tabulek systémových zdrojů jsou v některých buňkách rolovací nabídky ve kterých je omezené množství možností – to je kvůli dodržení integritních omezení servisní databáze). V druhém sloupci je pak tlačítko pro smazání řádku. Řádky se mažou ihned po kliknutí na tlačítko *Smazat* – bez dalšího potvrzování. Po editaci řádku je potřeba jej uložit kliknutím na tlačítko *Uložit*, aby změna zapsala. To možná vypadá trochu nelogicky – smazání proběhne ihned, editaci je potřeba potvrdit – je to z technických důvodů. Smazání řádku je jednodušší operace než jeho úprava, při mazání řádků se odesílají hodnoty pouze jednou, při editaci je potřeba odeslat původní hodnoty plus nové hodnoty vložené uživatelem. Původní záměr byl, aby bylo možno editovat celou tabulku najednou. Při spuštění režimu editace se ve všech řádcích objevily pole pro vstup textu. Hodnota každé buňky byla odeslána jako proměnná. Toto řešení bylo funkční jen u malých tabulek. U větších tabulek nastala chyba, protože PHP ve výchozím nastavení může v rámci jedné *session* alokovat maximálně 1000 proměnných, což už tabulka která má sto řádků a pět sloupců překračuje (za každou buňku jsou odeslány dvě proměnné). Tento limit lze sice zvednout, ale pro tabulky s desítkami tisíc záznamů by to nebylo řešením. Další možnost je hodnoty buněk načítat místo do jednoduchých proměnných do polí. Velikost polí sice není omezena, ale je omezeno množství paměti, které může PHP skript alokovat – ve výchozím nastavení

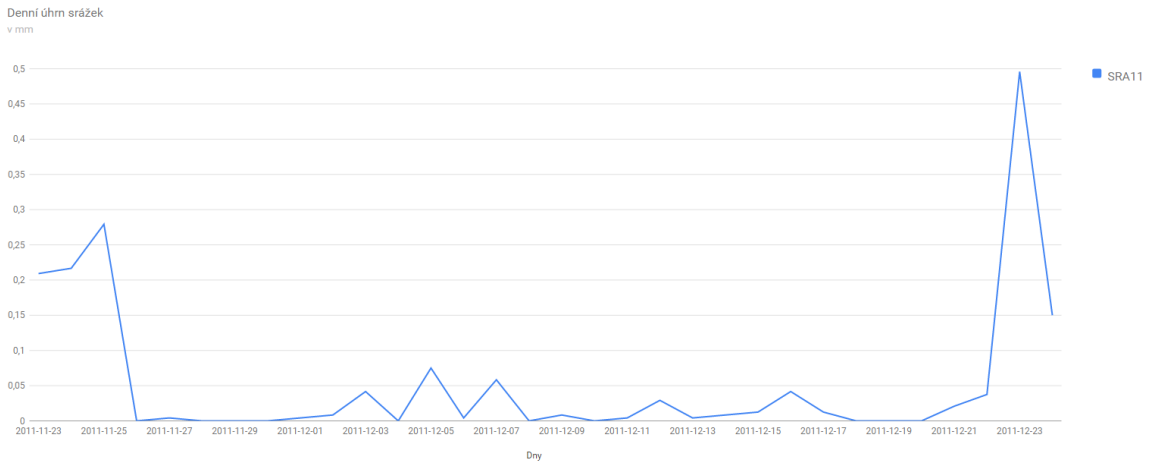
128MB. U tabulek s velkým množstvím záznamů (které lze u telemetrických dat předpokládat) by opět hrozilo, že bude paměť „spotřebována“. Její navyšování a řešení problémů hrubou silou není principálně příliš správné. Proto byla celá koncepce upravena a editace probíhá po jednotlivých řádcích, které si uživatel vybere.

Prohlížení dat

V záložce prohlížení dat jsou vizualizována data v podobě tabulek. Na rozdíl od prohlížení tabulek ze sekce *Databáze* zde nejsou celé tabulky ve výchozím stavu, ale již výsledky předdefinovaných a parametrizovaných dotazů. Uživatel zde nevidí strukturu databáze a tabulek ale pouze dotazy. Jednotlivé dotazy už mají „napevno“ nastaveny filtry (např. podle názvu senzoru), volitelně si uživatel může nastavit filtrování dle času měření a data měření. Dotazy lze třídit dle senzorů, se kterými jsou asociovány. I když může výsledek dotazu obsahovat data naměřená více senzory (pokud si ho uživatel takto nadefinuje), tak asociovat lze dotaz pouze s jedním senzorem. Prohlížení dat je přístupné i pro nepřihlášené uživatele.

Grafy

Aplikace umožňuje vizualizovat obsah tabulky v podobě spojnicového grafu, kde na ose X je datum měření a na ose Y naměřené hodnoty. Graf musí být vždy vztažen k jedné tabulce, nelze zobrazovat v jednom grafu hodnoty z více tabulek. V jednom grafu lze zobrazit zároveň údaje ze třech senzorů – tři linie ve spojnicovém grafu, data musí být ale uložena v jedné tabulce a musí jít o stejnou veličinu. Graf lze vykreslit pro rozpětí mezi dvěma daty měření, mezní hodnoty jsou první poslední den měření. Pokud je pro jeden den k dispozici více naměřených hodnot, je z nich vytvořen průměr a tato průměrná hodnota je pak zobrazena v grafu. Ve výchozím stavu je v aplikaci k dispozici set ukázkových grafů z databáze *vodni_hodnota_snehu*. Další grafy nad vlastními daty si uživatel musí nadefinovat v sekci *Definice dotazů*. Grafy jsou vytvořeny pomocí Google Chart API.



Obr. 13 Ukázka grafu pro Denní úhrn srážek (zdroj: vlastní zpracování)

Export dat

Ze všech databází, které mají povolenou editaci je v sekci *Export dat* možné exportovat data do podporovaných formátů. Mezi základní podporované formáty patří SQL, CSV, XML, dBase (soubory s příponou *.dbf*) a JSON. Tabulky které obsahují sloupec s datovým typem *geometry* je možné navíc exportovat do formátů pro prostorová data. Na výběr je Esri Shapefile, GML, KML a GeoJSON. Data lze exportovat jen jako celé tabulky, výběr v této sekci není podporován. Exportovat výběr je možné v sekci *Prohlížení/Prohlížení dat*, na výběr jsou zde formáty XML a CSV. Pro export dat do GML, KML, GeoJSON a SHP je využívána knihovna GDAL/OGR verze 1.9. Shapefile je exportován jako zip archiv, který obsahuje soubory s příponami *.shp .shx .prj a .dbf*. Pro export do dBase je využívána knihovna *php_dbase.dll* kterou je potřeba vložit do složky *ext* v kořenovém adresáři PHP a povolit tuto extenzi v souboru *php.ini* – do části *Dynamic Extensions* se vloží záznam *extension=php_dbase.dll*. Pro export do SQL je využívána utilita *pg_dump*, která je součástí standartní instalace PostgreSQL serveru. K *pg_dump* je potřeba nastavit absolutní cestu v konfiguračním souboru, stejně tak je potřeba nastavit absolutní cestu ke složce *export,* do které je jako mezikrok uložen soubor před odesláním. Na závěr je ještě potřeba na serveru, ze kterého chceme data exportovat v souboru *pg_hba.conf* povolit připojení - hodnotu změnit na *trust*. V případě že se jedná o databázi na *localhostu* nastavit *trust* na IP adresu 127.0.0.1, v případě, že databáze běží na vzdáleném serveru, nastavit trust na FQDN nebo IP adresu serveru hostujícího aplikaci. Export dat do SQL ze vzdáleného serveru je funkční jen pokud je na něm nainstalována stejná verze PostgreSQL (stejná verze *pg_dump*) jako na serveru, který hostuje aplikaci. V případě rozdílných verzí je export nefunkční. Export do ostatních formátů – XML, CSV a JSON je realizován vestavěnými funkcemi PHP. Export dat probíhá ve dvou krocích, v prvním kroku je vytvořen soubor na serveru ve složce *export*, v druhém kroku je pak odeslán uživateli. Po odeslání je soubor ze složky *export* smazán, aby nedocházelo k neúměrnému růstu velikosti této složky.

Import dat

Import dat je možný jednak ze souborového formátu CSV a jednak z databáze (na import dat z databáze lze zároveň pohlížet i jako na export – jedná se o přesun dat mezi dvěma databázemi, která bude zdrojová a cílová zvolí uživatel). Podporován je databázový systém PostgreSQL. Původně bylo v plánu zahrnout i systém MySQL, ale ten se do finální verze nedostal. Import je realizován skrze sadu formulářů, kde si uživatel vybere data, která chce importovat a jejich cílovou destinaci – databázi a tabulku. Uživatel má na výběr mezi importem do existující tabulky nebo vytvořením nové tabulky. Při vytváření nové tabulky si může zvolit názvy sloupců, jestli může sloupec obsahovat hodnotu *NULL* a který sloupec (sloupce) bude primárním klíčem. Má ještě možnost zvolit, aby se daný sloupec neimportoval. Datový typ sloupců je automaticky odvozen ze sloupců původní tabulky a nelze jej změnit. Při importu do existující tabulky pouze uživatel přiřadí odpovídající si sloupce, případně zvolí možnost neimportovat daný sloupec. Při importu ze souborového formátu CSV je k dispozici jen možnost importu do existující tabulky. Před samotným importem je možnost si importovaná data prohlédnout v samostatné tabulce. Import z CSV je realizován pomocí SQL příkazu *COPY*. Při importu z CSV, je importovaný soubor nejdříve nahrán na server do složky *import*, odkud je následně importován. Import z databáze byl původně řešen pomocí funkce *dblink*. Ta ale vyžadovala mít v databázi přidané rozšíření *dblink*, což nelze a priori očekávat u každé databáze a navíc by byl problematický import do existujících tabulek, proto bylo od tohoto řešení upuštěno. Import je tedy řešen v několika krocích. Při importu do nové tabulky, je v prvním kroku

vytvořena prázdná tabulka, resp. nejdříve je z odeslaných proměnných složen dotaz který tuto tabulku vytvoří. V dalším kroku je složena dvojice SQL dotazů, které přenášejí data řádek po řádku – jeden čte data ze zdrojové databáze, druhý je zapisuje do cílové databáze.

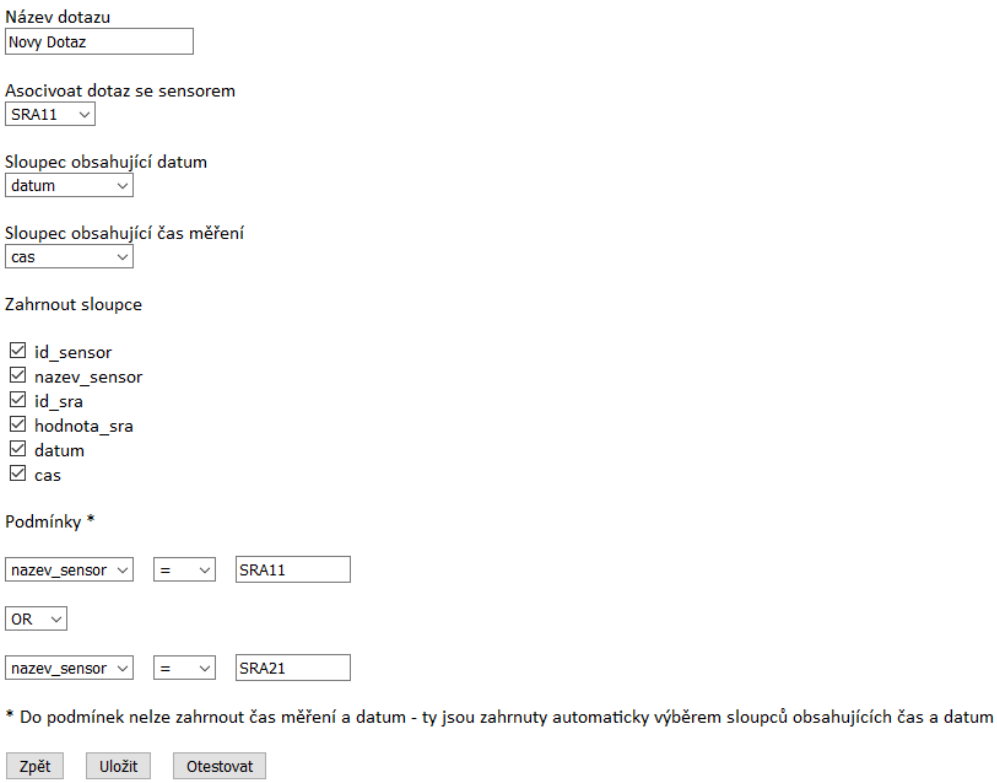
Výpočty

Sekce *Výpočty* se skládá z pěti částí. První z nich představuje výpočet vodní hodnoty sněhu přejatý z diplomové práce Libora Kimpla *Standardy pro dohledové centrum senzorové sítě (2012).* Vodní hodnota sněhu se stanovuje pro konkrétní datum na základě hodnot naměřených senzory. Vstupní veličiny pro výpočet jsou úhrn srážek, celková výška sněhu, výška nového sněhu a průměrný denní tlak vodní páry. Tento výpočet předpokládá použití konkrétních senzorů a je vázán na ukázkovou databázi *vodni_hodnota_snehu*, která je rovněž vytvořena Liborem Kimplem. Pokud by uživatel chtěl spočítat vodní hodnotu sněhu pro jiná data, musí použít jako šablonu tuto databázi. Další čtyři výpočty lze provádět nad libovolnou databází. Výpočet klouzavého průměru vypočítá průměrnou hodnotu z hodnot naměřených mezi dvěma zvolenými daty. Denní průměr spočítá průměrnou hodnotu z hodnot naměřených vybraným senzorem během jednoho dne. Výpočty klouzavého průměru a denních průměrů je potřeba nejdříve nadefinovat pro konkrétní tabulku a konkrétní senzory – je potřeba vybrat sloupec obsahující datum, sloupec obsahující naměřené hodnoty a sloupec s názvem senzoru. Dalším výpočtem je výpočet (vzdušné) vzdálenosti senzorů. Uživatel zvolí dvojici senzorů a jednotku, na kterou bude vzdálenost přepočítána. Vzdálenost je počítána na základě souřadnic, které uživatel zadal při registraci senzoru. Pro výpočet je použita funkce *st_distance()* z PostGIS. Poslední položkou v sekci výpočty je popisná statistika. V tomto případě se nejedná o výpočet ale souborné informace o vybrané tabulce. Tyto informace zahrnují počet řádků a sloupců, který sloupec (nebo sloupce) jsou primárním klíčem, a datové typy sloupců. Další informace jsou závislé na datovém typu daného sloupce. U číselných datových typů je zobrazena nejnižší a nejvyšší hodnota, suma a průměrná hodnota sloupce. U datového typu *character* je nejfrekventovanější hodnota a počet výskytů této hodnoty. U typu *date* je to nejnižší a nejvyšší datum – rozpětí, stejně tak datového typu *time*. U datového typu *geometry* je to výčet typů geometrie – bod, polygon nebo linie.

Definice dotazů

Jelikož ve výchozím stavu jsou k dispozici jen dotazy nad daty z ukázkové databáze, je umožněno uživateli nadefinovat si vlastní dotazy nad databází, kterou si připojí. Slovo dotazy může být možná trochu zavádějící, protože se v této sekce definují i grafy. Označení dotazy souvisí s tím, že na pozadí je vždy SQL dotaz (nebo více SQL dotazů) do databáze, i když je pro uživatele transparentní. Pro definování dotazu není potřeba znalost jazyka SQL, provádí se prostřednictvím formulářů, ve kterých se pouze vyberou názvy sloupců, případně hodnoty. Tvorba nového dotazu probíhá ve dvou krocích. V prvním kroku uživatel zvolí databázi, tabulku a typ dotazu který chce vytvořit. Na výběr je definování grafu, definování tabulky – prohlížení dat a definice výpočtů klouzavého průměru a denních průměrů. V druhém kroku už se pak vyberou jen konkrétní sloupce, kterých se bude dotaz týkat a doplní se název dotazu, pod kterým se uloží. Názvy databáze, tabulky a sloupců se při odeslání formuláře doplní do kostry SQL dotazu, který je uložen do příslušné tabulky uchovávající daný typ dotazu. Před uložením dotazu je možno jej ještě otestovat – přímo pod formulářem je vytvořen výstup z daného dotazu – tabulka, graf nebo výsledky výpočtu. U definování tabulky (*Prohlížení dat*) je formulář trochu složitější.

Uživatel si zde může vybrat sloupce, které budou zobrazeny a omezit dotaz dvěma podmínkami. Mezi podmínkami může být logický operátor OR nebo AND. Do podmínek nejdou zahrnout sloupce datum a čas, ty jsou zahrnuty automaticky. Přehled všech dotazů je zobrazen v sekci *Dotazy – přehled*. U každého dotazu je zároveň uvedena informace o databázi a tabulce, se kterou jsou spojeny. Zde lze dotazy mazat. Ještě je zde sekce *Vlastní dotaz*, určená pro pokročilejší uživatele kde je možné zapsat dotaz přímo v jazyce SQL. Tento dotaz je možné i uložit a při znovunačtení upravit. Při odpojení databáze se všechny dotazy asociované s touto databází smažou.



Obr. 14 Formulář pro definování nového dotazu (zdroj: vlastní zpracování)

Uživatelské účty

V aplikaci je možno pracovat ve dvou základních režimech - jako přihlášený uživatel nebo jako anonymní uživatel. Anonymní uživatel si může zobrazovat pouze předdefinované dotazy, grafy a prohlížet si mapu. Exportovat může pouze výběry z předdefinovaných v sekci dotazů *Prohlížení dat* do formátů XML a CSV. Ostatní operace s databází pro něj nejsou povoleny. Nemůže si prohlížet seznam připojených databází ani jejich vnitřní strukturu. Přihlášený uživatel naproti tomu může využívat všechny funkce aplikace. Může připojovat/odpojovat databáze, importovat nová data, exportovat data z libovolné tabulky do všech podporovaných formátů, definovat nové dotazy a má přístup do sekce *Výpočty*. Přihlášený uživatel má také přístup ke konfiguraci aplikace. Kromě přihlášených uživatelů je zde ještě uživatel s oprávněním spravovat ostatní uživatelské účty. Tento uživatel může vytvářet a mazat účty a přidělovat/odebírat oprávnění správy uživatelských účtů. Zvláštním případem je výchozí administrátorský účet. Tento účet nelze smazat ani mu odebrat právo spravovat ostatní účty. Přihlašovací údaje k výchozímu účtu by měl mít pouze správce serveru.

Uživatelské účty se spravují v záložce *Konfigurace/Uživatelské účty*. Pokud přihlášený uživatel nemá přidělené právo spravovat uživatelské účty, může si zde pouze změnit své heslo. Uživatel, který má toto právo přidělené, zde vidí tabulku všech uživatelský účtů. V tabulce může účty mazat nebo jim přidělit právo správy uživatelských účtů. Uživatelské účty kromě výchozího administrátorského účtu jsou uloženy v databázi v tabulce *users*. Přihlašovací údaje výchozího administrátora jsou uloženy v konfiguračním souboru *conf.xml*. Výchozí přihlašovací jméno je *admin* – lze jej stejně jako heslo změnit v konfiguračním souboru.

Logování

Aplikace si ukládá vlastní logy do složky *logs* která se nachází v kořenovém adresáři aplikace. Složka obsahuje tři podsložky *access_log*, *error_log* a *transact_log* do kterých se ukládají logy stejných názvů. Logy se ukládají do textových souborů (soubory s příponou *.txt*). Logování aplikace se konfiguruje v záložce *Konfigurace/Logování aplikace*. U *Error logu* je možné zvolit jestli se budou chybové hlášky zobrazovat přímo v aplikaci (vývojářský režim) a jestli se mají logy ukládat do souboru *error.log.txt*. Dále je možné zvolit, jestli se bude v názvu souboru ukládat datum – pokud ne, budou se všechny logy ukládat do jednoho souboru, pokud ano pro každý den kdy bude odesláno chybové hlášení, se vytvoří nový soubor. U *Access logu* a *Transact Logu* je možné zvolit pouze to jestli budou logy ukládány do souboru nebo ne. Zobrazování chybových hlášení přímo v aplikaci je ve výchozím stavu vypnuto, ostatní nastavení jsou ve výchozím stavu povolena.

Error logy – Jedná se o standartní logování chyb PHP. Logy se pouze ukládají do samostatné složky, aby je bylo možné snadno odlišit od chybových hlášení z jiných webů běžících na témže serveru. Standardně je lze najít ve složce *./php/logs/php_error_log*.

```
ini_set('log_errors', 1);                 //ukládat logy do souboru, 1 – ano, 0 – ne
ini_set('error_log', $fileName . '.txt');  //cesta k souboru + jméno souboru
ini_set('display_errors', 1);             //zobrazovat chybové hlášení v aplikaci, 1 – ano, 0 – ne
```

Část zdrojového kódu, která mění soubor pro ukládání chybových hlášení aplikace

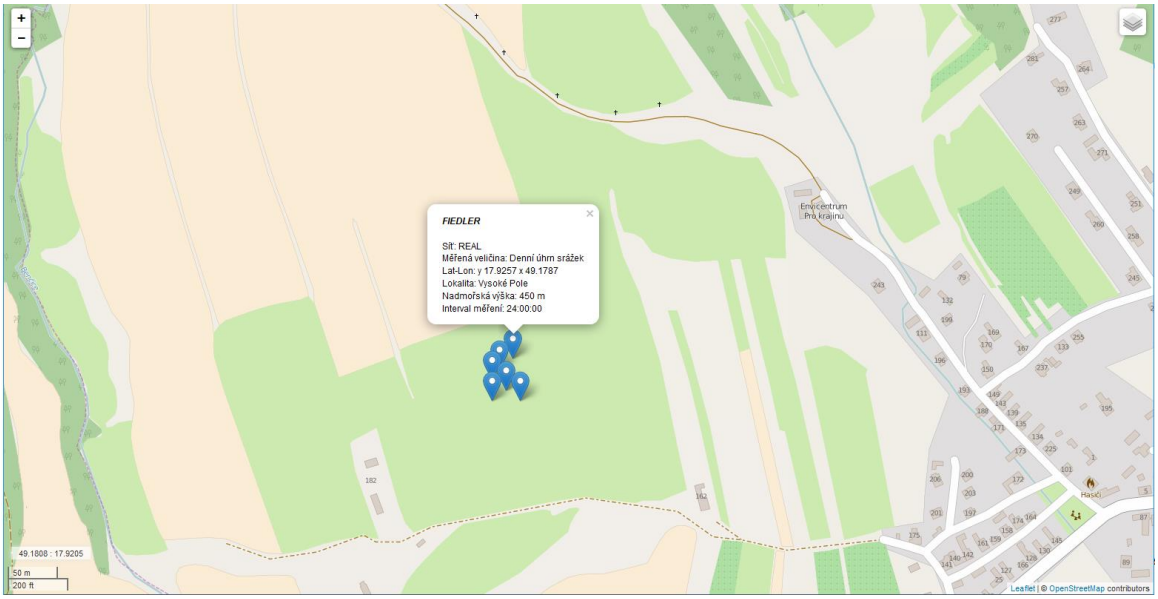
Access logy – V *access logu* je monitorováno přihlašování k aplikaci. Ukládá jméno uživatele, datum a čas přihlášení a odhlášení (pokud se uživatel korektně odhlásí). Do logu jsou také ukládány neúspěšné pokusy o přihlášení. Apache má také *access.log*, ten ale na rozdíl od logu aplikace monitoruje přístupy k jednotlivým stránkám, návratové stavové kódy a použitou metodu - GET/POST.

Transakční logy – transakční logy zaznamenávají jakoukoli manipulaci databázemi – import a export dat, editaci, mazání záznamů a tabulek. V logu se uloží jméno uživatelského účtu, datum a čas změny, typ transakce a databáze, nad kterou byla operace provedena.

Mapa

Spíše než o mapu se jedná o mapový náhled, který slouží pro vizualizaci polohy senzorů a sítí v terénu. Ty jsou zobrazeny jako špendlíky na souřadnicích, které uživatel zadal při registraci nového senzoru nebo sítě. Po kliknutí na špendlík se zobrazí *popup* okno

s informacemi o daném objektu jako je nadmořská výška, souřadnice, název lokality se kterou jsou síť nebo senzor asociovány a měřená veličina. V mapě má uživatel na výběr z několika podkladových vrstev. Výchozí je Open Street Map, dále je na výběr ortofoto mapa, hydrologická mapa, topografická mapa a mapa ve stupních šedi. Kromě OSM se jedná o mapové služby společnosti Esri. Tematické vrstvy jsou reprezentovány dvěma povinnými vrstvami *Senzory* a *Sítě* a volitelně si uživatel může nahrát libovolnou vrstvu ve formátu GeoJSON. Ve výchozím stavu je nahraná vrstva okresy. Mapa je postavena JavaScriptovém API Leaflet. Pro připojení podkladových map společnosti Esri je použito Esri Leaflet API. Nejstarší verze aplikace používaly pro vizualizaci OpenLayers, toto API byla ale postupně nahrazeno právě Leafletem. Důvodem byla hlavně přehlednější dokumentace a větší množství různých příkladů a tutoriálů na webu.



Obr. 15 Ukázka mapové aplikace (zdroj: vlastní zpracování)

Mapové podklady

V této sekci lze připojit novou mapovou podkladovou vrstvu, která bude následně přístupná v sekci *Mapa*. Datová vrstva musí být ve formátu GeoJSON, ostatní formáty nejsou akceptovány. Pro připojení nové vrstvy je potřeba nahrát soubor s příponou *.json*, zadat název vrstvy (pod tímto názvem bude vrstva zobrazena v legendě a zadat barvu v jaké se bude zobrazovat. Barvu je možné zadat hexadecimálně nebo její anglický název např. green - stejně jako v CSS. Dále je v této sekci tabulka s přehledem všech připojených vrstev, editací této tabulky lze upravovat barvu a název vrstvy nebo vrstvu smazat. Nahrané vrstvy jsou uložené ve složce *layers*, která se nachází v kořenovém adresáři aplikace.

Konfigurační soubor

Konfigurační soubor je XML soubor, ve kterém jsou uloženy přihlašovací údaje k servisní databázi, uživatelské jméno a heslo výchozího administrátorského účtu, konfigurace logování aplikace, cesta ke složce *import,* cesta ke složce *export* a cesta k utilitě *pg_dump*, která slouží pro export do formátu SQL. Všechny tři cesty musejí být absolutní, to je také důvod, proč musejí být uloženy v konfiguračním souboru. Konfigurační soubor se nachází

ve složce *conf* v kořenovém adresáři aplikace. Soubor je možné editovat přímo nebo lze nastavení měnit v GUI aplikace. Cesty k *pg_dump*, složce *import* a *export* nelze upravovat z GUI, protože se nepředpokládá, že by se tato nastavení měnila. Po nainstalování aplikace je pouze správce serveru musí upravit v *conf.xml*. V konfiguračním souboru je uvedeno heslo k servisní databázi a heslo výchozí správce v *plain textu*. Předpokládá se, že aplikace poběží na zabezpečeném serveru, na který bude mít přístup pouze pověřený správce.

```
<?xml version="1.0" encoding="UTF-8"?>
<conf>
  <mgmtdb>
    <!-- Připojení k servisní databázi -->
    <database>ControlCenterMgmtDB</database>
    <hostname>localhost</hostname>
    <port>5432</port>
    <user>postgres</user>
    <password>password</password>
  </mgmtdb>
  <default_account>
    <!-- Výchozí administrátorský účet -->
    <login>admin</login>
    <password>password</password>
  </default_account>
  <pgdump_path>
    <!-- Absolutní cesta k souboru pg_dump -->
    <path>C:\Program Files\PostgreSQL\9.5\bin\pg_dump.exe</path>
  </pgdump_path>
  <exportdir_path>
    <!-- Absolutní cesta ke složce export -->
    <path>C:\xampp\htdocs\DP_DEV\export\</path>
  </exportdir_path>
  <importdir_path>
    <!-- Absolutní cesta ke složce import -->
    <path>C:\xampp\htdocs\DP_DEV\import\</path>
  </importdir _path>
  <logs>
   <apperror>1</apperror>
   <logerror>1</logerror>
   <accesslog>1</accesslog>
   <transactlog>1</transactlog>
   <date>1</date>
 </logs>
</conf>
```

Struktura konfiguračního souboru conf.xml.

Prohlížeče

Aplikace byla při vývoji průběžně testována ve všech běžně používaných prohlížečích (prohlížeče, které zaujímají nějakou významnou část trhu) – Google Chrome, Internet Explorer 11, Microsoft Edge, Opera, Mozilla Firefox, Safari a dále ještě ve dvou minoritních prohlížečích – ve Vivaldi a v Maxthonu. Jediný prohlížeč, ve kterém jsou problémy a stránky se v něm nemusejí zobrazovat korektně, je Internet Explorer. Vzhledem k tomu, že stránky jsou validní podle standardu HTML5 a ostatní prohlížeče aplikaci zobrazují korektně, nebyl na to brán zřetel a prováděna žádná optimalizace pro tento prohlížeč. Internet Explorer také již nemá dominantní postavení na trhu (aktuálně má největší penetraci Google Chrome) a sám Microsoft končí s jeho vývojem (vývoj vykreslovacího jádra Trident je ukončen a dále je vyvíjen jeho fork EdgeHTML) a plánuje ho postupně nahrazovat novým prohlížečem Edge, ve kterém se stránky aplikace zobrazují korektně.

5 VÝSLEDKY

Hlavním výsledkem, kterého bylo dosaženo v rámci této diplomové práce je aplikace s názvem Dohledové centrum senzorové sítě KGI. Jedná se o jednoduchý informační systém postavený nad databází PostgreSQL, který umožňuje prohlížet senzorová data, vizualizovat je v tabelární podobě a v podobě grafů. Pro uživatele jsou v aplikaci ve výchozím stavu předchystány ukázkové sady dotazů nad ukázkovými daty (databáze *vodni_hodnota_snehu*), kromě nich si může nadefinovat vlastní dotazy. Pro definici nových dotazů není potřeba zasahovat do zdrojového kódu aplikace, ale je možné je definovat interaktivně z uživatelského rozhraní aplikace. Dohledové centrum umožňuje dynamicky připojovat (a odpojovat) vlastní PostgreSQL databáze. Z databází lze exportovat data do celé řady formátů např. XML, CSV atd. V případě, že tabulky obsahují datové typy *geometry* je podporován i export do prostorových formátů. Import dat do databází je podporován jak ze souborů CSV tak z jiných PostgreSQL databází. Na závěr byly naprogramovány jednoduché analytické nástroje pro popisnou statistiku tabulek, výpočtů denních a klouzavých průměrů naměřených hodnot a výpočet vzdálenosti senzorů. Aplikace také obsahuje modul pro výpočet vodní hodnoty sněhu převzatý z diplomové práce Libora Kimpla.

Přehled funkcionality prototypu Dohledového centra senzorové sítě vytvořeného v rámci diplomové práce:

- Dynamické připojování a opojování PostgreSQL databází
- Přehled připojených databází a tabulek v těchto databázích
- Prohlížení a editace tabulek ze schématu *Public*
- Možnost řadit tabulky dle konkrétního sloupce nebo filtrovat dle jednoho parametru
- Možnost vizualizovat data ve formě grafu
- Možnost definovat vlastní dotazy nad daty bez nutnosti znalosti jazyka SQL
- Přehledný registr senzorů, sítí, lokalit a měřených veličin
- Možnost exportu dat ze všech připojených databází do souborových formátů - CSV, XML, dBase, SQL, JSON a prostorových souborových formátů – SHP, GML, KML a GeoJSON
- Možnost importu dat z CSV
- Možnost exportu/importu dat z databáze do jiné databáze (PostgreSQL -> PostgreSQL)
- Podpora uživatelských účtů – možnost pracovat v (omezeném) režimu anonymního uživatele nebo jako přihlášený uživatel, jednoduchá správa uživatelských účtů
- Jednoduché analytické nástroje pro výpočet vzdálenosti senzorů, výpočet klouzavého průměru, výpočet denního průměru, popisná statistika a výpočet vodní hodnoty sněhu (Libor Kimpl)
- Vizualizace rozmístění senzorů a sítí v pomocí mapové aplikace
- Možnost přidat do mapové aplikace vlastní tematické vrstvy ve formátu GeoJSON
- Vlastní logování aplikace

6 DISKUZE

Od začátku byla snaha celou aplikaci pojmout jako maximálně univerzální, a „nešít" ji na míru konkrétní databázi, která bude mít přesně definovanou strukturu – tj. přesně pojmenované tabulky a přesně pojmenované sloupce s konkrétními datovými typy. Tento požadavek se podařilo zcela naplnit. A priori se očekává, že připojené databáze mohou mít heterogenní strukturu. Tomu byla také celá aplikace přizpůsobena. Všechny nástroje a funkce mohou přijímat vstup z libovolné databáze. Výjimkou je výpočet vodní hodnoty sněhu. Nástroje byly naprogramovány tak aby uživateli umožňovaly nadefinovat si vlastní sady dotazů nad databází, kterou si připojí. Ve výchozím stavu je definováno jen pár ukázkových dotazů nad ukázkovou databází *vodni_hodnota_snehu*. Další požadavky se rovněž podařilo naplnit – aplikace podporuje import dat ze souborových formátů a z databáze, export dat do celé řady formátů prostorových i neprostorových formátů. Data z připojených databází lze zobrazovat v podobě tabulek nebo uživatelsky definovaných grafů. Požadavek na vizualizaci rozmístění senzorů v terénu byl rovněž splněn – v aplikaci je stránka s mapovým náhledem vytvořeným pomocí Leaflet API.

V Dohledovém centru jsou senzory reprezentovány pouze jako instance v databázi, chybí zde propojení s reálnými senzory a implementace standardů OGC SWE. Během vývoje aplikace nebyly k dispozici žádné senzory, na kterých by implementaci standardů bylo možno testovat, proto od toho bylo nakonec upuštěno. Navíc by tím objem práce neúměrně narostl. Dále se nestihla implementovat podpora databázového systému MySQL, se kterým se na začátku počítalo – minimálně pro importy dat. Ta byla sice rozpracována, ale do finální verze se nedostala, protože se již nestihla odladit. V aplikaci také chybí podpora PostgreSQL schémat, pracovat lze jen s tabulkami ve výchozím schématu *Public*. Tento bod vnímám asi jako největší nedostatek aplikace.

Aplikace byla během vývoje neustále testována. Každá komponenta během svého vývoje prošla testováním a nakonec byla testována funkcionalita aplikace jako celku. Přesto nepředpokládám, že by se mi vzhledem k objemu zdrojových kódů (cca 11 tisíc řádků kódu – počítáno bez prázdných řádků a pouze vlastní kód, bez knihoven a bez komponent třetí strany) podařilo nalézt a odstranit všechny chyby a je tedy možné, že budou nějaké objeveny.

Při vývoji aplikace jsem se učil jazyk PHP, se kterým jsem měl před diplomovou prací jen minimální zkušenosti. Tento fakt se zcela určitě také promítl do „kvality" zdrojových kódů, která může být z tohoto důvodu v různých částech dost rozkolísaná.

7 ZÁVĚR

Tato práce se v souladu se zadáním věnovala vývoji prototypu Dohledového centra senzorové sítě. Jejím výsledkem je informační systém Dohledové centrum senzorové sítě KGI. Aplikace Dohledového centra je hostována serveru Katedry geoinformatiky. Přehled funkcionality aplikace viz. kapitola Vlastní zpracování. Tato aplikace sice nepodporuje přímé připojení senzorů a senzorových sítí, ale slouží jako platforma nabízející nástroje pro jednoduchou správu databází obsahujících senzorová data a jednoduché analytické operace nad těmito daty. Přenos dat ze senzorů do databází aplikaci neošetřuje a ponechává prostor pro řešení třetí strany. Pro vývoj aplikace byla snaha používat především open source a freeware technologie – jazyk PHP, databázový systém PostgreSQL, QGIS. Výjimkou z toho tvoří operační systém Windows a Windows Server a virtualizační platforma Hyper-V .

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

[1] *52°North* [online]. 2016 [cit. 2016-08-07]. Dostupné z: http://www.52north.org

[2] *AWS IoT Developers Guide*. [online] Amazon Web Services, 2016 [cit. 2016-08-07]. Dostupné z WWW: http://docs.aws.amazon.com/iot/latest/developerguide/iot-dg.pdf

[3] Building automation. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-08-08]. Dostupné z: https://en.wikipedia.org/wiki/Building_automation

[4] *Core Tenets of IoT*. [online] Amazon Web Services, 2016 [cit. 2016-08-07]. Dostupné z: https://d0.awsstatic.com/whitepapers/core-tenets-of-iot1.pdf

[5] *Ekotechnika* [online]. 2016 [cit. 2016-08-07]. Dostupné z: http://www.ekotechnika.cz

[6] *Fiedler: Elektronika pro ekologii* [online]. 2016 [cit. 2016-08-07]. Dostupné z: http://www.fiedler.company/cs

[7] DUDA, Petr. Publikace dat ze senzorových sítí. In *Datové infrastruktury pro prostorově informační společnost*. 1. vydání. Brno: Masarykova univerzita, 2012. s. 63-82, 20 s. ISBN 978-80-210-6014-2.

[8] FU, Pinde a Jiulin SUN. *Web GIS: Principles and applications*. Redlands, California: ESRI Press, 2011. ISBN 978-1-58948-245-6.

[9] KEPKA, Michal. *Implementace standardu Sensor Observation Service*. Plzeň, 2011. Diplomová práce. Západočeská univerzita v Plzni. Vedoucí práce Jan Ježek.

[10] KIMPL, Libor. *Standardy pro dohledové centrum senzorové sítě*. Olomouc, 2012. Diplomová práce. Univerzita Palackého. Vedoucí práce Vilém Pechanec.

[11] *Microsoft Azure IoT Reference Architecture*. [online] Microsoft Corporation, 2016 [cit. 2016-08-07]. Dostupné z WWW: http://download.microsoft.com/download/A/4/D/A4DAD253-BC21-41D3-B9D9-87D2AE6F0719/Microsoft_Azure_IoT_Reference_Architecture.pdf

[12] MURDOCK Roy, HOFFENBERG Steve, ROMMEL Chris. *Amazon AWS & Microsoft IoT Deep Dive*. [online] VDC Research, 2016. Dostupné z WWW: http://www.vdcresearch.com/Landing/iot1/V-Amazon-AWS-Microsoft-Azure-IoT-Deep-Dive.aspx

[13] OGC 10-025R1. *Observations and Measurements - XML Implementation*. 2.0. OGC, 2011.

[14] OGC 12-000. *OGC® SensorML: Model and XML Encoding Standard*. 2.0. OGC, 2014.

[15] OGC 12-006. *OGC® Sensor Observation Service Interface Standard*. 2.0. OGC, 2012.

[16] OGC 09-000. *OGC® Sensor Planning Service Implementation Standard*. 2.0. OGC, 2011.

[17] OGC 08-094R1. *OGC® SWE Common Data Model Encoding Standard*. 2.0. OGC, 2011.

[18] OGC 08-094R1. *OpenGIS® SWE Service Model Implementation Standard*. 2.0. OGC, 2011.

[19] *OGC* [online]. 2016 [cit. 2016-08-07]. Dostupné z: http://www.opengeospatial.org

[20] PECHANEC, Vilém. *Senzorové systémy a jejich integrace s GIS v environmentálních studiích*. Olomouc, 2014. Univerzita Palackého.

[21] Smart city. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-08-08]. Dostupné z: https://en.wikipedia.org/wiki/Smart_city

[22] Smart grid. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-08-08]. Dostupné z: https://en.wikipedia.org/wiki/Smart_grid

PŘÍLOHY

Seznam příloh

Vázané přílohy:

Postup zprovoznění aplikace

Volné přílohy:

Příloha 1: Poster

Příloha 2: DVD

Popis struktury DVD

- aplikace – obsahuje složku s aplikací *Dohledové centrum senzorové sítě KGI*
- databaze – obsahuje zálohy (soubory s příponou *.backup*) ukázkové databáze *vodni_hodnota_snehu* a servisní databáze *ControlCenterMgmtDB* (ta je obsažena dvakrát – jedna s ukázkovými dotazy a ukázkovými daty v registrech, druhá prázdná)
- ostani - obsahuje soubor *php_dbase.dll* a návod na zprovoznění aplikace
- poster – obsahuje PDF s posterem
- text_prace – obsahuje PDF soubor s textem práce
- web – obsahuje webové stránky k diplomové práci

Příloha 1: Postup zprovoznění aplikace

1) Nachystat si platformu, na které budeme aplikaci hostovat. Je vyžadován Windows Server, na Linuxu nebudou funkční některé části. Desktopové Windows lze použít pro otestování aplikace, ale pro hostování produkčního webového serveru nejsou vhodné.
2) Nainstalovat Apache HTTP Server - buď *stand alone* instalaci nebo jako součást nějakého předpřipraveného balíku např. XAMPP, WAMP, WAPP atd.
3) Vložit složku s aplikací do kořenového adresáře httpd.
4) Nainstalovat PHP, pokud již nebylo součástí předpřipraveného balíku.
5) Vložit do složky *./php/ext/* soubor *php_dbase.dll*. Soubor je přiložen na DVD.
6) Do souboru *php.ini* do sekce *Dynamic Extensions* vložit nový záznam *extension=php_dbase.dll.* Po vložení je potřeba restartovat httpd. Načtení extenze lze zkontrolovat pomocí *phpinfo* souboru.
7) Nainstalovat PostgreSQL. Podporované jsou verze 9.X (aplikace byla testována s verzemi 9.3, 9.4, 9.5)
8) Zkontrolovat v souboru *php.ini* v sekci *Dynamic Extensions* jestli není zakomentovaný řádek *extension=php_pgsql.dll*, pokud ano odkomentovat.
9) Obnovit ze souboru *ControlCenterMgmtDB.backup* servisní databázi. K dispozici je ještě verze *ControlCenterMgmtDB_example_data.backup* obsahující ukázková data. Samotná ukázková data jsou uložena v databázi *vodni_hodnota_snehu*, kterou je potřeba rovněž obnovit ze zálohy. Databázi je potřeba ještě připojit v aplikaci v sekci *Databáze/Připojit novou databázi* a připojení pojmenovat *vodni_hodnota_snehu*.
10) V souboru *conf.xml* nastavit přihlašovací údaje k servisní databázi
    - jméno databáze - *ControlCenterMgmtDB* pokud nebylo změněno.
    - *hostname* – Je možné uvést IP adresu serveru nebo hostname, pokud se server nachází v jiné doméně, je potřeba *hostname* uvést ve tvaru FQDN. U vzdáleného serveru je zároveň potřeba v souboru *pg_hba.conf* povolit připojení na IP adresu nebo hostname serveru na kterém je hostována aplikace. Pro *localhosta* může být uvedeno ve tvaru *localhost* nebo IP adresa *127.0.0.1*.
    - port – Defaultní port pro PostgreSQL je 5432. U vzdáleného serveru je potřeba zajistit aby nebyl port, na kterém databáze naslouchá blokován na *firewallu* nebo v případě Linuxu v *iptables*. Na Windows Serveru (pokud je aktivní firewall) jsou ve výchozím stavu automaticky blokovány (v příchozích pravidlech – *inbound rules*) všechny porty a je nutné vytvořit pravidlo, které povolí požadovaný port. Samotná instalace databázového serveru si pravidlo nevytvoří. Na Linuxu je obvykle ve výchozím stavu vše povoleno a je na uživateli, aby blokoval porty, které povoleny mít nechce.
    - uživatel – Přihlašovací role v PostgreSQL která má oprávnění na servisní databázi.
    - Heslo – Heslo k databázovému serveru.
11) V souboru *conf.xml* nastavit přihlašovací údaje výchozího správce aplikace.
12) V souboru *conf.xml* nastavit absolutní cestu ke složce *import* a *export*
13) V souboru *conf.xml* nastavit absolutní cestu k utilitě *pg_dump*. Nachází se na cestě *./PostgreSQL/9.X/bin/pg_dump.exe*.
14) V souboru *pg_hba.conf* je potřeba nastavit připojení na localhostu na hodnotu *trust*.