

Internet věcí (IoT) – vzdálený monitoring a analýza dat z hydrogeologického vrtu

Ing. Jana Sedláčková

Bakalářská práce
2018



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ing. Jana Sedláčková**
Osobní číslo: **A14207**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **kombinovaná**

Téma práce: **Internet věcí (IoT) – vzdálený monitoring a analýza dat z hydrogeologického vrtu**

Téma anglicky: **The Internet of Things (IoT) – The Remote Monitoring and Analysis of Data from a Hydrogeological (Underground) Borehole**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Zpracujte koncepční návrh a navrhnete architekturu řešení pro vzdálený monitoring a analýzu dat z hydrogeologického vrtu.
3. Vytvořte aplikaci v jazyce JAVA, která bude zajišťovat přenos dat na server, a aplikaci, která bude tato data vyhodnocovat a publikovat.
4. Návrh hardwarově realizujte a celý systém otestujte.
5. Zpracujte dokumentaci aplikace.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. BUJOK, Petr a Arnošt GRMELA. Hydrodynamické zkoušky a výzkum sond. Ostrava: Vysoká škola báňská, 1993. ISBN 80-7078-144-0.
2. SCHILDT, Herbert. Mistrovství – Java. Brno: Computer Press, 2014. Mistrovství. ISBN 978-80-251-4145-8.
3. SCHILDT, Herbert. Java 8: výukový kurs. Přeložil Jakub GONER. Brno: Computer Press, 2016. ISBN 978-80-251-4665-1.
4. ECKEL, Bruce. Myslíme v jazyku Java. Praha: Grada, 2000. Myslíme v---. ISBN 80-247-9010-6.
5. ECKEL, Bruce. Myslíme v jazyku Java: knihovna zkušeného programátora. Praha: Grada, 2001. Knihovna programátora (Grada). ISBN 80-247-0027-1.
6. Java – Největší česká online učebnice. cz – Ajtácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další. [online]. Copyright ? 2017 itnetwork.cz. Veškerý obsah webu [cit. 27.11.2017]. Dostupné z: <https://www.itnetwork.cz/java>

Vedoucí bakalářské práce:

Ing. Tomáš Sysala, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

15. prosince 2017

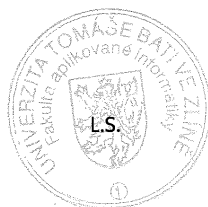
Termín odevzdání bakalářské práce:

25. května 2018

Ve Zlíně dne 15. prosince 2017



doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu


Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s přípoštění-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 24.5.2018


.....
podpis diplomanta

ABSTRAKT

Cílem této bakalářské práce je navrhnout řešení pro vzdálený monitoring a analýzu dat z hydrogeologického vrtu pomocí sítě IoT. Po základní rešerši dané problematiky je zde popsán výběr vhodné technologie přenosu, výběr čidla a hladinoměru. Součástí řešení je také návrh struktury databáze vhodné pro ukládání dat, a dále vytvoření aplikace v jazyce Java, která posílá data přijímá, a aplikace, která tato data vyhodnocuje a publikuje. V práci je popsána architektura obou aplikací, a také jejich implementace. V závěru bylo celé navržené řešení otestováno měřením v reálném hydrogeologickém vrtu.

Klíčová slova: IoT, LoRa, hladinoměr, Java, Thymeleaf, Spring, hydrogeologický vrt

ABSTRACT

The aim of this bachelor thesis is to design a solution for remote monitoring and analysis of data from a hydrogeological well through the IoT network. After a basic research of this issue, a selection of suitable transmission technology, sensor and waterlogger is described. The solution also includes designing a database suitable for storing data, and creating a Java application that receives data that is sent, and applications that evaluate and publish the data. The thesis describes the architecture of both applications as well as their implementation. At the end, the whole solution was tested by measuring in a real hydrogeological well.

Keywords:

IoT, LoRa, Waterlogger, Java, Thymeleaf, Spring, hydrogeological well

Chtěla bych poděkovat vedoucímu své práce Ing. Tomáši Sysalovi, Ph.D. za odborné vedení a konzultace, které mi poskytl v průběhu zpracování bakalářské práce. Dále bych ráda poděkovala pracovníkům společnosti ELKO EP s.r.o. Jaromíru Prikrylovi, Dis., a Ing. Liboru Juřicovi za pomoc při technické realizaci řešení. Poděkování patří také Mgr. Kamilu Ševečkovi, který mi jako lektor kurzů ukázal, že programování v Javě může být velká zábava.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 INTERNET VĚCÍ	11
1.1 TECHNOLOGIE POUŽÍVANÉ PRO BEZDRÁTOVÝ PŘENOS DAT.....	12
1.1.1 GSM, GPRS, LTE.....	13
1.1.2 Bluetooth.....	14
1.1.3 WiFi, WiFi HaLow.....	14
1.1.4 IQRF.....	15
1.1.5 MQTT.....	15
1.1.6 MiWi.....	15
1.1.7 Z-Wave.....	16
1.1.8 ZigBee.....	16
1.1.9 LoRa.....	16
1.1.10 SigFox.....	18
1.1.11 NB-IoT.....	19
2 MĚŘENÍ HLADINY PODZEMNÍ VODY V HYDROGEOLOGICKÝCH VRTECH	21
2.1 PRINCIPY MĚŘENÍ.....	23
2.2 STÁVAJÍCÍ POUŽÍVANÉ SYSTÉMY.....	24
2.2.1 Monitorovací systém ALA.....	25
2.2.2 Systém WATERLOGGER.....	25
2.2.3 H40 – GSM/GPRS datalogger, hladinoměr.....	26
II PRAKTICKÁ ČÁST	27
3 KONCEPČNÍ NÁVRH A ARCHITEKTURA APLIKACÍ	28
3.1 KONCEPČNÍ NÁVRH.....	28
3.1.1 Požadavky na frekvenci měření a ukládání dat.....	28
3.1.2 Volba vhodné přenosové technologie.....	28
3.1.3 Výběr čidla pro přenos dat.....	28
3.1.4 Hladinoměr.....	29
3.1.5 Popis procesu sběru dat a jejich přenosu z hladinoměru na server, formát dat.....	30
3.1.6 Struktura databáze pro ukládání dat.....	32
3.1.6.1 Tabulka UZIVATEL.....	33
3.1.6.2 Tabulka OPRAVNENI.....	34
3.1.6.3 Tabulka CIDLO.....	34
3.1.6.4 Tabulka ZAZNAM.....	34
3.1.6.5 Tabulka HLADINOMER.....	34
3.1.6.6 Tabulka VRT.....	35
3.2 ARCHITEKTURA APLIKACÍ.....	35
3.2.1 Aplikace zajišťující ukládání dat na server – program Collector.....	35
3.2.2 Aplikace pro publikování dat.....	37
3.2.2.1 Přihlašovací a registrační formuláře.....	37
3.2.2.2 Editační formuláře.....	38
3.2.2.3 Zobrazování dat.....	40
3.2.2.4 Odhlášení.....	42

4	IMPLEMENTACE	43
4.1	POUŽITÉ TECHNOLOGIE	43
4.1.1	Jazyk JAVA	43
4.1.2	Spring framework.....	43
4.1.3	Apache Tomcat	44
4.1.4	Apache Maven	44
4.1.5	Šablonovací systém Thymeleaf.....	45
4.1.6	Databáze Maria DB.....	45
4.1.7	Vývojové prostředí IntelliJ IDEA Ultimate.....	45
4.2	VSTUPNÍ KONFIGURACE	45
4.3	APLIKACE PRO UKLÁDÁNÍ DAT NA SERVERU	46
4.3.1	MainController	46
4.3.2	Repository	47
4.3.3	Entity	48
4.4	APLIKACE PRO PUBLIKOVÁNÍ DAT	48
4.4.1	Javové soubory.....	49
4.4.1.1	Balíček controller.....	50
4.4.1.2	Balíček entity	53
4.4.1.3	Balíček forms	53
4.4.1.4	Balíček logika	54
4.4.1.5	Balíček repository	56
4.4.2	Šablony.....	56
4.4.2.1	Složka login	57
4.4.2.2	Složka data.....	58
5	HARDWAROVÁ REALIZACE A TESTOVÁNÍ SYSTÉMU.....	59
6	DOKUMENTACE APLIKACE	62
6.1	SPECIFIKACE ZAŘÍZENÍ	62
6.2	POSTUP PŘI VYUŽÍVÁNÍ APLIKACE	62
	ZÁVĚR	64
	SEZNAM POUŽITÉ LITERATURY.....	65
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	68
	SEZNAM OBRÁZKŮ	69
	SEZNAM TABULEK.....	70
	SEZNAM PŘÍLOH.....	71

ÚVOD

Sledování hladiny podzemní vody ve vrtech je běžnou součástí hydrogeologické praxe. Tyto údaje slouží k poznání přirozených hydrogeologických podmínek v dané lokalitě, a zároveň umožňují sledovat jejich ovlivňování činností člověka. Na základě vědomostí o režimu podzemní vody můžeme lépe využívat její přirozené zásoby, což se v současné době dlouhodobého srážkového deficitu stává stále aktuálnější. Při realizaci hydrodynamických zkoušek můžeme vyhodnotit vlastnosti horninového prostředí a na základě výpočtů odhadnout, kolik podzemní vody se může z daného objektu čerpat, nebo jestli například čerpáním z naší studny nedojde k ovlivnění té sousedovy. Poznatky o úrovni hladiny podzemní vody jsou užitečné také např. v inženýrské nebo sanační geologii - při projektování staveb nebo odvodňování stavebních jam, nebo při vytváření hydraulické bariéry, která brání šíření kontaminace.

S rozvojem techniky se pro měření hladiny podzemní vody začaly stále více rozšiřovat technologie využívající principy dálkového přenosu dat. Pracovníci čerpacích osádek se zázemím pojezdne marigotky v pravidelných intervalech provádějící měření pomocí ručních hladinoměrů byli ve velké míře nahrazeni hladinoměry „automatickými“, které naměřené údaje ukládají do paměťové jednotky, a jsou schopné je dálkově odesílat příjemci nebo na server. Na českém trhu funguje několik takových systémů, všechny však posílají data pomocí sítě GSM/GPRS.

Cílem této práce je navrhnout funkční systém, který by prováděl sběr dat z hydrogeologického vrtu, a tato data přenášel na server pomocí technologie Internetu věcí (IoT). Internet věcí se v současné době velmi rychle rozvíjí, a v budoucnu se předpokládá jeho široké využití. Předkládaná práce tak může přispět k vytvoření funkčního systému řešení dálkového přenosu dat z hydrogeologického vrtu pomocí technologie IoT, u které se do budoucna předpokládá rychlý rozvoj.

I. TEORETICKÁ ČÁST

1 INTERNET VĚCÍ

Internet věcí (Internet of Things - IoT) je pojem, o kterém se hodně mluví, a v současné době prochází tato oblast bouřlivým rozvojem. Využívá se především ve dvou oblastech, kterými jsou průmyslové využití (automatizace, doprava, elektrotechnika, automobilový průmysl, zdravotnictví a další) a dále využití spotřebitelské (např. chytrá domácnost, chytré nakupování).

Existuje mnoho různých definic IoT. Dle jedné z nich jde o vzájemné, většinou bezdrátové propojení více zařízení, která spolu komunikují pomocí internetu nebo vlastní sítě. Zařízení musí komunikovat s ostatními zařízeními a musí být schopna snímání, sběru dat, ovládání nebo zpracování dat. Stručně se tedy dá říci, že se jedná o vzájemně propojenou síť snímačů, aktuátorů a řídicích systémů, které mají za cíl dosáhnout vyšší efektivity, bezpečnosti nebo komfortu ve srovnání se stavem před jejich nasazením [1].

Zařízení (věc) z pohledu Internetu věcí představuje neživý objekt, který v sobě obsahuje elektroniku, software a senzory, pomocí kterých snímá určitou veličinu (veličiny). Tyto informace jsou ve formě dat většinou následně přenášeny sítí do cloudu s cílem jejich následné analýzy. Cloud je vzdálený server, na kterém se data ukládají a mohou se s nimi provádět další operace s cílem jejich zpřístupnění koncovému uživateli, nejčastěji pomocí webové stránky nebo mobilní aplikace [2].

Dle [3] každá architektura IoT systému obsahuje základní stavební bloky (komponenty), kterými jsou Hardware, Middleware a Software. Jako **hardware** lze označit veškeré fyzické věci generující data, komunikační a výpočetní infrastrukturu, datová úložiště. **Middleware** je potom programové vybavení, které umožní propojení jednotlivých prvků internetu věcí a dále pak sběr, uložení a sdílení výsledků mezi různými systémy (např. pomocí cloudového úložiště). Nad získanými daty je použit **software** pro analýzu, skládání a kombinování velkých objemů heterogenních dat. Výsledky ze zpracování dat potom vedou k automatickému nebo poloautomatickému ovlivnění situace nebo alespoň pro podporu v rozhodovacích a organizačních procesech. Software také slouží pro sdílení a prezentaci výsledků. Důležitým faktorem při využívání IoT je zajištění bezpečnosti.

S IoT je těsně svázaný pojem **Industry 4.0** – čtvrtá průmyslová revoluce. Její charakteristikou a cílem je kompletní digitalizace, robotizace a automatizace většiny současných lidských činností pro zajištění větší rychlosti a efektivity výroby přesnějších, osobitějších, spolehlivějších a levnějších produktů, současně pro efektivnější využití materiálů a ekolo-

gičtějšímu průmyslu i lidskému životu [4]. Industry 4 bude zasažen nejen průmysl, ale také doprava, zemědělství, obchod, domácnosti a bydlení. Na průmyslové úrovni má jít o nahrazení manuální lidské práce robotizací, současné „manuální“ zadávání výrobních dat a postupů má být nahrazeno automatickým elektronickým předáváním informací prostřednictvím zapisovatelných RFID (Radio Frequency Identification) čipů umístěných na každém materiálu, věci a produktu. Data budou automaticky předávána mezi databázovými systémy běžícími na různých navzájem internetem propojených velkých datových serverech (tzv. Cloudech). Všechny tyto změny budou iniciovány především touhou ušetřit náklady a ještě více zefektivnit a zrychlit výrobu. V domácnostech budou vzájemně elektronicky propojeny jednotlivé domácí systémy, které spolu budou komunikovat s cílem maximální efektivity včetně spotřeby energií. Tzv. automatizovaný dům by měl sám reagovat na změny vnějších podmínek a řídit chod a bezpečnost celé domácnosti i zásobování bez lidské intervence. Výše uvedené změny vyvolají také změnu přístupu marketingu a reklamy, výuky a výchovy dětí, zdravotního a pojišťovacího systému, koncepce trhu práce a v důsledku toho i sociálně společenské změny.

1.1 Technologie používané pro bezdrátový přenos dat

Internet věcí je primárně určen pro vytvoření speciálních datových sítí ze senzorů a čidel (případně aktuátorů), které budou snímat požadovanou veličinu nebo provádět danou akci. Jinými slovy se jedná o zařízení, kterým stačí poslat jen několik zpráv během dne a nepotřebují být stále online, což jim umožní fungovat i několik let pouze s napájením pomocí baterií. Komunikace někdy dokonce probíhá pouze jedním směrem. Tato zařízení nemají žádnou IP adresu (místo ní obsahují jednoznačný identifikátor, podle kterého se určí, z jakého zařízení zpráva přišla), a proto je pro ně nutné vybudovat speciální datové komunikační síť pro Internet věcí [2].

Ke dvěma hlavním důvodům, které brání rychlejšímu rozvoji IoT, patří pořizovací cena jednotlivých čidel a nedostatečná komunikační infrastruktura pro IoT zařízení. Speciálně pro Internet věcí byl vyvinut nový typ sítě, který nese označení LPWAN (Low Power Wide Area Network – nízkoenergetická síť pro rozsáhlé území). Tento typ sítě umožňuje přenášet malé množství dat mezi IoT zařízeními na velké vzdálenosti. Nízká spotřeba zařízení, nízká přenosová rychlost a zamýšlené použití odlišuje tento typ sítě od klasické bezdrátové WAN (World Area Network) „internetové“ sítě, která je navržena pro připojení uživatelů nebo firem a přenášení více dat s využitím většího výkonu [2].

V současné době je pro IoT k dispozici poměrně velké množství možností přenosu dat, přičemž při rozhodování, kterou z nich zvolit, je nutno brát v úvahu její dosah, spolehlivost, rychlost přenosu dat, množství přenášených dat, energetickou spotřebu a v neposlední řadě také bezpečnost. Některé z technologií nejsou primárně určeny pro IoT, jiné fungují pouze pro přenosy na malé vzdálenosti a pro malý objem dat, a další splňují požadavky LPWAN. V České republice jsou aktuálně tři firmy, které budují komunikační infrastrukturu pro Internet věcí: Společnost SimpleCell, která využívá infrastrukturu T-Mobile a technologii Sigfox; České Radiokomunikace používající technologii LoRa a mobilní operátoři Vodafone a O₂, kteří provozují v rámci sítí LTE a 5G síť na technologii NB-IoT (Narrow Band IoT, neboli úzkopásmová síť) [2].

V dalším textu je uveden stručný přehled jednotlivých technologií, se zaměřením na tři výše uvedené.

1.1.1 GSM, GPRS, LTE

GSM technologie je označována jako 2. generace bezdrátových telefonních technologií mobilních sítí (2G) a slouží k přenosu hlasových dat a SMS, přičemž používá frekvence 900MHz a 1800MHz. Nástavbou pro 2G síť je technologie GPRS, která navíc nabízí i možnost přenosu dat na základě IP a WAP protokolů. Přenosová rychlost je však relativně nízká (cca 20kb/s). 3G je třetí generace mobilních technologií. Úkolem je přenos jak hlasu, tak i dat (e-mail, zprávy). Mobilní technologie 3G pracují v pásmu 2,1 GHz a umožňují využívání většího počtu služeb, jako je např. rychlý přenos pomocí nastavby sítě 3G HSDPA (High Speed Download PACKET Access), jehož teoretické maximum je 14,4 Mbit/s. Výhodou této technologie je vysoké pokrytí, velké množství operátorů, nízká cena za přenos dat, roamingový přenos dat bez poplatků v rámci EU, spolehlivost. Nevýhodami pak vysoká energetická spotřeba, nízká přenosová rychlost, náročnější komunikace s moduly [5].

V současnosti nejnovější technologií pro mobilní síť je technologie 4. Generace, kterou také označujeme jako LTE (Long Term Evolution). Je určena pro vysokorychlostní internet v mobilních sítích, přičemž umožňuje až 10x rychlejší připojení než síť 2G a 3G. Frekvenční pásmo pro Evropu je 700 – 2600 MHz s přenosovou rychlostí 70-250 Mb/s. Výhodami je vysoká přenosová rychlost, rychlé budování a rozšiřování sítí, roamingový přenos dat bez poplatků v rámci EU, spolehlivá technologie, perspektivní frekvenční pásmo. Nevýhodami pak vysoká energetická spotřeba a zatím vyšší cena modulů [5].

Obecně platí, že 4G či 3G sítě jsou pro komunikaci s IoT zařízení nevhodné jak po datové, tak po stránce spotřeby energie. V těchto sítích jednotlivá zařízení posílají spoustu různých dat pro zahájení a ukončení komunikace, což je v případě Internetu věcí krajně nepraktické, jelikož sítě musejí zpracovat o to více dat. Dále musíme mít na paměti, že na stávající infrastrukturu musejí paralelně běžet služby pro přenos dat z klasických mobilních zařízení jako jsou chytré telefony. Zařízení v 4G a 3G sítích rovněž komunikují mnohem častěji a „udržují“ spojení se základnovými stanicemi, což je opět pro IoT zařízení energeticky náročné a nevhodné.

1.1.2 Bluetooth

Technologie Bluetooth, která je vyvíjena od roku 1994, je určena pro bezdrátovou komunikaci více elektronických zařízení na krátkou vzdálenost. Pracuje v kmitočtovém pásmu od 2,402 GHz do 2 483 GHz. Toto řešení umožňuje Smart zařízením přístup k internetu bez nutnosti dodatečného zařízení. Nejvyšší přenosová rychlost je 24 Mb/s. Začátkem roku 2017 byla uvedena na trh verze Bluetooth 5, která deklaruje několikanásobně větší dosah a přenosovou kapacitu. Výhodami jsou nižší spotřeba a schopnost komunikace přímo se smartphony. Nevýhodami jsou krátký dosah, placená licence a také skutečnost, že, využívá přetížené frekvenční pásmo 2,4Ghz [5].

1.1.3 WiFi, WiFi HaLow

Technologie WiFi (Wireless Fidelity) je bezdrátová technologie, která využívá pro přenos rádiové vlny v sítích WLAN ve frekvenčním pásmu v rozsahu 2,4 a 5 GHz s rychlostí přenosu až 150 MB/s. V současnosti je WiFi nejvíce používanou technologií pro bezdrátový přenos dat. Základním prvkem sítě je tzv. přístupový bod vysílající signál, který je zařízení schopno rozpoznat a zpracovat. Výhodami jsou vysoká přenosová rychlost, skutečnost, že se jedná o rozšířenou technologii, a také vysoká bezpečnost. Nevýhodami pak vysoká energetická spotřeba a topologie bod-bod [5].

Wi-Fi HaLow je nejnovější technologie, která na rozdíl od běžných Wi-Fi sítí pracujících na frekvencích 2,4 a 5 GHz je provozována na 900 MHz, což zaručuje širší pokrytí a menší náchylnost k rušení. Přenos dat neprobíhá kontinuálně, ale v pravidelných dávkách, jejichž interval lze nastavit. Nižší vysílací výkon a odlišné schéma tak dovoluje provoz na baterie. Wi-Fi HaLow má být přímým konkurentem Bluetooth, jen s větším dosahem. Certifikace produktů je naplánována na letošní rok [6].

1.1.4 IQRF

IQRF je platforma českého výrobce, určená pro bezdrátovou konektivitu s malým výkonem, nízkou rychlostí a nízkým objemem dat. Dosah komunikace je v řádu desítek až stovek metrů, ve zvláštních případech i několik kilometrů. Využití je především v sítích IoT, zejména pro telemetrii, průmyslové řízení a automatizaci budov a měst (pouliční osvětlení, parkoviště atd.). Prvky IQRF mohou být použity s libovolným elektronickým zařízením, kdykoli je potřeba bezdrátového přenosu, např. dálkové ovládání, monitoring dálkově získaných dat nebo připojení více zařízení k bezdrátové síti. Používá bezlicenční pásma 868 MHz, 916 MHz a 433 MHz [6].

1.1.5 MQTT

MQTT je zkratka pro *MQ Telemetry Transport*. Jedná se o mimořádně jednoduchý a nenáročný protokol postavený nad TCP/IP a navržený pro jednoduchá zařízení, úzkou šířku pásma, vysokou latenci nebo nespolehlivé sítě. Byl vyvinut v roce 1999 doktorem Andy Stanford-Clarkem z IBM a Arlenem Nipperem z firmy Arcom (nyní Eurotech). Od roku 2013 je pod správou sdružení OASIS (The Organization for the Advancement of Structured Information Standards). Základní principy tohoto protokolu jsou minimalizace zatížení sítě a omezení požadavků na zdroje zařízení. Zároveň se snaží zajistit spolehlivost a určitý stupeň zajištění doručení zprávy. Tyto vlastnosti předurčují MQTT vhodným k nasazení v M2M (machine-to-machine) zařízeních a internetu věcí, kde jsou úzká šířka pásma a dlouhá výdrž baterie prioritou [6].

1.1.6 MiWi

MiWi je bezdrátový protokol navržený firmou Microchip Technology na bázi standardu IEEE 802.15.4. Je určen pro nízké přenosové rychlosti a krátké vzdálenosti – tedy především pro nízkorozpočtové sítě s omezenou pamětí jako je dálkové řízení a monitoring, automatizované odečty, sensorové sítě atd. Působí zejména na frekvenci 2,4 GHz a podporuje všechny síťové konfigurace (hvězdice, strom, mesh); dosah je potom 20 – 50 metrů. Mezi přednosti MiWi patří snadné vytváření a nasazení sítí, přenositelnost aplikací napříč různými RF vysílači a jednoduché škálování topologie sítě. Bohužel, všechny tyto výhody platí pouze při užití čipů firmy Microchip Technology [6].

1.1.7 Z-Wave

Z-Wave je bezdrátová komunikační technologie pro domácí automatizaci. Využívá ji více než 325 výrobců sdružených v Z-Wave Alianci. Nachází uplatnění v řízení osvětlení, systémech HVAC (Heating, Ventilation, and Air Conditioning), zabezpečovacích systémech, domácích kinech, automatickém ovládání oken a stínění, bazénové technice či ovládání garáží a přístupových systémech. Z-Wave minimalizuje spotřebu energie a je tak vhodná do zařízení napájených bateriemi. Je navržena pro poskytování spolehlivých datových přenosů malých objemů s nízkou latencí a do rychlosti 100 kbit/s. U nás pracuje v pásmu 868 MHz. Její dosah je přibližně 100 m, v závislosti na přímé viditelnosti a stavebních materiálech v cestě signálu [6].

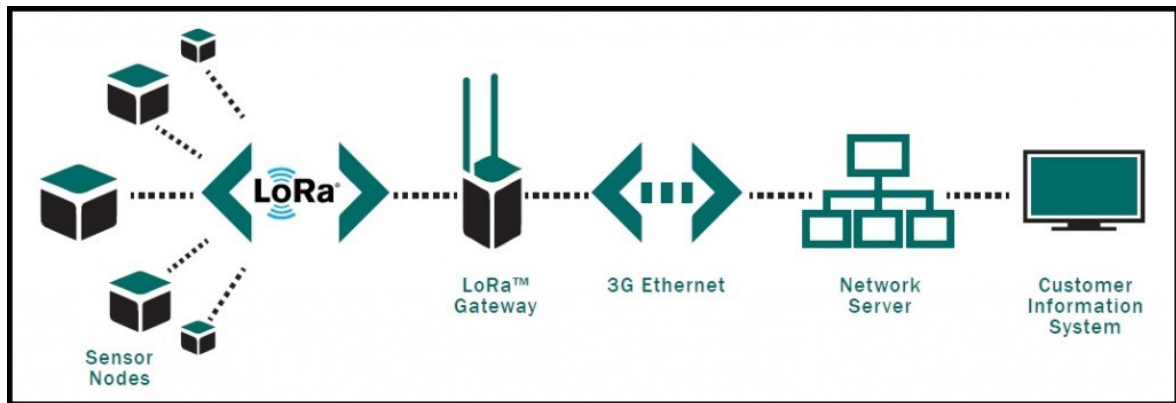
1.1.8 ZigBee

ZigBee je bezdrátová komunikační technologie vystavěná na standardu IEEE 802.15.4. Podobně jako Bluetooth je určena pro spojení nízkovýkonových zařízení v sítích PAN (personal-area-network) na malé vzdálenosti do 75 metrů. Primární určení směřuje do aplikací v průmyslu a senzorových sítích. Pracuje v bezlicenčních pásmech (generální povolení) přibližně 868 MHz, 902–928 MHz a 2,4 GHz. Přenosová rychlost činí 20, 40, 250 kbit/s. V současné době se na vývoji a rozvoji tohoto standardu podílí více než šedesát firem. ZigBee je navržena jako jednoduchá a flexibilní technologie pro tvorbu i rozsáhlejších bezdrátových sítí, u nichž není požadován přenos velkého objemu dat. K jejím hlavním přednostem patří spolehlivost, jednoduchá a nenáročná implementace, velmi nízká spotřeba energie a v neposlední řadě též příznivá cena. Díky těmto vlastnostem nalezne uplatnění v celé škále aplikací typu automatizace budov, spotřební elektronika, počítačové periferie, průmyslová automatizace nebo zdravotnictví [6].

1.1.9 LoRa

Lora je velmi spolehlivá bezdrátová technologie, která slouží pro přenos menšího množství dat na velkou vzdálenost. Základní prvek této technologie je přenos rádiových vln pomocí rozprostřeného spektra (patent společnosti Semtech), díky čemuž je Lora odolná vůči okolnímu rušení. V závislosti na typu zařízení je životnost baterie až 10 let. Základní zařízení (základna) je schopna pokrýt v krátkém časovém horizontu rozsáhlé území. Dosah je závislý na bariérách v daném území, jako je například zástavba území či jeho členitost. Pro přenos se v Evropě používá frekvence 868 MHz. K výhodám patří nízká spotřeba energie a

nízké provozní náklady, velký dosah, vysoká odolnost vůči rušení, možnost oboustranné komunikace, k nevýhodám pak nízká přenosová rychlost [5]. Schéma principu technologie LoRa zachycuje obrázek (Obr. 1).



Obr. 1. Schema přenosu dat v technologii LoRa [7]

LoRaWAN specifikuje komunikační protokol – Low Power Wide Area Network“ (LPWAN) a síťovou architekturu systému pro účely IoT. Specifikace LoRaWAN popisuje komunikační protokol mezi danými čidly vybavenými rádiovou jednotkou a základnovými stanicemi (LoRa Gateway). Gateway komunikuje se serverem prostřednictvím standardního IP připojení, koncové zařízení používá single-hop bezdrátovou komunikaci s gateway. Komunikace koncového zařízení je standardně obousměrná. Nejčastěji se používá architektura hvězdy, protože umožňuje delší životnost baterie. Prodloužení životnosti baterie a koncového zařízení, jakož i maximalizace celkové kapacity sítě je zajištěna síťovým serverem, který řídí přenos dat a RF kanál samostatně pro každé koncové zařízení prostřednictvím tzv. Adaptive data rate schématu (ADR). Řídicí server zajišťuje komunikaci s LoRa Gateway, řídí a dynamicky mění rádiové parametry komunikace, monitoruje stav celé sítě včetně provozních informací o stavu koncových čidel a zajišťuje směrování zpráv do odpovídajícího aplikačního serveru, a tím zajišťuje směrování zpráv jen a pouze určenému adresátovi. Jednotlivá IoT zařízení tedy komunikují pomocí LoRa technologie k nejbližší LoRa Gateway, odkud jsou data přeposlána přes 3G síť nebo Ethernet do nadřazeného serveru / cloudu. Zde mohou být opět dále zpracována a následně zpřístupněna koncovým uživatelům prostřednictvím webové stránky či mobilní aplikace. Pro přenos se využívá frekvenční pásmo 868 MHz, rychlost přenosu dat se pohybuje v rozpětí 0,25kb/s – 50kb/s. Jelikož je důležité dbát i na bezpečnost při přenosu dat, standardně se používá několik kódování dat – zabezpečení na síťové a aplikační úrovni a zabezpečení na zařízení. Koncový

bod má přidělen unikátní 64bitový identifikátor a dva 128bitové klíče, kterými jsou data šifrována pomocí AES. K výhodám patří nízká spotřeba, velký dosah, celosvětový standard a existence veřejných operátorů. K nevýhodám pak zpoplatnění přenosu dat operátorem a složité nastavení parametrů [2, 5].

Obecně LoRa pouze zpřístupňuje ostatním firmám technologii, se kterou jsou schopni si vybudovat vlastní síť pro Internet věcí. LoRa tedy poskytuje LoRa Gateway, komunikační modemy pro zařízení a popis standardu a rozhraní pro integrátory. V ČR v současné době používají technologii LoRa České radiokomunikace, které osadily své základnové stanice o LoRa Gateway, čímž byla vytvořena LoRa síť, ale veškerá data z těchto základnových stanic jsou směřována na servery poskytovatele a jsou tedy uloženy přímo u poskytovatele služby, což je podstatný rozdíl oproti dále popsanému Sigfoxu. LoRa také umožňuje vybudovat privátní síť, která bude dostupná například pouze v areálu průmyslového podniku, což může být výhodné pro vytvoření sítě Internetu věcí v uzavřených sítích. Dosah signálu z LoRa zařízení je až 20 km ve volné krajině a 2–5 km v husté městské zástavbě. Zatímco Sigfox je vhodný všude tam, kde je dostačující jednosměrná komunikace s periodou 10 minut, tak LoRa umožňuje obousměrnou komunikaci a je tedy vhodná pro čtení i zápis. LoRa se tedy hodí nejen pro odečty a snímání různých veličin, jako jsou teplota, vlhkost, tlak, CO₂ či spotřeby energií, ale jsou dostupná i LoRa zařízení, která umožňují zařízení vzdáleně ovládat [2].

1.1.10 SigFox

SigFox je technologie pro bezdrátový přenos menšího množství dat. IoT zařízení mohou komunikovat na poměrně velké vzdálenosti (až 50km v terénu a 10km v zastavěné oblasti). Přenos dat je spolehlivý a bezpečný. Kódování je realizováno na aplikační vrstvě pomocí AES128 a 16bitové autentizace. Zařízení, využívající v rámci IoT tuto technologii se většinou vyznačují nízkou spotřebou, takže baterie může vydržet více než 10 let. Každé toto zařízení je v síti jednoznačně identifikováno pomocí unikátního 32bitového Sigfox ID, které je přiřazeno při výrobě. Proces registrace zajišťuje, že v jednu chvíli má k zařízení přístup pouze jeden vlastník, což je především z důvodu zabezpečení a ochrany. Všechny zprávy jsou podepsány privátním klíčem a šifrovány, takže uživatel si může být jist pravostí zprávy. Sigfox je vysoce odolný vůči rušení. Přenos dat probíhá na frekvenci 868MHz s rychlostí 100b/s, přičemž počet zpráv za den vyslaných zařízeními je maximálně 144 (což v průměru odpovídá odeslání jedné zprávy za 10 minut), a přijmout může

pouze 4 zprávy denně. Zpětná komunikace je aktivována pouze na vyžádání koncovým zařízením (senzorem). Z tohoto popisu vyplývá, že Sigfox se hodí pro aplikace, kde se přenáší velmi malé množství dat a pouze několikrát za den. Typicky jsou vhodné na veškeré měření, odečty, sledování či zabezpečení apod. Obecně se Sigfox nehodí na nositelné technologie (např. chytré hodinky) a pro zařízení, kde je požadováno ovládání v reálném čase. Také se nehodí tam, kde se musí komunikovat oboustranně mnohokrát za den, což jsou např. platební terminály. Obecně k výhodám patří nízká spotřeba, velký dosah, vysoké pokrytí, roamingový přenos dat bez poplatků, k nevýhodám pak zpoplatnění přenosu dat operátorem, jednosměrná komunikace a přenos malého množství dat [2, 5].

Po vyslání signálu zařízením jej všechny základny v dosahu příjmu dekodují a následně odešlou pomocí zabezpečeného IP spojení, s ověřením pomocí jedinečného privátního klíče, přiřazeného danému zařízení, do Sigfox cloudu, odkud jsou data dostupná pro vlastníky zařízení pro další zpracování. Zákazník dostane přístup k centrálnímu cloudu a má k dispozici nástroje pro vývoj a využití aplikačního rozhraní (API). [2, 5].

Sigfox je globální operátor sítě pro Internet věcí se sídlem ve Francii. Data z IoT zařízení připojených do sítě Sigfox procházejí přes servery společnosti Sigfox ve Francii a až odsud jsou následně zpřístupněny dále. Strategie Sigfoxu je tedy najít lokálního partnera a vybudovat komunikační síť pro Sigfox (v ČR je tímto partnerem společnost SimpleCell). Síť Sigfox je aktuálně dostupná ve většině Evropy a buduje se infrastruktura v dalších 40 zemích po celém světě [2].

1.1.11 NB-IoT

NarrowBand IoT (NB-IoT) je nová technologie, která využívá LPWAN síť. Umožňuje připojení velkého počtu zařízení prostřednictvím již existujících frekvencí. Jelikož NB-IoT zařízení využívají licencovaného pásma mobilních operátorů, dosahují také nejvyšší rychlosti v porovnání s technologiemi LoRa a Sigfox, která dosahuje až 200 kbit/s. Dosah signálu z NB-IoT zařízení k základnové stanici je kolem 20 km a teoreticky lze se zařízeními komunikovat všude tam, kde je k dispozici LTE síť operátora. Kromě klasických aplikací, jako je odečet a měření různých veličin či sledování odpadu v kontejnerech, jsou NB-IoT zařízení vhodná pro aplikace, kde je nutná častější a obousměrná komunikace. Mezi takové aplikace patří například sledování zásilek, nositelné technologie (např. chytré hodinky či chytrá jízdní kola) či aplikace pro zemědělství. K výhodám této technologie

patří nízká spotřeba, roamingový přenos dat bez poplatků navíc, očekává se rychlé vybudování sítě [2, 5].

Síť NB-IoT lze spustit na existující komunikační infrastruktuře mobilních operátorů (v ČR ji budují Vodafone a O₂), kdy pouhou úpravou software na vysílacích stanicích se mu vyhradí část LTE pásma. Na rozdíl od sítí na technologii Sigfox a LoRa není tedy nutné instalovat žádné dodatečné antény, stačí jen softwarový update na stávající infrastruktuře. Oproti sítím Sigfox a LoRa je zatím pokrytí území České republiky sítí NB-IoT malé. Jelikož pokrytí sítí LTE je v případě Vodafone i O₂ 99 procent území ČR a samotné rozšíření o NB-IoT je otázka aktualizace software ve stávajících základnových stanicích, pak lze očekávat, že rychlost rozšíření bude poměrně rychlá. Vzhledem k tomu, že zařízení s technologií NB-IoT využívají ke své komunikaci pásmo LTE, pak tato zařízení v sobě obsahují komunikační modul se zabudovanou SIM kartou. Uživatel tedy po zakoupení a následném spuštění zařízení aktivuje SIM kartu, které se spojí s NB-IoT sítí. Zařízení vysílají nebo přijímají data z nejbližší NB-IoT základnové stanice (NB-IoT Gateway), odkud jsou přeposlány do nadřazeného serveru/cloudu, kde jsou uložena, zpracována a následně zpřístupněna koncovému uživateli přes webovou stránku či mobilní aplikaci [2].

2 MĚŘENÍ HLADINY PODZEMNÍ VODY V HYDROGEOLOGICKÝCH VRTECH

Ruční měření hladiny podzemní vody v hydrogeologických vrtech je v současné době stále více nahrazováno měřením automatickým. Ruční měření je využíváno především v případě jednorázových záměrů (např. pro potřeby hydrogeologických posudků), nebo záměrů periodicky opakovaných, ale v dlouhých časových intervalech. Příkladem může být monitoring hladiny podzemní vody v rámci jímacích území s např. čtvrtletní periodou, kdy je potřeba jednorázově zaměřit větší množství pozorovacích vrtů, nebo třeba i častější monitoring v rámci sanačního čerpání při probíhající dekontaminaci podzemní vody. V takovém případě by bylo osazení všech těchto vrtů automatickými hladinoměry příliš nákladné, a ručně provedený záměr je efektivnější.

Automatické hladinoměry jsou využívány spíše u měření s kratší periodou, ať už se jedná o krátkodobější měření např. při provádění hydrodynamických zkoušek (HDZ), nebo dlouhodobější monitoringy. V případě provádění hydrodynamických zkoušek v podmínkách neustáleného proudění je potřeba sledovat dynamickou změnu hladiny podzemní vody v krátkém časovém intervalu např. 1 minuty, a to po dobu od prvních desítek minut až po několik měsíců. K dlouhodobým měřením s využitím automatizovaných stanic patří např. monitoring podzemní vody prováděný ČHMÚ (Český hydrometeorologický ústav) v síti pozorovacích vrtů, pokrývajících celou republiku, v denním intervalu.

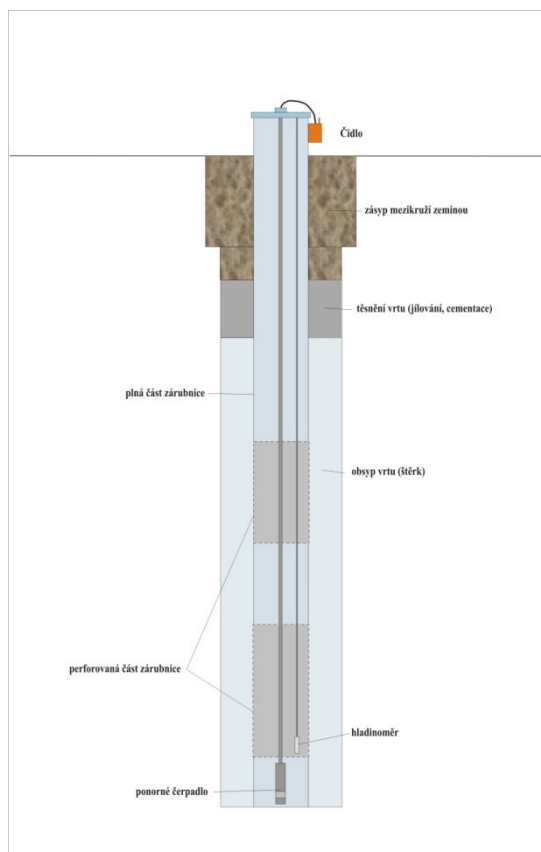
Jak vyplývá z výše uvedeného, požadavky na interval měření hladiny podzemní vody vyplývají z účelu měření a jsou velmi různorodé. V případě automatizovaného měření se většinou pohybují se v rozsahu od 1 minuty po 1 den, ve výjimečných případech mohou být i v řádu několika dní.

Záznamy hladiny podzemní vody jako výsledek automatizovaného měření se v běžné hydrogeologické praxi nejčastěji využívají pro vyhodnocení hydrodynamických zkoušek. Hydrodynamické zkoušky jsou základním zdrojem informací o fyzikálně-hydrogeologických parametrech zvodně, získaných přímo in-situ. Umožňují také objasnění vzájemných hydraulických vztahů mezi jednotlivými objekty, zjištění maximálního možného využití objektů poznání okrajových podmínek, kvality jímané tekutiny atd. [8]. Pro vyhodnocování HDZ v podmínkách neustáleného proudění s cílem vypočítat hydraulické parametry kolektoru se používají hodnoty úrovně hladiny podzemní vody, získané v postupně se prodlužujících intervalech (od 1 minuty až po 4 hodiny). Pro toto vyhodnocení existují speciální

programy, ve kterých je kromě záměrů hladiny potřeba použít také odborné znalosti hydrogeologa.

Další možností pro využití záznamů automatizovaného měření je získat přehled o úrovni hladiny podzemní vody v určitém časovém intervalu, ať už se jedná o její pohyb v průběhu HDZ nebo při provozním čerpání (často v souvislosti s čerpaným množstvím), nebo režimní měření (sledování přirozeného režimu podzemní vody v závislosti na geologických, hydrogeologických, klimatických a morfologických podmínkách). Tento přehled nejlépe získáme vykreslením grafu závislosti úrovně hladiny podzemní vody na čase, případně statistickým vyhodnocením (průměr, směrodatná odchylka, modus, medián, maximum, minimum, hodnoty křivky překročení).

Schematický náčrt hydrogeologického vrtu s instalovaným hladinoměrem a čidlem ukazuje obrázek (Obr. 2).



Obr. 2. Schéma hydrogeologického vrtu s hladinoměrem

2.1 Principy měření

Zařízení pro měření hladiny podzemní vody pracují na několika různých principech. Měření může být buď indikační (pouze indikuje, jestli je čidlo pod ponořené nebo ne), nebo tzv. registrační (výsledkem je naměřená hodnota). V hydrogeologických vrtech je možno využít následující principy měření [9]:

- Hladinoměry plovákové
- Hladinoměry akustické
- Hladinoměry hydrostatické (manometrické)
- Hladinoměry elektrokontaktní
- Hladinoměry pneumatické
- Hladinoměry bezdotekové

Plovákové hladinoměry

Základní součástí hladinoměru je plovák, obvykle v podobě lehkého dutého tělesa nebo tělesa vyrobeného z materiálu o velmi malé hustotě. Plovák je udržován na hladině vztlakovou silou, a hustota plováku tedy musí být vždy výrazně menší než hustota měřeného média. Plovák je ponořen do kapaliny jen velmi málo; poměr ponořené části plováku k jeho celkové výšce se rovná poměru hustot plováku a kapaliny. Při měření menších změn polohy hladiny (do 1 m) se používají plováky ve tvaru koule, pro měření velkých změn hladiny (do 40 m) se používají plováky válcového tvaru s vodicí tyčí. Pohyb plováku může být vyveden přes kladku lankem nebo řetízkem, obvykle ve spojení s protizávažím. Poloha plováku se určuje buď přímo přečtením polohy protizávaží na podložené stupnici, nebo se převádí na elektrický signál prostřednictvím převodníku [10].

Akustické hladinoměry

Akustické hladinoměry se používají pouze pro jednorázová měření. Jejich zástupcem je tzv. Rangova píšťala. Jedná se o kovovou trubku na jedné straně zavěšenou na pásmo, která má z vnější strany zářezy. Při rychlém ponoření do vody z ní uniká vzduch a píšťala vydává zvuk.

Hydrostatické (manometrické) hladinoměry

Poloha hladiny se vyhodnocuje z hydrostatického tlaku sloupce kapaliny ve vrtu. Pro měření polohy hladiny touto metodou se využívá ponorná sonda. Její podstatnou součástí je membrána s tenzometrickým snímačem. Tenzometrický senzor je chráněn oddělovací membránou z korozi vzdorného materiálu nebo plastu. Hydrostatický tlak, přímo úměrný výšce hladiny kapaliny nad oddělovací membránou, je přenášen prostřednictvím náplně inertního oleje na měřicí polovodičový čip s tenzometrickým můstkem. Výstupní signál je elektronickými obvody teplotně kompenzován a upraven na standardní elektrický výstup 4 až 20 Ma. Tenzometrický senzor v sondě měří celkový tlak, který je součtem hydrostatického a atmosférického tlaku. Pro vyhodnocení polohy hladiny je nutno atmosférický tlak odečíst, a proto je přívodní kabel k sondě opatřen propojovací hadičkou (kapilárou), která funguje jako přívod referenčního tlaku. Ponorné sondy jsou vhodné pro měření polohy hladiny podzemní vody ve vrtech hlubokých až 200 m [9].

Elektrokontaktní hladinoměry

Tyto hladinoměry fungují na principu vytvoření vodivého spojení v elektrodách při dotyku s hladinou. Využívají se především při jednorázovém měření, kdy je čidlo zavěšeno na pásmu a při dotyku s hladinou se rozsvítí led-dioda, případně ještě dojde k akustickému signálu. Mohou se používat také pro průběžná měření, kdy posun čidla zajišťuje elektromotor [9].

Pneumatické hladinoměry

Využívají měření tlaku v určité části sloupce vody ve vrtu pomocí tenké trubičky, kdy dochází k ustanovení tlakové rovnováhy mezi ústím trubičky a vodou ve vrtu. Měří se tlak na povrchu. Výhodou je nutnost minimálního prostor pro trubičku ve vrtu a s tím související minimální riziko poškození [9].

Bezdotekové hladinoměry

U těchto hladinoměrů nedochází ke kontaktu čidla s podzemní vodou. Patří sem hladinoměry ultrazvukové, optické, případně radarové. Při měření hladiny podzemní vody ve vrtech se příliš nepoužívají, neboť zde vlivem výstroje vrtu a instalací čerpadla, případně jiných zařízení ve vrtu vznikají odrazy, které vedou ke snížení přesnosti těchto metod.

2.2 Stávající používané systémy

Na českém trhu je v současné době k dispozici několik systémů pro automatické měření hladiny podzemní vody, které v sobě zahrnují na sebe navazující procesy měření pomocí

hladinoměru, přenos naměřených dat a jejich ukládání. Přehled těch nejpoužívanějších je uveden v dalším textu. Všechny se vyznačují tím, že přenos dat je prováděn pomocí sítě GSM/GPRS.

2.2.1 Monitorovací systém ALA

Monitorovací systém ALA má českého výrobce Ing. Aleš Sekanina [11]. Zahrnuje celý řetězec zpracování dat – sondy pro měření v terénu, přenos dat sítě GSM a grafické zobrazení na internetu. Komunikace se sondami probíhá pomocí e-mailů, SMS a dalších GSM a internetových protokolů.

Hladinoměry ALA jsou sondy pro měření hladiny a teploty vody s možností automatického přenosu dat přes GSM na internetový server. V nerezovém tělese zanořeném pod hladinou je umístěno tlakové a teplotní čidlo a měřicí elektronika včetně záznamníku. Závěsný kabel k ponornému tělesu může mít délku až několik stovek metrů, a je zakončen GSM přenosovým modulem. Nízká spotřeba energie umožňuje provoz více než 10 roků bez výměny baterií. Hladinoměr může být vybaven externím měřicím vstupem pro připojení průtokoměru. Systém varovných zpráv umožňuje zadat meze měřených veličin, při jejichž překročení se rozešlou varovné e-maily nebo SMS.

Přenos dat z hladinoměru i příkazů pro hladinoměr probíhá jako čitelný text přes e-mail a SMS, a to buďto pravidelně jako blok nových dat ze záznamníku od posledního přenosu, nebo při každém zápisu nových dat do záznamníku. Ke komunikaci s hladinoměrem není nutné žádné další programové vybavení, stačí poštovní klient, e-mailová schránka a mobilní telefon. Změřená data je možno sledovat na internetovém serveru ALA, který zajišťuje příjem e-mailů s daty a jejich převod do databáze a běžícího grafu.

2.2.2 Systém WATERLOGGER

Jedná se o telemetrický systém pro monitoring hladiny, teploty a elektrické vodivosti povrchových a podzemních vod s možností rozšíření o další senzory distribuovaný firmou Eko-technika spol. s r.o. [12]. Má integrované barometrické čidlo pro nepřímou kompenzaci atmosférického tlaku, ultrazvukový senzor hladiny nebo manometrický hladinoměr se zdvojeným záznamem dat. U typu EWS je k dispozici funkce pro lokální varovné systémy (alarmové SMS při překročení limitních hodnot nebo gradientu dané veličiny). Je zde možnost napájení z miniaturního solárního panelu. Dálkový přenos dat probíhá na e-mail, FTP server nebo do webové databáze EnviroDATA pomocí 4-pásmového GSM/GPRS

modemu. Tyto hladinoměry využívá ČHMÚ pro svůj monitorovací systém podzemní a povrchové vody.

2.2.3 H40 – GSM/GPRS datalogger, hladinoměr

Sestava vyráběná českou firmou FIEDLER AMS s.r.o., která zahrnuje datalogger H40, ke kterému se nejčastěji připojuje hydrostatický snímač hladiny TSH22 [13]. Sestava hladinoměru je určeno pro sledování hladiny a teploty vody ve vrtu, v jámkách, nádržích a v otevřených tocích. Měřicí sestavu lze dodat v základním provedení bez datových přenosů nebo v provedení se zabudovaným GSM/GPRS+SMS komunikačním modulem pro předávání dat do databáze na server. Hladinoměr je složen z ponorného tlakového snímače TSH22 a z malého dataloggeru H40, ze kterého je tlakový snímač také napájen. Jedná se o robustní přístroje v nerezovém provedení s vysokým krytím, které jsou uzpůsobeny pro trvalý provoz v úzkém a vlhkém prostředí vrtu. Oba přístroje jsou spojeny speciálním kabelem obsahujícím kompenzační kapiláru atmosférického tlaku vzduchu. Data Logger H40 má vlastní napájecí lithiovou baterii 3,6V/19Ah, která díky extrémně nízké proudové spotřebě samotného dataloggeru i připojeného snímače dovoluje až 20 let provozu měřicí sestavy bez výměny baterie. Obvykle se však hydrologger vybavuje GSM/GPRS datovým modulem pro předávání změřených dat na server, a proto je potřeba počítat se zkrácením intervalu pro výměnu baterie u takovéto telemetrické stanice na 8 až 12 let v závislosti na četnosti a objemu odesílaných dat. Hydrologger H40 obsahuje velkou datovou paměť, která pojme data za více než 10 let provozu při obvyklém jednohodinovém intervalu měření. Tento interval je nastavitelný v rozsahu od 1 minuty do 1 dne.

Stejná firma vyrábí také další typy dataloggerů a malých telemetrických stanic, např. Hydrologger H1 nebo telemetrickou stanici Stella.

II. PRAKTICKÁ ČÁST

3 KONCEPČNÍ NÁVRH A ARCHITEKTURA APLIKACÍ

Aby bylo možno udělat koncepční návrh aplikace, je nejdříve nutno analyzovat požadavky, které budou na danou aplikaci kladeny. S tím souvisí i analýza možných vstupních parametrů a také očekávaných výstupů.

3.1 Koncepční návrh

3.1.1 Požadavky na frekvenci měření a ukládání dat

Jak již bylo uvedeno v kapitole 2, jsou požadavky na frekvenci měření v hydrogeologické praxi poměrně různorodé, nejčastěji se však pohybují v intervalu od 1 minuty po 1 den. Minimální interval 1 minuty je nezbytný pro vyhodnocení hydrodynamických zkoušek v podmínkách neustáleného proudění. Proto budeme při návrhu celého systému počítat s intervalem měření 1 minuta.

3.1.2 Volba vhodné přenosové technologie

Výběr vhodné technologie pro přenos dat z hladinoměru na server byl dán již slovy „Internet věcí“ v názvu této práce – zvolená technologie by měla být typicky vyvinutá pro IoT. Stávající systémy využívají pro přenos dat technologie GSM/GPRS, která však tento požadavek nesplňuje. Hydrogeologické vrty se často nacházejí mimo obydlenu zástavbu, a také je potřeba monitorovat různé vrty v rámci celé ČR (někdy i v zahraničí). To na přenosovou technologii klade požadavek dosahu signálu na větší vzdálenosti, s širokým pokrytím v rámci celé ČR. Z technologií uvedených v kapitole 1.1 těmto požadavkům vyhovují technologie LoRa, Sigfox a NB-IoT.

Technologii Sigfox nelze použít vzhledem k omezenému počtu zpráv, které je možno poslat za 24 hodin – 144 zpráv odpovídá intervalu 10 minut, což nevyhovuje požadavku na měření každou minutu. Obě zbývající technologie LoRa a NB-IoT jsou vhodné. Vzhledem k tomu, že NB-IoT je v ČR v počátcích svého vývoje, byla zvolena technologie **LoRa**.

3.1.3 Výběr čidla pro přenos dat

Na trhu je velké množství různých senzorů a čidel, které jsou schopny přenášet data v rámci sítě LoRa. Vzhledem k místu bydliště a také situování hydrogeologického vrtu v Holešově bylo zvoleno čidlo společnosti ELKO EP, s.r.o. se sídlem v Holešově AirIM-100, jehož specifikace je uvedena v katalogovém listu v příloze. Jedná se o univerzální

senzor, který se používá pro hlídání výšky hladiny, teploty, průchodu napětí či proudu, měření baterie, měření energií z domácích měřidel a další [14]. Napájení je pomocí jedné 3,6 V baterie s životností minimálně 5 let dle četnosti vysílaných zpráv. Má impulsní, analogový, binární vstup a svorky pro připojení čidla teploty. Jeho vzhled je patný z obrázku (Obr. 3).



Obr. 3. Čidlo AirIM-100 společnosti ELKO EP, s.r.o. [14].

3.1.4 Hladinoměr

Požadavky na hladinoměr jsou následující:

- Musí umožňovat kontinuální měření
- Musí být možno instalovat ho do hydrogeologického vrtu, do hloubky až cca 100 m
- Musí umožňovat měření proměnlivé výšky vodního sloupce, s rozsahem měření až desítky metrů (možné snížení hladiny při hydrodynamických zkouškách)
- Musí mít možnost napojení na čidlo AirIM-100 (viz kapitola 3.1.3)
- Jelikož přijde do styku s pitnou vodou, měl by splňovat požadavky zdravotní nezávadnosti

Z možných principů měření hladiny podzemní vody, uvedených v kapitole 2.1, byl jako nejlepší vyhodnocen hydrostatický hladinoměr. V této kategorii byl pak vybrán hladinoměr české firmy Dinel, s.r.o. **HLM-25S**, jehož specifikace je uvedena v příloze. Jedná se o hladinoměr, který je primárně určený pro spojitě měření hladiny neagresivních kapalin v beztlakých nádržích, vrtech, studnách apod. Je zde možnost volby libovolného rozsahu



výšky sloupce kapaliny až 100 m nastavitelného po 10 cm. Výstup je proudový nebo napěťový. Hladinoměr má certifikát zdravotní nezávadnosti pro styk s pitnou vodou. Kabel sondy obsahuje vyrovnávací kapiláru [15]. Jeho vzhled je patrný z obrázku (Obr. 4).

Obr. 4. Hladinoměr HLM-25S [15]

Pro naše účely byl zvolen hladinoměr s proudovým výstupem (větší stabilita vůči možnému indukovanému rušení na kabelu), s rozsahem výšky sloupce hladiny 20 m.

3.1.5 Popis procesu sběru dat a jejich přenosu z hladinoměru na server, formát dat

Hladinoměr měří kontinuálně výšku vodního sloupce nad hladinoměrem. Při proudovém výstupu 4 – 20 mA a zvoleném maximálním rozsahu výšky sloupce hladiny 20 m odpovídá sloupec 0 m hodnotě 4 mA, sloupec 20 m hodnotě 20 mA. Pokud je např. vodní sloupec 10 m, bude na výstupu hladinoměru napětí 12 mA.

Čidlo AirIM-100 obsahuje 12 bitový převodník, díky kterému je umožněno při maximálním sloupci 20 m měření s přesností cca 0,5 cm, což je pro naše účely plně postačující. Naměřená hodnota je uložena jako hexadecimální číslo, které je začleněno do paketu přenášených dat. Vysílaný paket přijde na server ve formátu *json* pomocí příkazu POST pro-

tokolu HTTP, kdy v hlavičce je obsažen unikátní token uživatele, v těle potom vlastní data. Struktura dat vypadá následovně:

```
{,data":{"EUI":"020102fffe0e0000","fcnt":868100000,"31on":false,"data":"0a081F66","cmd":{"gw":{"ts":1525421848,"gws":[{"lon":17.5783297,"ts":1525421848,"lat":49.3333095,"snr":7,"gweui":"F03D29FFFF000031","rsi":107}],"freq":868100000,"toa":46,"dr":{"SF7BW1254\\V5},"port":1},"type":"D","tech":"L"}}
```

Význam jednotlivých parametrů shrnuje tabulka č. 1.

Tab. 1. Význam parametrů v těle příkazu POST, formát json

Parametr	Význam parametru
EUI	Device EUI, 64 bitový jednoznačný identifikátor zařízení sítě LoRa
fcnt	Frame counter
ack	Acknowledgement code
data	Payload – část přenášených dat, hlavní účel přenosu
cmd	Identifikuje typ zprávy – přijímají se zprávy s hodnotou gw
gw	gateway – základnová stanice
ts	Timestamp – časové razítko serveru
gws	Pole s objekty popisujícími jednotlivé základnové stanice
lon	Longitude – zeměpisná délka
lat	Latitude – zeměpisná šířka
snr	Signal-to-noise ratio, poměr signálu vůči šumu.
Gweui	Identifikátor základnové stanice
rsi	Received signal strength indication – úhrn přijímaného signálu v radiovém kanálu
freq	Frekvenční pásmo
toa	Time on air – doba vysílání
dr	Spreading factor (činitel rozprostření), přenosová rychlost, kódování
port	Port
type	Typ zprávy
tech	Použitá technologie

Další informace týkající se formátu dat a technologie přenosu LoRa jsou uvedeny v lit. [16].

Vlastní data z měření jsou tedy uložena u parametru „data“ a v uvedeném případě mají hodnotu „0a081F66“, což je hexadecimální číslo o velikosti 4 bytů. Dle specifikace čidla

AirIM-100 první byte (tedy 0a) značí funkci výrobku, druhý byte (08) značí stav zařízení, a ve třetím a čtvrtém bytu (0a28) je uložena vlastní naměřená hodnota (dále v textu označená jako x), a to v μA .

Vzorec, kterým se vypočítá naměřená hodnota vodního sloupce, bude vypadat následovně:

$$h = \left(\frac{x}{1000} - 4 \right) \cdot \frac{h_{max}}{16}$$

h	<i>naměřená výška vodního sloupce (m)</i>
x	<i>hodnota získaná z parametru data (v μA)</i>
h_{max}	<i>maximální výška vodního sloupce hladinoměru (rozsah měření)</i>
1000	<i>představuje převod z μA na mA</i>
4, 16	<i>tato čísla souvisí s rozsahem proudového výstupu 4 – 20 mA</i>

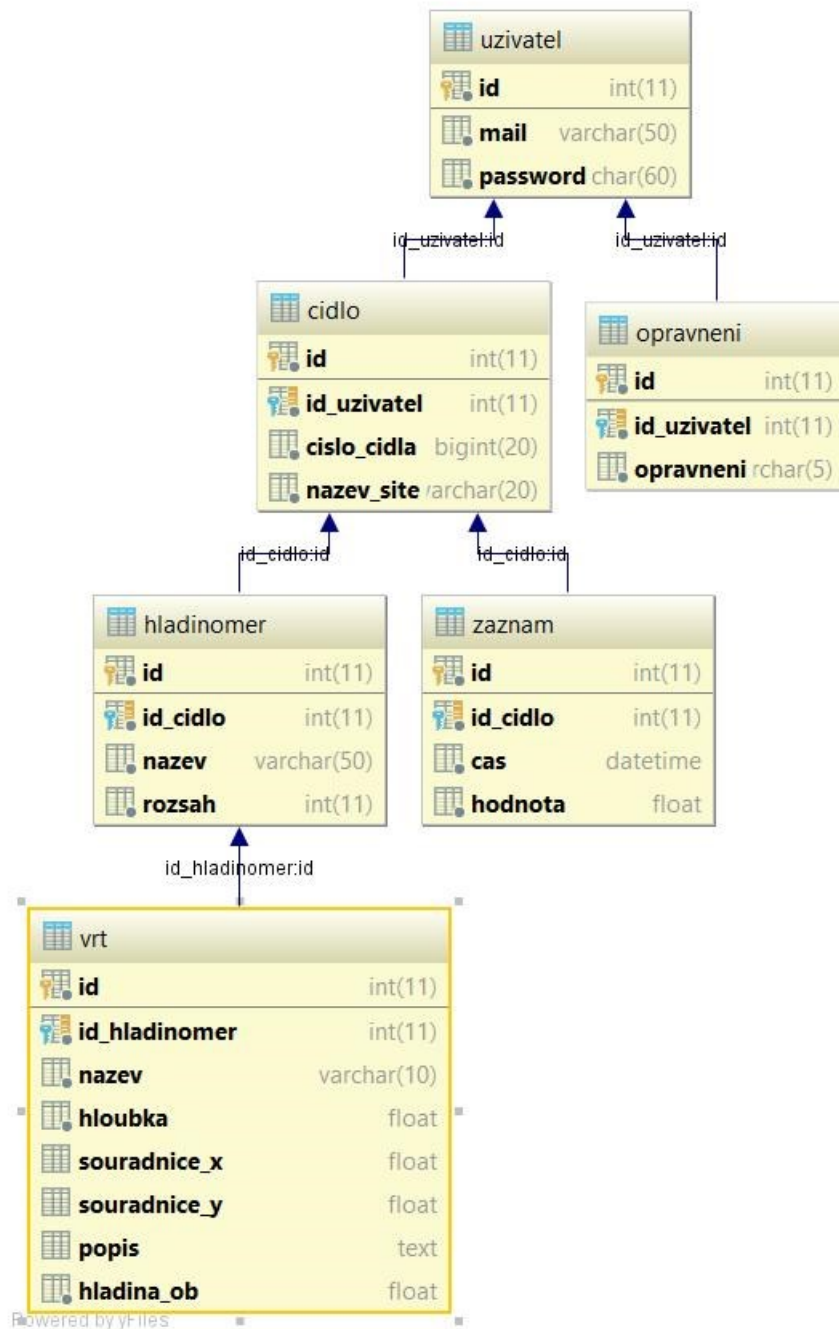
Ve výše uvedeném případě by tedy přijatá hodnota 1F66 (8038 dekadicky) znamenala výšku vodního sloupce nad sondou 5,05 m.

Pro potřeby ukládání dat na server a jejich dalšího využití bude nutno z výše uvedeného těla příkazu post získat kromě naměřené hodnoty také časovou značku (ts), abychom věděli, kdy byl údaj změřen, a dále hodnotu EUI, která jednoznačně identifikuje dané zařízení (v našem případě čidlo AirIM-100) a umožní tak získaná data přiřadit příslušnému uživateli.

3.1.6 Struktura databáze pro ukládání dat

Pro potřeby ukládání dat bylo vytvořeno několik databázových tabulek. Jedná se o tabulky Uživatel, Opraveni, Vrt, Hladinomer, Cidlo a Zaznam. Vycházíme z předpokladu, že aplikaci může využívat větší počet uživatelů. Každý z nich může provádět měření na více vrtech současně a může mít tomu odpovídající počet hladinoměrů a s nimi spojených čidel. Jednomu čidlu odpovídá jeden hladinoměr. Každý přenesený záznam měření z nějakého vrtu je vždy přiřazen jednomu konkrétnímu čidlu, které je spojeno s konkrétním hladinoměrem. Tato sestava čidla a hladinoměru může provádět více měření po sobě na různých vrtech.

Graficky jsou jednotlivé databázové tabulky znázorněny na obrázku (Obr. 5).



Obr. 5. Schéma databázových tabulek

3.1.6.1 Tabulka UZIVATEL

Tato tabulka obsahuje základní údaje o jednotlivých uživateliích a má atributy id, mail a password. Primárním klíčem je celočíselné automaticky generované *id* typu int. Textový atribut *mail* obsahuje mailovou adresu uživatele a je typu *varchar*. Atribut *password* obsahuje heslo uložené pomocí hashovací funkce MD5 typu *varchar*.

Do této tabulky se ukládají data z aplikace automaticky při registraci uživatele.

3.1.6.2 Tabulka OPRAVNENI

V této tabulce jsou uloženy informace o oprávnění jednotlivých uživatelů. Obsahuje atribut *id*, *id_uzivatel* a *opraveneni*. Primárním klíčem je celočíselné automaticky generované *id* typu *int*, cizím klíčem do tabulky UZIVATEL je *id_uzivatel*. Atribut *opraveneni* je textový atribut pro název oprávnění s omezením na 5 znaků. Primárně předpokládáme, že hodnota tohoto atributu bude buď „*admin*“ nebo „*user*“.

3.1.6.3 Tabulka CIDLO

Tato tabulka obsahuje údaje o čidlu a má 4 atributy – automaticky generovaný primární klíč *id*, cizí klíč do tabulky UZIVATEL *id_uzivatel*, celočíselné identifikační číslo čidla nazvané *cislo_cidla* odpovídající EUI ve formátu bigint a textový atribut informačního charakteru s omezenou délkou 20 znaků *nazev_site*, do kterého se vyplňuje název technologie použité pro bezdrátový přenos dat – implicitně je nastavena LoRa.

Do této tabulky nemá uživatel přístup a vyplňuje ji pouze administrátor.

3.1.6.4 Tabulka ZAZNAM

Do této tabulky se automaticky ukládají všechny záznamy měření z jednotlivých sond. Obsahuje atributy *id*, *id_cidlo*, *cas* a *hodnota*.

Primárním klíčem je celočíselné automaticky generované *id* typu *int*. *Id_cidlo* je cizím klíčem do tabulky CIDLO. *Cas* je atribut typu *datetime*, ve kterém je uložený přesný čas provedeného měření, a v atributu *hodnota* je uložena naměřená hodnota v mA.

Data do této tabulky jsou ukládána automaticky pomocí aplikace zajišťující ukládání dat (viz kapitola 3.2.1).

3.1.6.5 Tabulka HLADINOMER

V této tabulce jsou uloženy údaje o hladinoměru. Obsahuje atributy *id*, *id_cidlo*, *nazev*, *rozsah*.

Primárním klíčem je celočíselné automaticky generované *id* typu *int*. *Id_cidlo* je cizím klíčem do tabulky CIDLO. Textový atribut *nazev* je určen pro název hladinoměru s omezením na 50 znaků. Implicitně je zde nastaven hladinoměr Dinel HLM-25S. Atribut *rozsah* je celočíselná hodnota typu *int*, ve které je uložen měřicí rozsah hladinoměru (ma-

ximální mocnost vodního sloupce, který je sonda schopna měřit). Tato hodnota se v aplikaci používá pro přepočítání hodnoty napětí naměřeného hladinoměrem na konkrétní výšku vodního sloupce.

Údaje do této tabulky vyplňuje administrátor, nicméně uživatel je může pomocí aplikace editovat.

3.1.6.6 Tabulka VRT

V tabulce VRT jsou shromážděny všechny informace, týkající se vrtu, ve kterém je instalovaný hladinoměr. Obsahuje atributy *id*, *id_hladinomer*, *nazev*, *hloubka*, *souradnice_x*, *souradnice_y*, *popis* a *hladina_ob*.

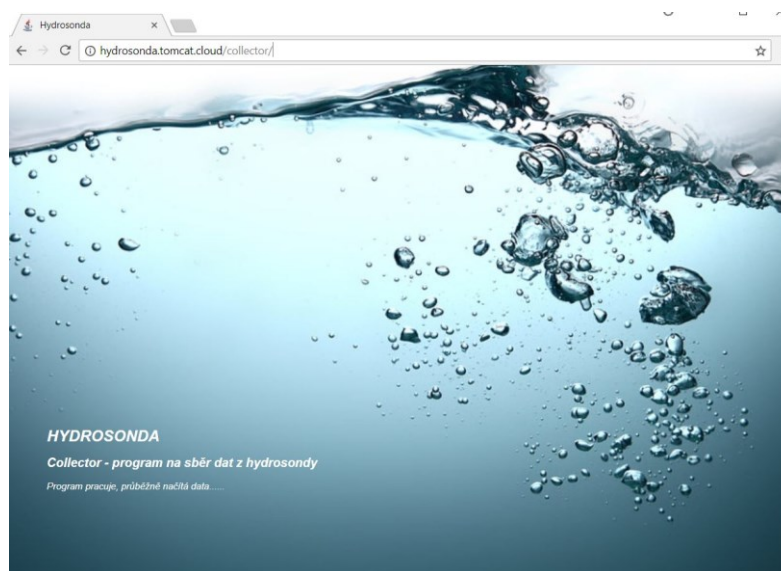
Primárním klíčem je celočíselné automaticky generované *id* typu *int*. *Id_hladinomer* je cizím klíčem do tabulky HLADINOMER. *Nazev* je textový atribut pro název vrtu s omezením na 10 znaků. Atribut *hloubka* typu *float* udává hloubku vrtu v metrech (předpokládané zaokrouhlení je na centimetry, tedy 2 desetinná místa). *Souradnice_x* a *souradnice_y* jsou opět typu *float* a udávají souřadnice vrtu ve formátu S-JTSK v metrech, opět se zaokrouhlením na centimetry. Atribut *popis* slouží k textovému popisu vrtu, například k jeho umístění, účelu, prováděným zkouškám atd. a je omezen na 1000 znaků. Atribut *hladina_ob* je typu *float* a je v něm uložena hloubka hladiny podzemní vody od odměrného bodu (OB – většinou okraj zhlaví vrtu) v metrech, opět s předpokládaným zaokrouhlením na centimetry. Tento údaj může být využit pro přepočítání výšky vodního sloupce nad hladinoměrem na metry od OB.

Přístup do této tabulky je povolen jednotlivým uživatelům, kteří si zde mohou vytvářet nové záznamy pro jednotlivé vrty. Atribut *id* se generuje automaticky. Atribut *id_hladinomer* se generuje automaticky na základě předchozího výběru čidla, ke kterému bude vrt patřit.

3.2 Architektura aplikací

3.2.1 Aplikace zajišťující ukládání dat na server – program Collector

Tato aplikace je navržena tak, že pouze přijímá data posílaná z čidla, a s uživatelem nekomunikuje. Je nasazena na adrese /collector/ a po jejím zadání se pouze zobrazí stránka, oznamující jméno programu. Tato stránka je ukázána na obrázku (Obr. 6).



Obr. 6. Úvodní webová stránka programu Collector

Posílaná data je nutno směřovat na url adresu /collector/feed. Před zahájením posílání dat je nutné provést registraci v aplikaci pro publikování dat, a znát EUI čidla. Poté administrátor provede v databázi přiřazení EUI k příslušnému uživateli. Bez tohoto přiřazení nebudou záznamy ukládány. Aplikace testuje přijaté EUI a hledá v databázi uživatele, který k tomuto čidlu patří. Pokud ho nenalezne, data neuloží. Kromě čidla přiřadí admin zaregistrovanému uživateli také hladinoměr.

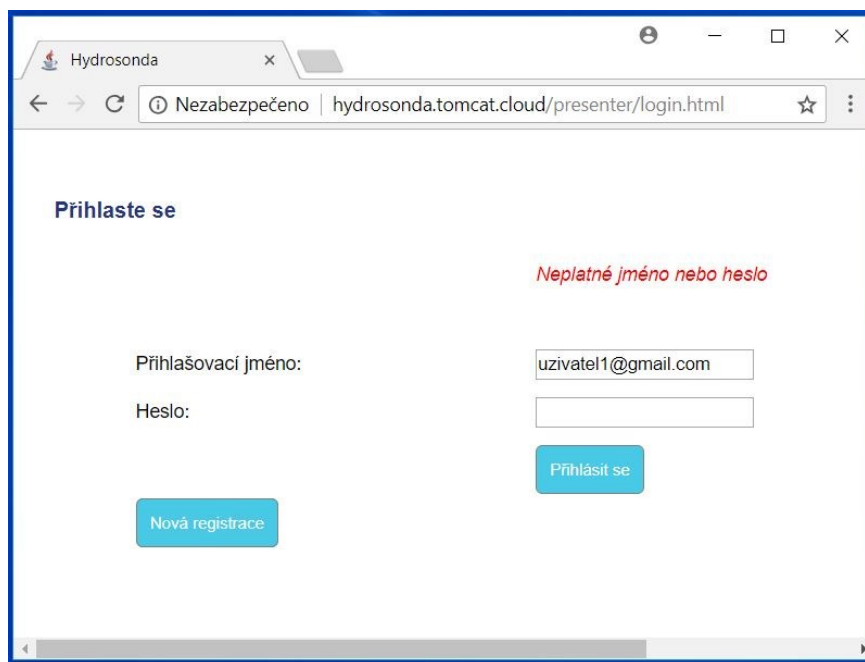
Zde je nutno poznamenat, že v reálném provozu tato aplikace musí běžet na spolehlivém serveru, který nevypadává a kde nehrozí přerušení jeho činnosti. Pokud by docházelo k výpadkům, budou data vzhledem k použité technologii navždy ztracena.

Formát přijímaných dat (JSON) je popsán v kapitole 3.1.5. V těchto datech aplikace najde naměřené napětí (μA) v hexadecimálním tvaru, které uloží do databáze. Do databáze je dále uloženo EUI a časový údaj přiřazený k měření (timestamp). Aplikace pracuje pouze s tabulkou ZAZNAM.

3.2.2 Aplikace pro publikování dat

3.2.2.1 Přihlašovací a registrační formuláře

Tato aplikace slouží k zadávání některých dat do databáze a především k jejich prohlížení. Je nasazena na adrese /presenter/. Po zadání adresy do prohlížeče se objeví úvodní okno, pomocí kterého dojde k přesměrování do přihlašovacího formuláře (Obr. 7).



The screenshot shows a web browser window with the following elements:

- Browser tab: Hydrosonda
- Address bar: hydrosonda.tomcat.cloud/presenter/login.html
- Page title: Přihlaste se
- Error message: Neplatné jméno nebo heslo
- Form fields:
 - Přihlašovací jméno: uzivatel1@gmail.com
 - Heslo: (empty)
- Buttons: Přihlásit se, Nová registrace

Obr. 7. Přihlašovací formulář

Jako přihlašovací jméno slouží vždy emailová adresa. Pokud je zadané přihlašovací jméno, které není uloženo v databázi, nebo je zadáno špatné heslo, objeví se chybové hlášení. Pokud ještě není uživatel zaregistrovaný, může provést registraci s využitím tlačítka *Nová registrace* (Obr. 8).

The image shows a web browser window with the title 'Hydrosonda'. The address bar displays 'Nezabezpečeno | hydrosonda.tomcat.cloud/presenter/registrace.html'. The main content area is titled 'Zaregistrujte se'. It features three input fields for registration: 'Email' (containing '1245'), 'Heslo', and 'Heslo (znovu)'. To the right of each field is a red error message: 'Zadejte platnou e-mailovou adresu', 'Zadejte heslo (minimálně 6 znaků)', and 'Potvrďte prosím zadané heslo (minimálně 6 znaků)'. A red error message at the top right says 'Požadované údaje nejsou zadány správně'. A blue button labeled 'Zaregistrovat se' is at the bottom.

Obr. 8. Registrační formulář

Registrační údaje jsou programem validovány. Email adresa musí být ve tvaru emailové adresy, heslo vyžaduje alespoň 6 znaků a musí být opakovaně zadáno správně. Pokud je registrace úspěšná, provede program přesměrování zpět na přihlašovací formulář, kde se pokračuje přihlášením.

3.2.2.2 Editační formuláře

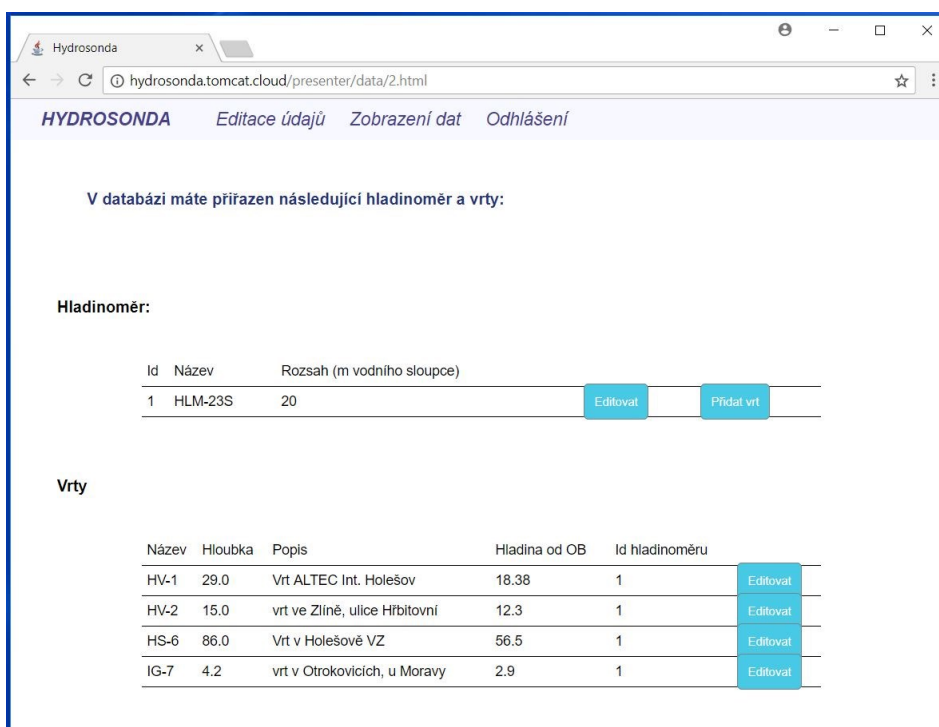
Po úspěšném přihlášení budeme přesměrováni na úvodní stránku programu, na které je zobrazeno pouze menu. Zde máme na výběr z *Editace údajů*, kdy můžeme upravovat informace o přiřazeném hladinoměru a zadávat a editovat vrt(y), ve kterých probíhá měření. Další položkou menu je *Zobrazení dat*, pomocí níž si můžeme prohlížet naměřená data. Poslední položkou menu je *Odhlášení*. Toto menu je dostupné ze všech dalších stránek programu.

Při volbě položky *Editace údajů* budeme přesměrováni na stránku, kde se provádí výběr čidla, ke kterému jsou přiřazeny hladinoměry a vrty, které chceme editovat nebo přidávat. Jeden uživatel může mít přiřazených více čidel (systém čidlo – hladinoměr). Na obrázku (Obr. 9) je vidět pouze možnost jedné volby – jedná se o aktuálně měřící soustavu hladinoměr-čidlo, která je ve zkušebním provozu v reálném vrtu v Holešově (viz. kapitola 5).



Obr. 9. Výběr čidla

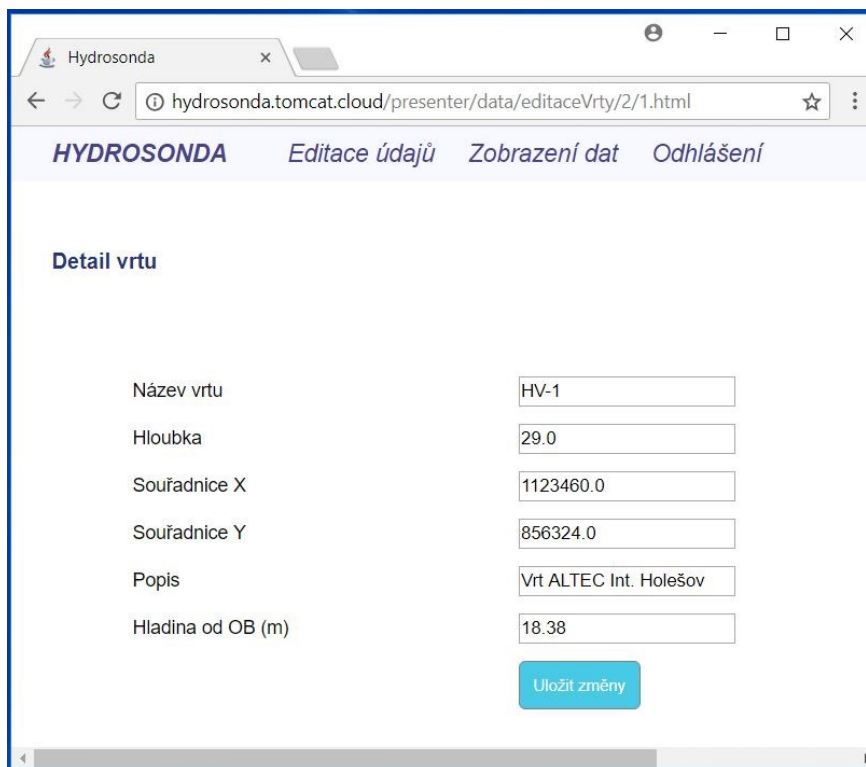
Výběrem konkrétního čidla se dostaneme na stránku editace (Obr. 10). Zde se nám zobrazí hladinoměr, který je přiřazený k danému čidlu, a dále seznam všech vrtů, které jsou přiřazeny k hladinoměru



Obr. 10. Editace

Vlastní hladinoměr můžeme editovat - měnit název a položku rozsah, která se využívá k přepočtu naměřené hodnoty napětí na velikost sloupce vody. Dále můžeme k danému

hladinoměru přiřadit nový vrt, nebo editovat vrt stávající. Na obrázku (Obr. 11) je ukázka formuláře pro editování vrtu; ostatní formuláře jsou obdobné.



The screenshot shows a web browser window with the URL `hydrosonda.tomcat.cloud/presenter/data/editaceVrty/2/1.html`. The page title is 'HYDROSONDA' and the navigation menu includes 'Editace údajů', 'Zobrazení dat', and 'Odhlášení'. The main content area is titled 'Detail vrtu' and contains a form with the following fields:

Název vrtu	<input type="text" value="HV-1"/>
Hloubka	<input type="text" value="29.0"/>
Souřadnice X	<input type="text" value="1123460.0"/>
Souřadnice Y	<input type="text" value="856324.0"/>
Popis	<input type="text" value="Vrt ALTEC Int. Holešov"/>
Hladina od OB (m)	<input type="text" value="18.38"/>

At the bottom right of the form is a blue button labeled 'Uložit změny'.

Obr. 11. Formulář editace vrtu

3.2.2.3 Zobrazení dat

Po volbě položky menu Zobrazení dat budeme přesměrováni na stránku, kde se provádí výběr čidla, ke kterému chceme zobrazit příslušná data. Tato stránka vypadá téměř totožně jako v případě editace údajů (Obr. 9). Po volbě čidla se zobrazí souhrnná stránka s přehledem měření, odpovídajících tomuto čidlu, a jejíž části můžeme vidět na obrázcích (Obr. 12, Obr. 13).

V databázi máte přiřazen následující hladinoměry a vrty:

Hladinoměry:

T 1	T 2	T celk	H max	H min	H prum	SO
18.5.18 8:55	24.5.18 4:56	140:00:26	5.07	-4.99	4,6	2.14

T 1 - datum a čas zahájení měření
 T 2 - datum a čas ukončení měření (posledního zápisu do databáze)
 T celk - celková doba měření (hh:mm:ss)
 H max - maximální naměřená hodnota (m vodního sloupce)
 H min - minimální naměřená hodnota (m vodního sloupce)
 H prum - průměrná naměřená hodnota (m vodního sloupce)
 SO - Směrodatná odchylka měření

Vrty

Název	Hloubka	Popis	Hladina od OB	Id hladinoměru
HV-1	29.0	Vrt ALTEC Int. Holešov	18.38	1
HV-2	15.0	vrt ve Zlíně, ulice Hřbitovní	12.3	1
HS-6	86.0	Vrt v Holešově VZ	56.5	1
IG-7	4.2	vrt v Otrokovicích, u Moravy	2.9	1

Obr. 12. Zobrazení statistických údajů

Na obrázku (Obr. 12) jsou vidět vypočtené statistiky – začátek a konec měření (jako konec se bere doba posledního aktuálního záznamu), doba trvání měření, minimální, maximální a průměrná naměřená hodnota a směrodatná odchylka. Veškeré hodnoty se s každým obnovením prohlížeče průběžně přepočítávají. Dále je zde zobrazena tabulka vrtů příslušejících danému hladinoměru. Pod touto tabulkou pak začíná tabulka s konkrétními záznamy úrovně hladiny (vodního sloupce), jak je vidět na obrázku (Obr. 13).

Seznam měření příslušejících k hladinoměru

Čas měření	Hodnota (m vodního sloupce)
18.5.18 8:55	0.04
18.5.18 8:56	0.02
18.5.18 8:57	0.02
18.5.18 8:58	0.03
18.5.18 8:59	0.03
18.5.18 9:00	0.02
18.5.18 9:01	0.02
18.5.18 9:02	0.02
18.5.18 9:03	0.03
18.5.18 9:04	0.02
18.5.18 9:05	0.03
18.5.18 9:06	0.02
18.5.18 9:07	0.04

Obr. 13. Zobrazení naměřených dat

3.2.2.4 Odhlášení

Poslední položka menu zahrnuje odhlášení. Při její volbě dojde k odhlášení uživatele a k přesměrování programu do přihlašovacího formuláře.

4 IMPLEMENTACE

Pro požadované účely bylo nutno implementovat 2 samostatné aplikace. První z nich zachytává data, která přicházejí ze sondy HTTP metodou POST, a ukládá je do databáze. Pracuje pouze s jednou tabulkou vytvořené databáze (tabulka *zaznam*). Vrací pouze úvodní webovou stránku s informací o programu. Druhá aplikace umožňuje registraci uživatelů, jejich přihlášení, vkládání a úpravu některých dat a také zobrazování těchto dat, včetně dat uložených první aplikací.

Obě aplikace jsou typu server-side rendering, což znamená, že webové stránky jsou generovány na straně serveru

4.1 Použité technologie

4.1.1 Jazyk JAVA

Jazyk Java je v současnosti jedním z nejpobulárnějších programovacích jazyků, který je stále vyvíjen a vylepšován. Jedná se o objektově orientovaný jazyk, jehož základ tvoří Java Standard Edition (JSE), která se používá v běžných desktopových aplikacích. Jejím rozšířením je potom Java Enterprise Edition (JEE) [17], která je zaměřená na tvorbu webových aplikací v Javě. Jedná se vlastně o sadu knihoven, která rozšiřuje edici JSE [18].

JEE funguje na architektuře klient-server. Java běží na straně serveru a zpracovává požadavky, které jsou zasílány od klienta (HTML stránky), a výstupem je opět HTML stránka, kterou pošle zpět klientovi.

4.1.2 Spring framework

Spring framework je vícevrstvý aplikační rámec, který usnadňuje vývoj JEE aplikací. Patří do třídy tzv. lightweight (odlehčených) JEE kontejnerů [19].

Spring je rozdělený do jednotlivých modulů, přičemž je možno vybrat si jen ty moduly, které jsou potřeba. Základ tvoří tzv. „Core“ kontejner s moduly Beans, Core, Context a Expressions, které zahrnují konfigurační model a mechanismus „dependency injection“. Skupina modulů „Data Access/Integration“ obsahuje kromě jiných také modul JDBC, který umožňuje přístup k databázím. K dalším skupinám modulů patří Web, AOP (Aspect Oriented Programming), Instrumentation a Test.

Jádro Springu je postaveno na využití návrhového vzoru Inversion of Control, který řeší vazby mezi jednotlivými komponentami. Díky Spring frameworku je možno vytvářet program jako skupinu vzájemně nezávislých komponent, které se propojí až po spuštění programu. Způsob propojení se určuje pomocí XML souboru, takže jednotlivé závislosti není nutné psát do zdrojového kódu.

Modul Web-Servlet ze skupiny Web obsahuje Springovou implementaci Model View Controlleru (MVC), což je jeden z nejnámějších návrhových vzorů. Jeho hlavní myšlenkou je oddělení výpočetní logiky od logiky prezenční. Architektura MVC dělí aplikaci na 3 logické části Model, View a Controller, které lze upravovat samostatně. Model reprezentuje data a informace, s nimiž aplikace pracuje a které jsou prostřednictvím uživatelského rozhraní zpřístupněny uživateli. View zpřístupňuje informace reprezentované modelem uživateli a Controller řídí celý proces, reaguje na události a zajišťuje změny v modelu nebo pohledu (view) [20].

Pro rychlé spuštění celé aplikace byl využit modul Spring Boot, který slouží k tvorbě soběstačných (stand-alone) aplikací s minimální nutností konfigurace. Tento modul již v sobě obsahuje webserver Tomcat, Jetty nebo Undertow, díky čemuž odpadá nutnost nasazování souboru war. Dále obsahuje startovací konfiguraci POM (Project Object Model) souboru, čímž zjednodušuje konfiguraci Mavenu, a automaticky při každé vhodné příležitosti konfiguruje Spring. Také nabízí předpřipravené nástroje na měření metrik, kontroly aktivity a externí nastavení. Funguje bez nutnosti jakéhokoliv generování kódu a XML konfigurací [19].

4.1.3 Apache Tomcat

Apache Tomcat je webový server a servlet kontejner vyvíjený společností Apache Software Foundation jako open source projekt, který je založený na jazyce Java [21]. Je jednoduchý, transparentní, s poměrně malou náročností na výpočetní výkon. V současné době je to jeden z nejpoužívanějších serverů pro Javu. Při vývoji obou aplikací funguje jako server, na kterém tyto aplikace běží.

4.1.4 Apache Maven

Maven je nástroj od společnosti Apache Software Foundation, který slouží ke správě a sestavování aplikací postavených nad platformou Java [22]. Jeho využitím odpadá závis-

lost na konkrétním IDE (vývojovém prostředí), přičemž jeho formát je akceptován všemi vývojovými prostředími Javy.

Základním principem fungování Mavenu je popis projektu pomocí Project Object Model (POM). Tento model popisuje softwarový projekt včetně jeho závislostí na externích knihovnách a různých funkcí s tím spojených. Popis je uložen v souboru pom.xml v kořenovém adresáři projektu. Jednou ze zásadních výhod je řešení závislostí – závislosti na externích knihovnách jsou uloženy v tomto souboru v tzv. „Dependencies“.

Maven odděluje zdrojové kódy tříd od ostatních souborů, jako jsou konfigurační soubory, šablony, obrázky apod. Maven také umožňuje rozdělit proces buildu do více fází, což je výhodné např. během vývoje aplikace.

4.1.5 Šablonovací systém Thymeleaf

Thymeleaf je moderní server-side šablonovací nástroj. Jeho hlavním cílem je vytvářet šablony, které budou korektně zobrazovány ve webových brawserech a zároveň budou sloužit jako statické prototypy, což umožní těsnou spolupráci vývojových týmů (spolupráce backend a frontem vývojářů). Je obdobou známějšího nástroje JSP. Výhodou je snadná integrace se Spring MVC a jeho rozšiřitelnost. Funguje tak, že do tagů již vytvořené šablony webové aplikace, která je vytvořena pomocí jazyka HTML, se doplní nové atributy, pomocí kterých jsou šablonovacím nástrojem doplněna nová data [23].

4.1.6 Databáze Maria DB

MariaDB je relační databáze, která je odnoží databáze MySQL, přičemž většina funkcí obou databází je stejná. Je vyvíjena původními vývojáři MySQL jako open source. Je rychlá, škálovatelná a robustní, s množstvím nástrojů a pluginů, což ji předurčuje k širokému použití. Poslední verze MariaDB zahrnuje také nástroje pro GIS a JSON [24].

4.1.7 Vývojové prostředí IntelliJ IDEA Ultimate

IntelliJ IDEA Ultimate je vývojové prostředí společnosti JetBrains pro jazyk Java, které umožňuje programovat webové aplikace. Má podporu všech výše zmíněných technologií.

4.2 Vstupní konfigurace

Při programování obou aplikací bylo potřeba nastavit konfigurační soubory. Jedná se o soubor pom.xml, ve kterém je uložena konfigurace Mavenu, a dále konfigurace Springu

uložená v souboru application.properties. Výsledná konfigurace obou těchto souborů je součástí přílohy na CD. Jelikož byl využit SpringBoot, probíhalo spouštění programu v rámci jeho vytváření a ladění pomocí interního serveru Tomcat, který je zabudován přímo v aplikaci.

4.3 Aplikace pro ukládání dat na serveru

Tato aplikace je poměrně jednoduchá. Jelikož pouze přijímá data, není potřeba vytvářet šablony pro komunikaci s uživatelem. Důležitých je pouze 6 následujících javových tříd:

- Collector – obsahuje spouštěcí metodu
- MainController – třída, řídí celý proces, reaguje na události a zajišťuje změny. Stará se o mapování webových stránek
- Zaznam, Cidlo – entity, které definují vlastnosti objektů Zaznam a Cidlo.
- ZaznamRepository, CidloRepository – obsahuje definici databázového připojení a všechny metody, které jsou potřebné pro práci s databázovými tabulkami ZAZNAM a CIDLO.

4.3.1 MainController

Tato třída obsahuje základní metodu `processFeed`, která zachytává data přicházející z čidla formou příkazu POST protokolu http ve formátu JSON. Z těchto dat metoda vybírá hodnoty odpovídající EUI, času (timestamp) a naměřeným datům a převádí je na vhodné formáty dat. Výsledná získaná data pak uloží do databáze. Metoda vrací šablonu "`feed-template`", která však není v programu využita.

```
@RequestMapping(value = "/feed", method = RequestMethod.POST)
public ModelAndView processFeed(@RequestBody String text, @RequestHeader("token") String
token) {

    int prvniIndex;

    prvniIndex = text.lastIndexOf("EUI");
    Long cisloCidla = dataCi-
dlo.findOne(Long.valueOf(text.substring(prvniIndex+8, prvniIndex+24), 16)).getId();

    prvniIndex = text.lastIndexOf("ts") ;
    LocalDateTime casMereni =
        LocalDateTime
me.ofInstant(Instant.ofEpochSecond(Long.valueOf(text.substring(prvniIndex+5, prvniIndex+15))),
        TimeZone.getDefault().toZoneId());

    prvniIndex = text.lastIndexOf("data");
    int hladina = Integer.valueOf(text.substring(prvniIndex+13, prvniIndex+17), 16);

    Zaznam zaznam = new Zaznam(cisloCidla, casMereni, hladina);
    dataSupplier.save(zaznam);

    ModelAndView dataHolder = new ModelAndView("feed-template");
```

```

    return dataHolder;
}

```

4.3.2 Repository

V Repository jsou definovány metody a proměnné, které jsou nutné pro práci s databází. K připojení k databázi je určena základní třída JdbcTemplate, jejíž použití dokumentuje následující kód pro třídu **ZaznamRepository**. Kromě databázového připojení v konstruktoru třídy jsou zde definovány metody pro ukládání dat save(), update() a add().

```

public class ZaznamRepository {

    private DataSource databaseConfiguration;
    private JdbcTemplate requestSender;
    private RowMapper<Zaznam> converter;

    public ZaznamRepository() {
        try {
            MariaDbDataSource databaseConfiguration = new MariaDbDataSource();
            databaseConfiguration.setUsername("student");
            databaseConfiguration.setPassword("password");
            databaseConfiguration.setUrl("jdbc:mysql://localhost:3306/Hydrosonda_new");

            requestSender = new JdbcTemplate(databaseConfiguration);
            converter = BeanPropertyRowMapper.newInstance(Zaznam.class);
        } catch (SQLException ex) {
            throw new RuntimeException(ex);
        }
    }

    public Zaznam save(Zaznam zaznamToSave) {
        if (zaznamToSave.getId() != null) {
            return update(zaznamToSave);
        } else {
            return add(zaznamToSave);
        }
    }

    private Zaznam update(Zaznam zaznamToSave) {
        //Zaznam udaj = clone(zaznamToSave);
        requestSender.update(
            "UPDATE Zaznam SET id_cidlo = ?, cas = ?, hodnota = ? WHERE id_zaznam = ?",
            zaznamToSave.getId_cidlo(), zaznamToSave.getCas(), zaznam-
            ToSave.getHodnota(), zaznamToSave.getId_cidlo());
        return zaznamToSave;
    }

    private Zaznam add(Zaznam zaznamToAdd) {
        GeneratedKeyHolder keyHolder = new GeneratedKeyHolder();
        String sql = "INSERT INTO Zaznam (id_cidlo, cas, hodnota) VALUES (?, ?, ?)";
        requestSender.update((Connection con) -> {
            PreparedStatement command = con.prepareStatement(sql, Sta-
            tement.RETURN_GENERATED_KEYS);
            command.setLong(1, zaznamToAdd.getId_cidlo());
            command.setObject(2, zaznamToAdd.getCas());
            command.setFloat(3, zaznamToAdd.getHodnota());
            return command;
        },
            keyHolder);
        zaznamToAdd.setId(keyHolder.getKey().longValue());
        return (zaznamToAdd);
    }
}

```

Třída CidloRepository obsahuje kromě konstruktoru pouze metodu findOne(), která vrátí konkrétní čidlo podle jeho EUI.

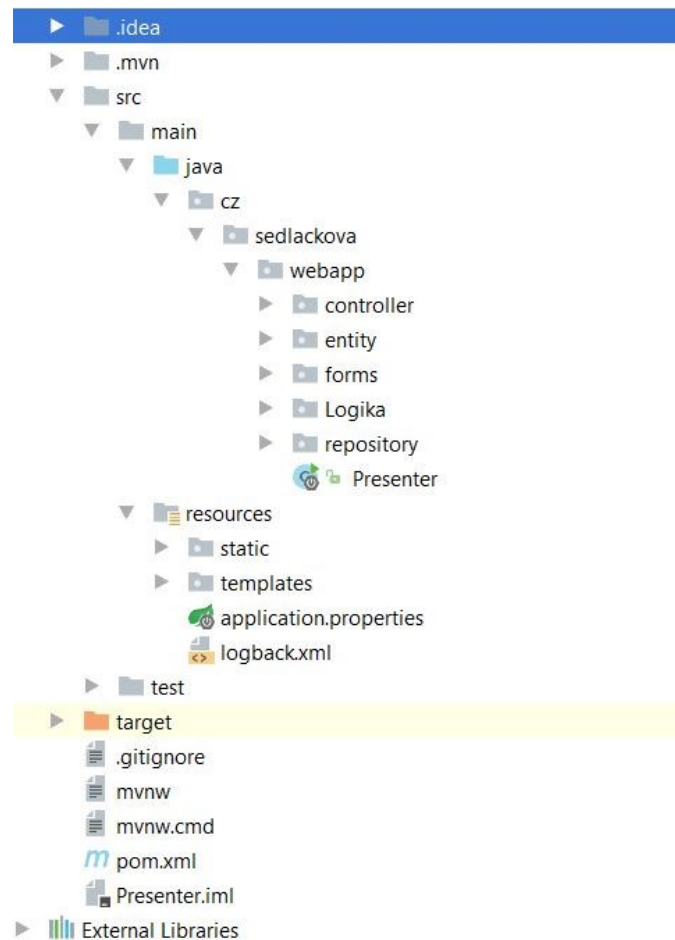
```
public Cidlo findOne(Long id) {  
    Cidlo cidlo = requestSender.queryForObject("select id, id_uzivatel, cislo_cidla, na-  
zev_site" +  
        " from cidlo where cislo_cidla=?", converter, id);  
    return cidlo;  
}
```

4.3.3 Entity

Bylo nutno vytvořit entity odpovídající databázovým tabulkám ZAZNAM a CIDLO. Tyto entity obsahují pouze definici proměnných, vygenerované getry a setry, konstruktory a metody toString(), hash() a equals.

4.4 Aplikace pro publikování dat

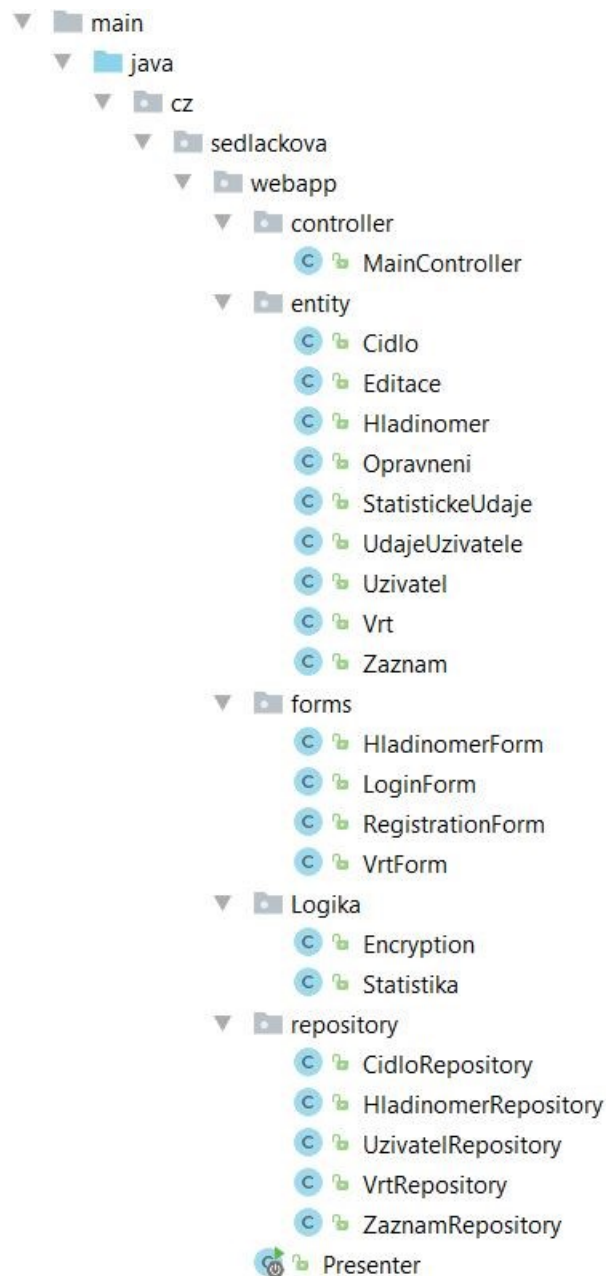
Aplikace pro publikování dat je podstatně rozsáhlejší a složitější, než aplikace předchozí. Je zde vytvořeno velké množství souborů, které spadají do 2 hlavních kategorií, jak vyplývá z obrázku (Obr. 14) – javové soubory a šablony (templates). Javové soubory jsou uloženy v balíčku webapp, šablony ve složce resources – templates. Vše je doplněno statickými soubory ve složce resources-static, která obsahuje soubor .css a obrázek s ikonou, a dále soubory konfiguračními. V dalším textu jsou uvedeny pouze příklady kódu. Celý zdrojový kód je součástí přílohy.



Obr. 14. Struktura balíčků a adresářů

4.4.1 Javové soubory

Na obrázku (Obr. 14) je zachycena struktura uložení všech javových tříd, které jsou rozdělené do tzv. „packages“ (balíčků) neboli jmenných prostorů, které odpovídají jednotlivým adresářům. Jedná se o balíčky *controller*, *entity*, *forms*, *logika* a *repository*. Samostatně potom stojí *Presenter*, který obsahuje spouštěcí třídu.



Obr. 15. Struktura uspořádání javových tříd

4.4.1.1 Balíček controller

V tomto balíčku je uložena pouze jedna třída, **MainController**, která se díky Springové anotaci `@Controller` stará o mapování požadovaných url adres na příslušné šablony s využitím anotace `@RequestMapping()`. Návrátovým typem jednotlivých funkcí je `ModelAndView`, který vrací současně „Model“, tedy zpracovaná data, a „View“ – webovou stránku, pomocí níž se tato data zobrazují (server-side rendering). Pro každou webovou stránku, která se v aplikaci používá, musí být v této třídě provedeno namapování na pří-

slušnou šablonu. Přehled všech webových stránek a jim odpovídajících šablon je uveden v tabulce (Tab. 2).

Tab. 2. Namapované webové stránky (Controller) a odpovídající šablony

Namapovaná webová stránka	Odpovídající šablona	Popis, přesměrování
/	login/index	Úvodní stránka aplikace
/login.html GET	login/login	Přihlašovací formulář
/login.html POST		Přesměrování na /data.html v případě úspěšného přihlášení
/login.html POST	login/login	V případě neúspěšného přihlášení
/registrace.html GET	login/registration	Registrační formulář
/registrace.html POST	login/registration	V případě chybné registrace
/registrace.html POST		Přesměrování na /registrace-uspesna.html
/registrace-uspesna.html	login/registrationOK	Potvrzovací stránka o úspěšném přihlášení
/data.html	data/index	Úvodní stránka s menu
/editace-seznam.html	data/editace	Stránka se seznamem čidel
/data/{cislo}.html	data/editaceHladVrty	Stránka s možností volby editace hladinoměru a vrtů a přidání vrtů
/data/editaceHladinomer/{cislo}.html GET	data/detailHladinomeru	Stránka s možností editace hladinoměru
/data/editaceHladinomer/{cislo}.html POST		Přesměrování na /data/{cislo}.html
/data/editaceVrty/{cislo1}/{cislo2}.html GET	data/detailVrtu	Stránka s možností editace vrtu
/data/editaceVrty/{cislo1}/{cislo2}.html POST		Přesměrování na /data/{cislo}.html
/data/newVrt/{cislo1}/{cislo2}.html GET	data/novyVrt	Stránka s možností zadání nového vrtu
/data/newVrt/{cislo1}/{cislo2}.html POST		Přesměrování na /data/{cislo}.html
/zobrazeni-seznam.html	data/zobrazeni	Stránka se seznamem čidel
/data/zobrazeni/{cislo}.html	data/statistika	Stránka se zobrazením statistických údajů a naměřených dat

Jako ukázkou jedné z metod se springovou anotací zde uvádím metodu addRegistration() s parametrem POST, která reaguje na vyplněný registrační formulář (ten se předtím zavolal

stejnou metodou, ale s parametrem GET). Kromě již zmíněné anotace `@RequestMapping` jsou zde použity také anotace `@Valid` a `@ModelAttribute`, které souvisejí s validací vstupních údajů. Pokud se do formuláře zadají data, která neodpovídají parametrům nastaveným ve třídě `RegistrationForm`, objeví se chybová hlášení, jejichž text je také nastaven v této třídě.

```
@RequestMapping(value="/registrace.html", method = RequestMethod.POST)
public ModelAndView addRegistration(@Valid @ModelAttribute("registrace") RegistrationForm
    vyplnenyFormular, BindingResult validationErrors) {

    RegistrationForm uzivatel = new RegistrationForm();
    uzivatel.setMail(vyplnenyFormular.getMail());
    uzivatel.setPassword(vyplnenyFormular.getPassword());
    uzivatel.setPasswordAgain(vyplnenyFormular.getPasswordAgain());

    if(validationErrors.hasErrors()) {
        ModelAndView dataHolder = new ModelAndView("login/registration");
        dataHolder.addObject("chybnaRegistrace", "Požadované údaje nejsou zadány správně");
        dataHolder.addObject("registrace", vyplnenyFormular);
        return dataHolder;
    }

    if (!(uzivatel.getPassword().equals(uzivatel.getPasswordAgain())) ) {
        ModelAndView dataHolder = new ModelAndView("login/registration");
        dataHolder.addObject("registrace", uzivatel);
        dataHolder.addObject("chyba", "Opakované heslo není stejné");
        return dataHolder;
    }

    if (checkRegistration(vyplnenyFormular)) {
        ModelAndView dataHolder = new ModelAndView("login/registration");
        dataHolder.addObject("registrace", uzivatel);
        dataHolder.addObject("chyba", "Uživatel se stejnou e-mailovou adresou již je v
databázi");
        return dataHolder;
    }

    Uzivatel novyUzivatel = new Uzivatel();
    novyUzivatel.setMail(uzivatel.getMail());
    novyUzivatel.setPassword(uzivatel.getPassword());
    dataUzivatel.save(novyUzivatel);
    return new ModelAndView("redirect:/registrace-uspesna.html");
}
```

Dalším příkladem, kde je ukázán způsob předávání parametru při volání dané adresy, je metoda `zobrazStatistiku()`. Tato metoda převezme vstupní parametr (uložený v šabloně v proměnné `sezna.id`), který je předán voláním ze šablony `zobrazeni.html`, a to následující částí:

```
<td><a class="odkaz" th:href="{data/zobrazeni/' + sezna.id + '.html'}"
    href="detail.html">Vybrat</a></td>
```

Tento parametr, který představuje identifikační číslo čidla, si uloží do proměnné `id`. Metoda vrací odpovídající seznam vrtů (v proměnné „`vrt`“), a seznam všech měření, který patří

danému čidlu (v proměnné „zaznamy“), a dále vyhodnocené statistické údaje v proměnné „statistika“.

```
@RequestMapping("/data/zobrazeni/{cislo}.html")
public ModelAndView zobrazStatistiku(@PathVariable("cislo") Long id) {
    ModelAndView dataHolder;
    dataHolder = new ModelAndView("data/statistika");

    List<Zaznam> seznamZaznamu = dataZaznam.findListDleCidla(id) ;
    List<Vrt> seznamVrtu = dataVrt.findAllDleCidla(id);
    Statistika statistika = new Statistika();
    Long idHladinomeru = dataHladinomer.findOneDleCidla(id).getId();
    Long rozsahHladinomeru = dataHladinomer.findOne(idHladinomeru).getRozsah();
    List<Zaznam> prepoctenySeznamZaznamu = statistika.prepocetHodnot(seznamZaznamu, rozsahHladinomeru);
    StatistickeUdaje statistickeUdaje = statistika.statistika(prepoctenySeznamZaznamu);

    dataHolder.addObject("vrty", seznamVrtu);
    dataHolder.addObject("zaznamy", prepoctenySeznamZaznamu);
    dataHolder.addObject("statistika", statistickeUdaje);
    return dataHolder;
}
```

4.4.1.2 Balíček entity

Tento balíček obsahuje třídy, které definují vlastnosti jednotlivých objektů. Vesměs se jedná o entity odpovídající databázovým tabulkám. Tyto třídy obsahují pouze definici proměnných, vygenerované getry a setry, konstruktory a metody toString(), hash() a equals. Slouží především k tomu, aby mohly předávat nebo nastavovat hodnoty svých proměnných pomocí metod get a set. Přehled všech vytvořených entit podává následující seznam.

- Uzivatel
- UdajeUzivatele
- Cidlo
- Hladinomer
- Vrt
- Zaznam
- Opraveni
- Editace
- StatistickeUdaje

4.4.1.3 Balíček forms

Balíček forms obsahuje formuláře, pomocí kterých jsou uživatelem zadávány údaje - například při přihlášení, registraci, editaci nebo zadávání parametrů a nových objektů. Jsou zde třídy LoginForm, RegistrationForm, HladinomerForm a VrtForm.

Dále je uveden zdrojový kód pro třídu RegistrationForm, kde jsou použity již dříve zmíněné Springové validační anotace.

```
package cz.sedlackova.webapp.forms;

import javax.validation.constraints.*;

public class RegistrationForm {

    @NotNull
    @Pattern(regexp = "[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,4}", message = "Zadejte platnou e-mailovou adresu")
    private String mail;
    @NotNull
    @Size(min=6, message = "Zadejte heslo (minimálně 6 znaků)")
    private String password;
    @NotNull
    @Size(min=6, message = "Potvrzte prosím zadané heslo (minimálně 6 znaků)")
    private String passwordAgain;

    public RegistrationForm() {
    }

    public RegistrationForm(String mail, String password, String passwordAgain) {
        this.mail = mail;
        this.password = password;
        this.passwordAgain = passwordAgain;
    }

    public String getMail() {
        return mail;
    }

    public void setMail(String newValue) {
        mail = newValue;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String newValue) {
        password = newValue;
    }

    public String getPasswordAgain() {
        return passwordAgain;
    }

    public void setPasswordAgain(String newValue) {
        passwordAgain = newValue;
    }
}
```

Kromě zmíněných validačních anotací třída obsahuje pouze proměnné, konstruktory a metody a settery.

4.4.1.4 *Balíček logika*

V balíčku logika jsou dvě třídy, a to Encryption a Statistika.

Třída Encryption je určena pro generování „hashe“ hesla, které zadají uživatelé při registraci. Ve třídě jsou 2 metody pro 2 různé hashovací algoritmy, a to MD5 a SHA1. V programu je použit algoritmus MD5. Otisk hesla (tzv. hash) se ukládá do databáze kvůli

tomu, aby hesla nebyla uložena v čitelné podobě. Z otisku nelze zpětně zrekonstruovat původní podobu hesla.

Ve třídě Statistika probíhá výpočet všech statistických funkcí, převod hodnoty naměřené hladinoměrem v mA na úroveň výšky vodního sloupce nad sondou (viz vzorec v kapitole 3.1.6), různé převody časových formátů, zaokrouhlování

```
package cz.sedlackova.webapp.Logika;

import java.time.*;
import java.time.format.*;
import java.time.temporal.*;
import java.util.*;
import cz.sedlackova.webapp.entity.*;

public class Statistika {

    public StatistickeUdaje statistika(List<Zaznam> zaznamy) {

        LocalDateTime cas, min, max;
        Duration delkaZaznamu;
        String minim, maxim, delkaZaznamuString;
        float soucet, prumer, maximum, minimum, odchylka;
        soucet = prumer = maximum = minimum = odchylka = 0;
        min = LocalDateTime.MAX;
        max = LocalDateTime.MIN;

        for (Zaznam jedenzaznam : zaznamy) {
            if (jedenzaznam.getCas().isAfter(max)) max=jedenzaznam.getCas();
            if (jedenzaznam.getCas().isBefore(min)) min=jedenzaznam.getCas();
            soucet = soucet + jedenzaznam.getHodnota();
            if (jedenzaznam.getHodnota()>maximum) maximum = jedenzaznam.getHodnota();
            if (jedenzaznam.getHodnota()<minimum) minimum = jedenzaznam.getHodnota();
        }

        min = formatovaniCasu(min);
        maxim = formatovaniCasu(max);

        delkaZaznamu = Duration.between(min, max);
        delkaZaznamuString = formatDuration(delkaZaznamu);

        for (Zaznam jedenzaznam : zaznamy) {
            odchylka = odchylka + (jedenzaznam.getHodnota()-prumer);
        }
        if(zaznamy.size()!=0) {
            odchylka= (float) Math.sqrt( (odchylka / (float) zaznamy.size()));
        } else odchylka = 0;

        prumer = zaokrouhleniDveMista(soucet/ (float) zaznamy.size());
        odchylka = zaokrouhleniDveMista(odchylka);

        StatistickeUdaje statistickeUdaje = new StatistickeUdaje();
        statistickeUdaje.setDelka(delkaZaznamu);
        statistickeUdaje.setDelkaStr(delkaZaznamuString);
        statistickeUdaje.setMinCasStr(minim);
        statistickeUdaje.setMaxCasStr(maxim);
        statistickeUdaje.setMinCas(min);
        statistickeUdaje.setMaximum(maximum);
        statistickeUdaje.setMinimum(minimum);
        statistickeUdaje.setPrumer(prumer);
        statistickeUdaje.setSmerodatnaOdchylka(odchylka);

        return statistickeUdaje;
    }

    public List<Zaznam> prepocetHodnot(List<Zaznam> zaznamy, Long rozsah) {

        for (Zaznam jedenzaznam : zaznamy) {
```

```

        float novaHodnota = zaokrouhleniDveMista((jedenzaznam.getHodnota()/100-
4)*rozsah/16);
        jedenzaznam.setHodnota(novaHodnota);
        jedenzaznam.setCasStr(formatovaniCasu(jedenzaznam.getCas()));
    }
    return zaznamy;
}

public float zaokrouhleniDveMista(float hodnota) {

    int nTemp = (int)((hodnota + 0.005) * 100.0);
    hodnota = ((float)nTemp)/100.0F;
    return hodnota;
}

public String formatovaniCasu(LocalDateTime cas) {

    String novyCas =
cas.format(DateTimeFormatter.ofLocalizedDateTime(FormatStyle.SHORT, FormatStyle.SHORT));
    return novyCas;
}

public static String formatDuration(Duration duration) {
    long seconds = duration.getSeconds();
    long absSeconds = Math.abs(seconds);
    String positive = String.format(
        "%d:%02d:%02d",
        absSeconds / 3600,
        (absSeconds % 3600) / 60,
        absSeconds % 60);
    return seconds < 0 ? "-" + positive : positive;
}
}

```

4.4.1.5 Balíček repository

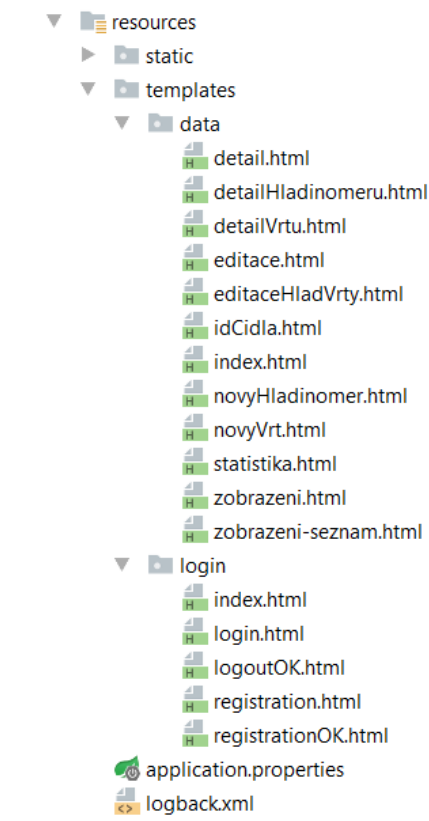
V tomto balíčku jsou zařazeny všechny třídy, ve kterých jsou definovány metody a proměnné nutné pro práci s databází. Jedná se o následující třídy.

- UzivatelRepository
- CidloRepository
- HladinoerRepository
- VrtRepository
- Zaznam.repository

Struktura a funkce Repository již byla popsána u aplikace pro ukládání dat na server, včetně uvedení kódu. Proto zde již další podrobnosti nejsou uvedeny.

4.4.2 Šablony

Šablony, neboli Templates, jsou HTML stránky, které mají v sobě vloženy speciální značky šablonovacího systému, pomocí kterých jsou předávána data vygenerovaná javou. V aplikaci jsou uloženy v adresáři resources – templates, na rozdíl od statických webových stránek, které jsou uloženy v resources – static. Přehled vytvořených šablon je na obrázku (Obr. 16).



Obr. 16. Šablony

4.4.2.1 Složka login

Ve složce login jsou uloženy všechny šablony, které se týkají přihlášení uživatele do aplikace. Jako příklad je zde uvedena kód částí souboru registration.html, která zahrnuje také validační značky s chybovým hlášením. Odpovídající část jazykové třídy MainController je prezentován v kapitole 4.4.1.1.

```

<div class="polozky">
  <div class="chyba" th:text="${chybnaRegistrace}"></div>
  <form action="#" th:action="@{/registrace.html}" th:object="${registrace}" method="post">
    <table class="tabulka formular">
      <tr>
        <td>
          <label for="em">Email:</label>
        </td>
        <td>
          <input id="em" name="mail"
            th:field="**{mail}"/>
        </td>
        <td class="validace" th:if="${#fields.hasErrors('mail')}"
          th:errors="**{mail}"></td>
      </tr>
      <tr>
        <td>
          <label for="hes">Heslo:</label>
        </td>
        <td>
          <input id="hes" name="password" type="password"

```

```

                th:field="*{password}"/>                </td>
                <td class="validace" th:if="{#fields.hasErrors('password')}"
th:errors="*{password}"></td>
            </tr>
            <tr>
                <td>
                <label for="hes2">Heslo (znovu):</label>
                </td>
                <td>
                <input id="hes2" name="passwordAgain" type="password"
                th:field="*{passwordAgain}"/>
                <td class="validace" th:if="{#fields.hasErrors('passwordAgain')}"
th:errors="*{passwordAgain}"></td>
            </tr>
            <tr>
                <td>&nbsp;</td>
                <td class="validace" th:text="{chyba}"></td>
            </tr>
            <tr>
                <td>&nbsp;</td>
                <td>
                <input class="odkaz" type="submit" value="Zaregistrovat se"/>
                </td>
            </tr>
        </table>
    </form>
</div>

```

Data vygenerovaná na straně serveru se do tagů v šabloně vkládají pomocí Thymeleafové značky **th:**.

4.4.2.2 Složka data

Ve složce data jsou uloženy šablony, které se zabývají zpracováním a ukládáním dat, tedy té části aplikace, která se objeví po přihlášení uživatele. Jako příklad je uvedena část souboru statistika.html, která je zodpovědná za vypsání tabulky se seznamem vrtů, příslušejících danému čidlu (odpovídající kód v javové třídě MainController je prezentován v kapitole 4.4.1.1).

```

<table class="tabulka">
    <tr>
        <td>Název</td>
        <td>Hloubka</td>
        <td>Popis</td>
        <td>Hladina od OB</td>
        <td>Id hladinoměru</td>
        <td></td>
    </tr>
    <tr th:each="seznam : {vrty}">
        <td th:text="{seznam.nazev}">HV-1</td>
        <td th:text="{seznam.hloubka}">20</td>
        <td th:text="{seznam.popis}">Vrt Otrokovice</td>
        <td th:text="{seznam.hladina_ob}">5.6</td>
        <td th:text="{seznam.id_hladinomer}">3</td>
    </tr>

```

Ostatní .html soubory mají podobnou strukturu, kompletní výpis celého kódu je uveden v příloze.

5 HARDWAROVÁ REALIZACE A TESTOVÁNÍ SYSTÉMU

Vzhledem k dlouhé době dodání hladinoměru proběhlo v první fázi testování příjmu dat z čidla společnosti ELKO EP, s.r.o., které bylo napojeno na teplotní senzor. Cílem tohoto testování bylo zachytit posílaný paket, ověřit si jeho strukturu a určit postup pro získání potřebných dat. Toto testování probíhalo od začátku května. Aplikace byla nasazena na adrese <http://hydrosonda.tomcat.cloud/collector/>, vlastní data byla posílána na adresu [http://hydrosonda.tomcat.cloud/collector/ feed](http://hydrosonda.tomcat.cloud/collector/feed). Pro účely přijímání dat byla vytvořena jediná databázová tabulka o 3 polích – body, token a timestamp. Toto zkušební testování přispělo k rychlejšímu vývoji samotné aplikace.

Pro vlastní testování byl vybrán hydrogeologický vrt v Holešově, ul. Boženy Němcové, s hloubkou vrtu cca 29 m a hladinou pohybující se kolem 18 – 20 m od odměrného bodu, kterým je horní okraj betonové skruže. Ve vrtu již bylo instalováno čerpadlo, které je umístěno v kalníku, v hloubce cca 27 m pod terénem.

Vlastní testování bylo zahájeno 18.5.2017. Zvolené hardwarové komponenty jsou popsány v kapitole 3.1. Hladinoměr byl instalován do hloubky cca 24 m od horního okraje skruže, přičemž délka kabelu hladinoměru je 25 m. Jelikož k jeho napájení nebylo možno využít čidla (to je napájeno pouze baterií 3,6 V), bylo použito nejjednodušší řešení a hladinoměr byl připojen ke zdroji elektrické energie u vrtu. Pokud by v místě nebyl přístupný zdroj elektrické energie, bylo by alternativním řešením použití solárního panelu.

Ještě před instalací bylo zaměstnancem firmy ELKO EP, s.r.o. v místě vrtu provedeno měření velikosti signálu sítě LoRa Českých radiokomunikací, a vzhledem k jeho malé síle (stínění zástavbou rodinných domů) bylo rozhodnuto o instalaci vlastní gateway v blízkém okolí vrtu.

Stav vrtu se zapuštěným hladinoměrem a čidlem instalovaným na vnější straně betonové skruže dokumentuje obrázek (Obr. 17).



Obr. 17. Fotografie vrtu s instalovaným hladinoměrem a čidlem

Před zahájením testování byla na stejnou url adresu jako v předchozím případě, tedy <http://hydrosonda.tomcat.cloud/collector/>, nasazena upravená aplikace, která již byla funkční a prakticky odpovídá aplikaci v současném stavu. V databázi na tomto serveru jsou tedy od 18.5.2018, od cca 9:00 hodin ráno ukládány záznamy s velikostí výstupního proudu, časem měření a EUI. Interval měření byl nastaven na 10 minut.

Poté byla na adresu <http://hydrosonda.tomcat.cloud/presenter> nasazena druhá aplikace na zobrazování a vyhodnocování dat. Tato aplikace je na této adrese v současné době funkční, a je tak možné sledovat načítání dat z hladinoměru.

Z hardwarové realizace a testování vyplynulo několik skutečností, které jsou shrnuty v dalším textu.

- Bylo ověřeno, že interval měření může být nastaven na 1 minutu. Tento interval byl zkušebně nastaven ještě před instalací hladinoměru, při testování čidla, a jak je možno ověřit v aplikaci, v době mezi 8:55 – 9:24 posílal data (i když ne relevantní) v intervalu 1 minuty.
- Ukázalo se, že není jistota doručení každého datového paketu. Ze 6 zpráv v hodině jich bylo většinou doručeno 4 – 5, někdy 6, výjimečně jenom 3. Důvodem je prav-

děpodobně skutečnost, že trasa signálu ze zařízení na gateway (GW) je na dané lokalitě v zákrytu několika budov, takže dochází k přenosu spíše odrazy. Technologie LoRa je náchylná na překážky v cestě, pro větší efektivitu je nutná přímá viditelnost zařízení na GW.

- Interval měření nebyl přesně 10 minut, ale o něco méně. Při konzultaci s pracovníkem ELKO EP, s.r.o. mi bylo sděleno, že nelze zajistit přesný čas (v rámci sekund) z několika důvodů. Hlavní odchylka je dána principem fungování čidla. Jelikož pracuje na baterie, tak se velkou část času udržuje v power down modu. Z tohoto režimu se probouzí jednou za 1 s a provádí nutné operace. Tento čas je počítán jako 1 s, i když to přesně 1 s není. Další odchylka vzniká v modulu RTC (Real Time Clock), jehož krystal je závislý na teplotě. Další odchylka může vzniknout na straně gateway.

Z výše uvedených skutečností lze vyvodit následující závěry:

- Použitá technologie má omezení dané pokrytím území signálem. V některých místech bude možno měřit bez problémů, někde to naopak nebude fungovat. Zajímavé by bylo vyzkoušet měření mimo obydlená místa. Z tohoto hlediska by patrně bylo praktičtější použít síť NarrowBand IoT, která má výhodu využití stávající sítě LTE mobilních operátorů.
- Výhodou je možnost nastavení intervalu měření na 1 minutu, což třeba neumožňuje síť SigFox. Nemožnost nastavení přesného časového intervalu nebude pro měření hladiny podzemní vody pravděpodobně znamenat žádné omezení.
- Velkou výhodou oproti stávajícím systémům je, že není nutno používat SIM karty.
- Omezením je nutnost připojení elektrického proudu, nicméně, což by však jistě šlo vyřešit nějakým jiným způsobem (např. již zmíněnými solárními panely).

6 DOKUMENTACE APLIKACE

Aplikace Hydrosonda je webovou aplikací, sloužící k přijímání dat z hladinoměru, instalovaného v hydrogeologickém vrtu. Skládá se ze dvou částí – aplikace, která data přijímá a ukládá je do databáze na server (url adresa Aplikace/Collector) a aplikace, která umožňuje editovat související údaje a zobrazovat přijímaná data (url adresa Aplikace/Presenter).

6.1 Specifikace zařízení

Aplikace je vytvořena specificky pro následující zařízení:

Hladinoměr: HLM-25S firmy Dinel s proudovým výstupem 4 – 20 mA

Čidlo: AirIM-100 firmy ELKO EP, s.r.o.

Přenosová technologie: LoRa

Ukázka přijímaného formátu dat (JSON):

```
{„data“:“{“EUI“:“020102fffe0e0000“,“fcent“:868100000,“62on“:false,“data“:“0a081F66“,“cmd“:“gw“,“ts“:1525421848,“gws“:[{“lon“:17.5783297,“ts“:1525421848,“lat“:49.3333095,“snr“:7,“gweui“:“F03D29FFFF000031“,“rssi“:107}],“freq“:868100000,“toa“:46,“dr“:“SF7BW125 4\\V5“,“port“:1}“,“type“:“D“,“tech“:“L“}
```

Aplikaci lze využít i na jiný druh hladinoměru nebo čidla s následujícím omezením:

Hladinoměr - je nutný proudový výstup 4 – 20 mA.

Čidlo - je podstatný odpovídající formát dat, přičemž podstatné jsou identifikátory „EUI“ a „ts“ (časový otisk). Hodnota výstupního proudu musí být uložena jako hexadecimální číslo ve druhých 2 bytech hodnoty za identifikátorem „data“ (v uvedeném případě je to 1F66). Mezi daným identifikátorem a jeho hodnotou musí být odpovídající počet oddělovacích znaků.

6.2 Postup při využívání aplikace

- Na webové stránce, kde je aplikace instalována, provede uživatel registraci. Jako přihlašovací jméno je nutno zadat emailovou adresu, heslo musí být alespoň 6 místné.
- Administrátor databáze (aplikace) přiřadí uživateli do databáze EUI (64 bitový jednoznačný identifikátor zařízení sítě LoRa), a také hladinoměr (název hladinoměru a

maximální výšku vodního sloupce). Oba údaje zadané k hladinoměru může uživatel později v aplikaci změnit. Číslo EUI může změnit nebo přidat pouze administrátor.

- Teprve po přiřazení čísla EUI uživateli je možno zahájit posílání dat na server (tedy zahájit měření). **DATA, KTERÁ BUDOU POSLÁNA NA SERVER PŘED PŘÍRAZENÍM ČÍSLA EUI UŽIVATELI NEBUDOU DO DATABÁZE ULOŽENA!!!**
- Po zaregistrování EUI do databáze se uživatel přihlásí do aplikace a přes položku menu *Editace údajů* se mu zobrazí seznam čidel, která má k sobě přiřazená. Po vybrání daného čidla se mu dále zobrazí přiřazený hladinoměr, jehož údaje může editovat. Dále může danému hladinoměru přidat vrt, případně i více vrtů. Vždy platí, že 1 čidlu je přiřazen právě 1 hladinoměr, ale tato sestava čidlo-hladinoměr může mít přiřazena větší množství vrtů, pokud je použita opakovaně na jiném vrtu.
- S využitím položky menu *Zobrazení dat* je možno si uložená data prohlédnout (opět je potřeba vybrat čidlo). Zobrazeny jsou tyto statistické údaje měření: začátek měření, konec měření (pokud měření stále probíhá, zobrazuje se zde čas poslední změřené hodnoty), celková doba měření, maximální, minimální a průměrná naměřená hodnota a směrodatná odchylka měření. Ukládání dat probíhá v reálném čase, takže při každém obnovení prohlížeče se načítají nová data, která případně byla uložena do databáze (dle nastaveného intervalu měření), která přicházejí na server, a statistické údaje se přepočítávají. Dále je zobrazen seznam přiřazených vrtů a seznam jednotlivých naměřených hodnot (doba měření + naměřená hodnota).
- Pro odhlášení uživatele slouží položka menu *Odhlášení*.

ZÁVĚR

Výsledkem této práce je navržení funkčního systému pro realizaci vzdáleného monitoringu úrovně hladiny podzemní vody v hydrogeologickém vrtu, zahrnující jak návrh hardwarové realizace, tak vývoj aplikace v jazyce Java, která přijímá a dále zpracovává měřená data.

Nejdříve bylo nutno shromáždit informace o dostupných technologiích pro dálkový přenos dat, využívaných pro IoT, a provést výběr této technologie. Tento výběr byl ovlivněn výsledky analýzy využití naměřených dat hydrogeology, a to především z hlediska frekvence měření. Dalším krokem byl výběr vhodného zařízení, schopného realizovat přenos dat. Následoval výběr hladinoměru, který je možno použít v hydrogeologickém vrtu, a který zároveň bude komunikovat s vybraným zařízením pro přenos dat.

Jakmile byla známa technologie přenosu a odpovídající formát přenášených dat, bylo možno vytvořit aplikaci sloužící k jejich zachytávání a ukládání do databáze. V první fázi byl ukládán celý obsah paketu přicházejícího ve formátu JSON jako jeden řetězec, což umožnilo provést analýzu vstupních dat. Následně byla aplikace upravena tak, aby ukládala do databáze jednotlivá příchozí data, nezbytná pro další využití.

Souběžně s výše uvedenými kroky byl proveden návrh databázové struktury, kterou bude možno využít pro přijímání, zobrazování a vyhodnocování dat. Jelikož se jedná o webovou aplikaci, bylo nutno vyřešit také bezpečnost, danou nutností přihlašování jednotlivých uživatelů. S tím souvisel také postup při využívání aplikace – některé úkony může provést samotný uživatel, některé, např. přiřazení čidla uživateli, musí zadat administrátor (přesný postup těchto kroků je uveden v dokumentaci aplikace v kapitole č. 6).

Poté bylo navrženo uživatelské prostředí aplikace pro vyhodnocování dat. Následným krokem bylo vytvoření této aplikace. Její uživatelské prostředí je popsáno v kapitole 3.2, implementace obou aplikací v kapitole č. 4. Vzhledem k velkému množství obsažených souborů (jak javových tříd, tak šablon) jsou v textu popsány pouze některé z nich. Kompletní kód je součástí příloh.

Ve finální fázi byl celý navržený systém realizován a otestován - zvolené komponenty byly instalovány do reálného vrtu. Test proběhl úspěšně, naměřená data se přenášela do aplikace pro přijímání dat, která je ukládala do vytvořené databáze. Po přihlášení do aplikace pro vyhodnocování dat bylo možno tato data sledovat, včetně statistického vyhodnocení.

SEZNAM POUŽITÉ LITERATURY

- [1] Co je IoT? – IoT portál. *IoT portál – Brána do světa internetu věcí* [online]. Copyright © 2018 [cit. 12.05.2018]. Dostupné z: <https://www.iot-portal.cz/co-je-iot/>
- [2] Elektrotechnika – TZB-info. *Elektrotechnika – TZB-info* [online]. Copyright © Fotolia.com [cit. 13.05.2018]. Dostupné z: <https://elektro.tzb-info.cz/informacni-a-telekomunikacni-technologie/16519-site-pro-internet-veci-v-ceske-republice>
- [3] Internet věcí – Pavel Pohanka. *Úvodní stránka – Pavel Pohanka* [online]. Copyright © PP 2017 [cit. 12.05.2018]. Dostupné z: <http://i2ot.eu/internet-of-things/>
- [4] Co se skrývá pod výrazy Industry 4.0 / Průmysl 4.0 ? | Automatizace.HW.cz. *Automatizace.HW.cz | Elektronika v automatizaci* [online]. Copyright © 1997 [cit. 22.05.2018]. Dostupné z: <https://automatizace.hw.cz/mimochodem/co-je-se-skriva-pod-vyrazy-industry-40-prumysl-40.html>
- [5] Internet of Things (2.část) – Technologie pro bezdrátový přenos dat | Vývoj.HW.cz. *Vývoj.HW.cz | Vše o elektronice a programování* [online]. Copyright © 1997 [cit. 12.05.2018]. Dostupné z: <https://vyvoj.hw.cz/internet-of-things-2cast-technologie-pro-bezdratovy-prenos-dat.html>
- [6] Technologie – IoT portál. *IoT portál – Brána do světa internetu věcí* [online]. Copyright © 2018 [cit. 12.05.2018]. Dostupné z: <https://www.iot-portal.cz/technologie/>
- [7] Připravujeme. *Měření Energie SDS moduly* [online]. Copyright © 2015 Všechna práva vyhrazena pro [cit. 24.05.2018]. Dostupné z: <http://www.merenienergie.cz/pripravujeme.html>
- [8] BUJOK, Petr a Arnošt GRMELA. *Hydrodynamické zkoušky a výzkum sond*. Ostrava: Vysoká škola báňská v Ostravě, 1993. ISBN 80-7078-144-0.
- [9] Učební materiály /um/. *Veřejné služby Informačního systému* [online]. [cit. 13.05.2018]. Dostupné z: <https://is.muni.cz/el/1431/podzim2005/G7401/um/>
- [10] JSP.cz – měření a regulace ... Měření hladiny. [online]. [cit. 13.05.2018]. Dostupné z: http://www.jsp.cz/cz/sortiment/seznam_dle_kategorii/snimace_hladiny/teorie-hladina/

- [11] ALA. *ALA* [online]. Copyright © [cit. 13.05.2018]. Dostupné z: <http://www.ala1.com/>
- [12] Automatické hladinoměry | Ekotechnika. *Přístroje pro diagnostiku životního prostředí* | *Ekotechnika* [online]. Copyright © Ekotechnika spol. s r.o. [cit. 14.05.2018]. Dostupné z: <http://www.ekotechnika.cz/automaticke-hladinomy>
- [13] H40 – GSM/GPRS datalogger, hladinoměr | FIEDLER. *Homepage | FIEDLER | Electronics for ecology* [online]. [cit. 14.05.2018]. Dostupné z: <https://www.fiedler.company/cs/produkty/male-telemetricke-stanice-dataloggery/sestava-h40-pro-mereni-hladiny-teploty>
- [14] Univerzální modul AirIM-100L • ElkoEP. *Hlavní strana • ElkoEP* [online]. Copyright © 2016 ELKO EP [cit. 14.05.2018]. Dostupné z: <https://www.elkoep.cz/univerzalni-modul-airim-100l>
- [15] Ponorný hydrostatický hladinoměr HLM-25S – Dinel, s.r.o.. *Dinel, s.r.o. – Měření hladiny a průtoku* [online]. [cit. 14.05.2018]. Dostupné z: <http://www.dinel.cz/vyrobky/kontinualni-hladinomy/ponorny-hydrostaticky-hladinomer-hlm-25s>
- [16] Technické aspekty technologie LoRa – #PRIPOJME. *#PRIPOJME – České Radiokomunikace a.s.* [online]. Copyright © 2017 ČESKÉ RADIOKOMUNIKACE A.S. [cit. 14.05.2018]. Dostupné z: <https://pripoj.me/technicke-aspekty-technologie-lora/>
- [17] Java EE – Documentation | Oracle Technology Network | Oracle . [online]. Copyright © Oracle [cit. 15.05.2018]. Dostupné z: <http://www.oracle.com/technetwork/java/javaee/documentation/index.html>
- [18] Java Enterprise Edition (JEE). [online]. Copyright © 2018 itnetwork.cz. [cit. 15.05.2018]. Dostupné z: <https://www.itnetwork.cz/java/jee>
- [19] Spring Documentation. *Spring* [online]. Copyright © [cit. 15.05.2018]. Dostupné z: <https://spring.io/docs>
- [20] Model-view-controller – FI WIKI. [online]. [cit. 15.05.2018]. Dostupné z: <https://kore.fi.muni.cz/wiki/index.php?title=MVC>
- [21] Apache Tomcat® - Welcome!. *Apache Tomcat® - Welcome!* [online]. Copyright © 1999 [cit. 15.05.2018]. Dostupné z: <http://tomcat.apache.org/>

- [22] Maven – Welcome to Apache Maven. *Maven – Welcome to Apache Maven* [online]. Copyright ©2002 [cit. 16.05.2018]. Dostupné z: <https://maven.apache.org/>
- [23] Thymeleaf. *Thymeleaf* [online]. Copyright © The Thymeleaf Team [cit. 16.05.2018]. Dostupné z: <https://www.thymeleaf.org/index.html>
- [24] MariaDB.org – Supporting continuity and open collaboration. *MariaDB.org – Supporting continuity and open collaboration* [online]. [cit. 16.05.2018]. Dostupné z: <https://mariadb.org/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AES	Advanced Encryption Standard – standard pokročilého šifrování
BTS	Base Transceiver Station – vysílač a přijímač radiových signálů
ČHMÚ	Český hydrometeorologický ústav
EUI	64 bitový jednoznačný identifikátor zařízení sítě LoRa
GW	Gateway
HDZ	Hydrodynamické zkoušky
HSDPA	High Speed Download PACKET Access, protokol mobilních telefonů
IDE	Integrated Development Environment – vývojové prostředí
IoT	Internet of Things – internet věcí
J2SE	Java Standard Edition
J2EE	Java Enterprise Edition
LPWAN	Low Power Wide Area Network – nízkoenergetická síť pro rozsáhlé území
OB	Odměrný bod
RFID	Radio Frequency Identification – identifikace na radiové frekvenci, identifikátory navržené identifikaci zboží, navazující na systém čárových kódů
RTC	Real Time Clock – hodiny reálného času, počítačové hodiny
M2M	Machine-to-machine – komunikace mezi stroji
OB	Odměrný bod
PAN	Personal Area Network – osobní síť
RF	Rádiová frekvence
WAN	World Area Network
WAP	Wireless Application Protocol

SEZNAM OBRÁZKŮ

Obr. 1.	Schema přenosu dat v technologii LoRa [7]	17
Obr. 2.	Schéma hydrogeologického vrtu s hladinoměrem	22
Obr. 3.	Čidlo AirIM-100 společnosti ELKO EP, s.r.o. [14].....	29
Obr. 4.	Hladinoměr HLM-25S [15].....	30
Obr. 5.	Schéma databázových tabulek	33
Obr. 6.	Úvodní webová stránka programu Collector	36
Obr. 7.	Přihlašovací formulář	37
Obr. 8.	Registrační formulář.....	38
Obr. 9.	Výběr čidla	39
Obr. 10.	Editace.....	39
Obr. 11.	Formulář editace vrtu	40
Obr. 12.	Zobrazení statistických údajů.....	41
Obr. 13.	Zobrazení naměřených dat	41
Obr. 14.	Struktura balíčků a adresářů.....	49
Obr. 15.	Struktura uspořádání javových tříd	50
Obr. 16.	Šablony.....	57
Obr. 17.	Fotografie vrtu s instalovaným hladinoměrem a čidlem.....	60

SEZNAM TABULEK

Tab. 1.	Význam parametrů v těle příkazu POST, formát json	31
Tab. 2.	Namapované webové stránky (Controller) a odpovídající šablony	51

SEZNAM PŘÍLOH

P I CD

PŘÍLOHA P I: CD

Obsah přiloženého CD:

- Bakalářská práce v elektronické podobě
- Katalogový list čidla AirIM-100
- Technická specifikace hladinoměru HLM-25S
- Zdrojový kód programů obou aplikací
- Spustitelné aplikace (war)
- příkazy pro založení databáze