

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informačních technologií**

**Implementace SMART technologií do počítačové sítě obce**  
Bakalářská práce

Autor: Václav Tomíček  
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Pavel Blažek

Hradec Králové

srpen 2019

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 13.8.2019

.....

Václav Tomíček

#### Poděkování:

Velké poděkování patří vedoucímu bakalářské práce Ing. Pavlu Blažkovi za metodické vedení práce, ochotu a vytrvalost. Dále bych chtěl poděkovat za pomoc při mechanické úpravě zařízení Bc. Aleši Jeriovi.

## **Anotace**

Bakalářská práce se zabývá problematikou napojení Smart technologií do stávající datové sítě obce a jejich zabezpečení. Práce je věnována seznámení s termínem Smart Cities a dílčích částí, které do tohoto konceptu zapadají. Jsou zde rozebrány základní definice a oblasti využití. Dále se práce věnuje síťové struktuře chytrého města a jejího fungování s kombinací s databázovými systémy. V další kapitole je rozebráno, jak by mohlo chytré město fungovat a jaký je aktuální stav implementace konceptu Smart city.

V rámci praktického řešení je popsáno využití služeb ve Firebase, úprava IoT zařízení a jeho napojení do Wi-Fi sítě. Dále je pak představeno řešení komunikace takového zařízení s databází. V reakci na to byl vytvořen jednoduchý informační systém pomocí webové a mobilní aplikace. Webová aplikace slouží k zobrazování aktuálních dat a upozornění, tato upozornění je možné spravovat. Mobilní aplikace pracuje na mobilním systému Android a obdobně jako webová aplikace zobrazuje aktuální data a upozornění.

## **Annotation**

### **Title: Implementation of SMART technologies into the city's computer network**

The bachelor thesis deals with a problematics of connection of Smart technologies to existing community data network and its security. The thesis deals with problematics of Smart Cities and other technologies linked to it. Basic definitions and areas of use of these technologies are discussed. The thesis deals with network structure of a smart city and its combination with database systems. The next chapter discusses how a smart city could work and what is the current state of implementation of the Smart city concept.

The practical part of the thesis describes use of services in Firebase, IoT device modification and its connection to Wi-Fi network. Furthermore, the thesis deals with problematics of communication of the device with database. A simple information system was created using a web and mobile application. The web application shows current data and notifications, these notifications can be managed using the system. The mobile application works on the Android mobile system, and similar web application, it displays current data and notifications.

## Obsah

1	Úvod.....	0
2	Cíl práce.....	1
3	Metodika zpracování.....	2
4	Chytrá města (Smart cities).....	3
4.1	Chytrá ekonomika (Smart economy).....	3
4.2	Chytrá doprava (Smart mobility).....	3
4.3	Chytré životní prostředí (Smart environment) .....	4
4.4	Chytrí lidé (Smart people) .....	4
4.5	Chytré bydlení (Smart living) .....	4
4.6	Chytrá vláda (Smart governance) .....	4
5	Síťová struktura chytrého města .....	5
5.1	Město.....	5
5.2	Ulice .....	5
5.3	Nástup IoT.....	6
5.4	Cloud Computing a Big Data.....	8
5.5	Správa internetových služeb.....	10
5.5.1	Zabezpečená překryvná síť (Secure Overlay Network) pro služby Chytrého města .....	11
6	Fungování chytrého města .....	12
7	Současný stav Smart Cities .....	14
8	Napojení Smart technologií do současné sítě.....	17
8.1	Použitá zařízení a služby .....	19
8.1.1	Mikrotik RB493G .....	19
8.1.2	Mikrotik miniPCI R52nM.....	19
8.1.3	ESP8266 DHT11 teploměr a vlhkoměr + Wi-Fi modul ESP-01 .....	19

8.1.4	Napájení monitorovací jednotky .....	20
8.1.5	Firestore .....	20
8.2	Příprava a programování .....	21
8.2.1	Připojení do sítě .....	21
8.2.2	Příprava databáze.....	22
8.2.3	Programování ESP-01 se senzorem ESP8266 DHT11 .....	22
8.2.4	Programování webové aplikace.....	26
8.2.5	Programování Android aplikace.....	30
8.3	Shrnutí výsledků.....	36
9	Závěry a doporučení .....	37
10	Seznam použité literatury .....	38

## Seznam obrázků

Obrázek 1: Vylepšené veřejné osvětlení vlevo, vpravo současné osvětlení. [25].....	14
Obrázek 2: Mapa pokrytí města Plzně přístupovými body sítě PLZEN LoRa [16] ...	16
Obrázek 3 Návrh řešené situace, zdroj: vlastní.....	18
Obrázek 4 Schéma zapojení, zdroj: vlastní .....	21
Obrázek 5 Oprávnění Firebase Realtime Database, zdroj: vlastní.....	22
Obrázek 6 ESP-01 pin diagram [26] .....	22
Obrázek 7 - Zabezpečení a odeslání dat do databáze, zdroj: vlastní.....	24
Obrázek 8 Propojení krajního pinu CPU a pinu RESET [26] .....	26
Obrázek 9 Uložení dat v databázi, zdroj: vlastní.....	27
Obrázek 10 Výsledná webová aplikace, zdroj: vlastní .....	30
Obrázek 11 Návrh aplikace a rozmístění komponent, zdroj: vlastní.....	32
Obrázek 12 Výsledná Android aplikace, zdroj: vlastní.....	35

## Seznam tabulek

Nenalezena položka seznamu obrázků.



# 1 Úvod

Do roku 2030 má být na zemi přes 8,6 miliardy obyvatel, z toho každý druhý bude žít ve městě. S rozšiřující se infrastrukturou města nastávají problémy s dopravou, zdravotnictvím, bezpečnostní situací a s důležitými zdroji jako je voda nebo elektřina. Proto je potřeba zajistit veškeré potřebné zdroje k životu, udržet další rozvoj měst s narůstajícím počtem obyvatel a co nejvíce zamezit plýtvání se zdroji. S tím také souvisí informovanost obyvatel a jejich zapojení do konceptu.

V dnešní době je to velmi aktuální a důležité téma, které se rozšiřuje napříč světem a má vysoký potenciál pro budoucnost. Také proto jsem si vybral téma bakalářské práce Implementace SMART technologií do počítačové sítě obce. Je v ní uplatněn technický pohled na dané téma. K úspěšné implementaci technologií je však třeba znát "trendy" a koncepty spojené s ní. Teprve poté je možné se zabírat senzory, jejich zapojením do datové sítě města, odesláním dat do databázového systému a prezentaci výsledných dat v aplikacích. Ty pak umožňují správcům měst zlepšovat životní podmínky a zjednodušovat každodenní činnosti obyvatel.

## **2 Cíl práce**

Cílem bakalářské práce je seznámit se s konceptem Smart city a základními prvky síťové infrastruktury. Práce by měla popsat, jak má Smart city fungovat a sdělit aktuální stav. Dalším cílem je návrh nasazení Smart technologií s využitím datové sítě města. Technologie by měli být porovnány, zjištěny jejich silné a slabé stránky, způsob získávání dat a jejich ukládání. Mělo by být nastíněno, jejich využití a náročnost implementace pro obyvatele města.

### **3 Metodika zpracování**

K dosažení cíle bylo nutné se seznámit se základními koncepty vytváření struktur Smart city. Na základě získaných a setříděných informací pak bylo možno hledat technické možnosti pro realizaci praktické části. V posledním kroku byl sestaven funkční prvek schopný integrace do uvažovaného systému.

## 4 Chytrá města (Smart cities)

Chytrému městu nelze přiřadit jasnou definici. Chytré město je novodobý projekt, který se snaží propojit všechny části města do jedné velké struktury. Ať už se jedná o ekonomiku města, o dopravu, životní prostředí, vzdělání či o bydlení.

Podle evropské studie Mapping Smart cities in the EU [1] lze rozdělit chytré město na 6 menších celků, toto vymezení nám pomůže pochopit základní strukturu chytrého města a jeho funkci. Důležité je také zmínit, že se všechny tyto struktury navzájem prolínají a nelze se zaměřit pouze na jednu z těchto struktur.

### 4.1 Chytrá ekonomika (Smart economy)

Prvním celkem je chytrá ekonomika. Tím rozumíme elektronický obchod, pokročilou výrobu, která zvyšuje produktivitu a optimalizuje náklady, inovace podporující nové produkty, služby a podnikání. Chytrá ekonomika zahrnuje také místní a globální propojení, kde se rozvíjí spolupráce mezi městy a státy, ale i kolaborativní ekonomika. Velmi důležitá součást k fungování chytré ekonomiky, je zájem občanů, kteří mají potřebné vzdělání a dokáží ovládat nové technologie.

### 4.2 Chytrá doprava (Smart mobility)

Druhým celkem je chytrá doprava. To znamená udržitelné, bezpečné a propojené dopravní systémy, které mohou zahrnovat tramvaje, autobusy, vlaky, metro, automobily a kola. Snaží se upřednostňovat nemotorové možnosti. Informace o dopravě poskytované v reálném čase a zlepšit efektivitu dojíždění, ušetřit náklady a snížit emise CO<sub>2</sub>. Uživatelé tohoto systému mohou poskytovat informace v reálném čase a přispět k dlouhodobému plánování. Krokem ke zlepšení dopravy je vylepšená komunikace mezi lidmi. Pokud 3 lidé jedou stejným směrem, není potřeba, aby každý jel svým vlastním autem. V Česku na tomto konceptu funguje aplikace známá pod jménem BlaBlaCar. Dopravě by mělo pomoci také chytré parkování. Jeho správnou implementací má potenciální řidič dopředu k dispozici informace o možnostech parkování ve zvolených oblastech.

### **4.3 Chytré životní prostředí (*Smart environment*)**

Třetím celkem je chytré životní prostředí. Zahrnujeme do toho energii z obnovitelných zdrojů, energetické sítě, kontrolu znečištění, zelené budovy a efektivitu využívání zdrojů. Dále také městské služby, jako je pouliční osvětlení, nakládání s odpady, kanalizační systém, vodní zdroje a další. Veškeré tyto služby jsou monitorovány a vyhodnocovány v reálném čase, díky čemuž můžeme docílit zvýšení kvality a efektivity služeb. Tím je možné dosáhnout menšího znečištění města a ovzduší, ale také zlepšit kvalitu vody nebo snížit energetickou náročnost.

### **4.4 Chytrí lidé (*Smart people*)**

Dalším celkem jsou tzv. chytrí lidé. Je zde kladen důraz na přístup ke vzdělání, získávání nových znalostí, zkušeností a jejich sdílení s ostatními. Dále nabízí lidem a komunitám pomáhat s rozvojem města. Tím že jim jsou poskytnuty data, se kterými mohou manipulovat a analyzovat je, vzniknou tak nové potřebné služby či produkty ve městě. V neposlední řadě je zde ještě využití sociálních sítí a mobilních či webových aplikací, které mohou usnadnit možnosti informovat obyvatele o počasí, důležitých změnách nebo bezpečnostní situaci.

### **4.5 Chytré bydlení (*Smart living*)**

Chytrým bydlením se rozumí zdravé a bezpečné bydlení v kulturně živém městě. Zahrnuje inteligentní budovy, které mají minimální náklady na vytápění a klimatizaci. Dům pracuje například i s informacemi o počasí. V jeho kontextu je myšleno také na bezpečnost města, kulturní vyžití, přístup k veřejné dopravě, atraktivita města pro investory a nové obyvatele.

### **4.6 Chytrá vláda (*Smart governance*)**

Moderní technologie by měli zjednodušit administrativu, snížit počet úřadů a vytvořit jednotná řešení pro veřejné služby. Zjednodušit komunikaci mezi obyvateli a městem, případně dalšími úřady, aby obyvatelé měli možnost efektivně vyřídit svoje záležitosti. Měla by být tímto zajištěna i transparentnost organizace města. Obyvatelům by řešení mělo přinést lepší informovanost o dění ve městě a možnost zapojit se do rozhodování.

## 5 Síťová struktura chytrého města

S rostoucí urbanizací a vznikem inteligentních služeb vzniká větší poptávka po kvalitním Internetovém připojení. Díky všudypřítomnému bezdrátovému pokrytí a páteřní síťové infrastruktuře by k ní měli přístup všichni občané města, návštěvníci a turisté. Stabilní a dostatečně dimenzovaná infrastruktura je navíc velmi důležitá pro všechny součásti chytrého města.

### 5.1 Město

Město představuje logickou vrstvu, která zprostředkovává spojení mezi pouliční vrstvou a Internetem, případně městským datovým centrem.

Jedná se o body, které sdružují veškerou komunikaci z pouličních přístupových bodů, na které jsou napojeny veškerá koncová zařízení. [2]

#### **Komponenty:**

- Kořenové AP – jedná se o mezičlánek mezi pouličními přístupovými body a datovým centrem. V tomto případě je tím myšlen router s firewallem zajišťující zabezpečení sítě a přístup do Internetu.
- Kořenové přepínače (Switche) – Zařízení obsahující velké množství síťových portů. Dále sdružuje pouliční přístupové body a udává správný směr toku dat.
- Mlhové zařízení (Fog devices) – jsou zařízení, které komunikují se serverem, ale dokáží také vyhodnotit určité situace samostatně. Zamezují zbytečného toku dat na server a zpět. Může to být kamera v chytrých autech, kdy je potřeba rychle vyhodnotit situaci a posílání dat na server, aby je vyhodnotil, je příliš zdlouhavé. Dalším takovým zařízením může být směrovač nebo přepínač.

### 5.2 Ulice

Logická vrstva, která spojuje koncová zařízení s městskou vrstvou. Tato síťová přístupová vrstva (Access layer network) zahrnuje chytré osvětlení, chytré parkoviště, přístupové body pro hotspoty Wi-Fi, řešení pro sledování bezpečnosti města, odpady a dalších čidel. [2]

Zařízení v této vrstvě zahrnují přístupové body k připojení na městskou síť, které jsou namontované na pouliční infrastruktuře, jedná se především o robustní přístupové spínače v pouličních skřínkách, síťové směrovače v celém městě a RFID (identifikace na radiové frekvenci) nebo Wi-Fi používané pro městské služby.

Senzory třetích stran a aplikace města jsou specializovanou oblastí.

- Senzory připojené pomocí kabelových a bezdrátových (RF, Wi-Fi, GPRS) propojení.
- Řídící jednotky pro ovládání modulů
- Systémy řízení ke sledování celkového využití, poruch, trendů atd.

Komponenty:

- Venkovní přístupové body
- Robustní pouliční přepínače
- Robustní směrovače
- Venkovní IP kamery
- Inteligentní služby s využitím senzorů

### 5.3 *Nástup IoT*

Všeobecně se jedná se o koncept připojení jakéhokoliv zařízení do internetové sítě domu, města, podniku. Lze do něj zahrnout vše od počítačů, přenosných počítačů, chytrých telefonů až po domácí spotřebiče jako je kávovar, pračka, pečící trouba, ale i například lampa a odpadkový koš.

Za první IoT zařízení je považováno dle článku *The little-known story of the first IoT device* [3] modifikovaný automat Coca Coly na univerzitě Carnegieho–Mellonových. Automat byl upraven tak, aby pomocí internetu posílal informace, zda automat není prázdný, nebo před jakou dobou byl automat doplněn. Poté kdokoliv, kdo byl připojen k místní síti, mohl jednoduše zjistit, zda se v automatu nachází Coca Cola a jestli je dostatečně vychlazená.

V roce 1999 poprvé použil koncept IoT Kevin Ashton při prezentaci pro společnost Procter & Gamble. Ashton chtěl přilákat pozornost vrcholového vedení na novou technologii zvanou RFID (Radio Frequency Identification) a protože byl Internet v roce 1999 novým trendem, nazval svoji prezentaci „Internet věcí“.

Bohužel termín Internet of Things nezískal žádnou rozšířenou pozornost. Koncept IoT začal získávat oblibu v roce 2010, kdy z úniku informací vyplynulo, že služba StreetView společnosti Google neudělala pouze 360° obrázky, ale že také uložila spoustu dat o Wi-Fi sítích. Ve stejném roce Čínská vláda oznámila, že v jejich pětiletém plánu učiní z IoT strategickou prioritu.

Termín „Internet věcí“ se dostal do širokého povědomí poté, co Google v roce 2014 oznámil, že koupí společnost Nest za 3,2 miliardy dolarů. Současně se na téma IoT konal největší světový veletrh spotřební elektroniky (Consumer Electronics Show) v Las Vegas. [4]

V dnešní době se nárůst zařízení komunikujících přes Internet šíří obrovskou rychlostí. Podle Analytické firmy Gartner, jak je uvedeno v článku A Simple Explanation of 'The Internet Of Things'[5], bude do roku 2020 více než 26 miliard připojených zařízení. Ale odhady se každým dnem mění a tak nám nezbyvá než si na rok 2020 počkat. IoT je tedy obrovská síť spojených věcí, které produkují data a komunikují mezi sebou.

Dnešní internetová síť funguje pomocí bezdrátových a kabelových sítí. S nástupem chytrých zařízení, díky kterým jsme stále připojeni do internetu, se tyto sítě staly mnohem více než jen prostým prostředkem pro komunikaci. [6] Nyní díky tomu sledujeme vlastní zdraví, naše každodenní zvyky, ale i spotřebu energie v domě.

Smart City a IoT jsou již dlouho diskutovány jako nový věk pro život a rozvoj měst. S příslibem zajistit větší bezpečnost, efektivnější využití energie a celkově lepší kvalitu života lidí žijících v městském prostředí je toto dobrá zpráva. Pozornost zaměřená na nejnovější technologie je správný směr, ale je třeba se také zaměřit na efektivní využití dat z nich plynoucí. Zde nastupuje spojení IoT a technologie BigData.



## 5.4 *Cloud Computing a Big Data*

Dvě základní technologie, které vstupují do běžných informačních technologií, jsou Big Data a Cloud Computing. Tyto technologie nejsou stejné. Jedna technologie, Big Data představuje obsah, nebo chceme-li data, zatímco druhá technologie, Cloud Computing představuje infrastrukturu pro ukládání těchto dat. Kombinací obou technologií můžeme dosáhnout výborných výsledků. Díky Cloud Computingu, můžeme využívat Big Data ke zpracování dat pohodlně v počítačových systémech a přenosných počítačích. Jednoduše řešeno, Big Data je velký objem dat, v dnešní době v řádech tera až peta bajtů. Data mohou být strukturovaná nebo i nestrukturovaná. Takové množství dat není možné zpracovávat tradiční databázovou a softwarovou technikou. [7]

Dle článku What is Cloud Computing [8] je Cloud Computing jednoduchý způsob, jak přistoupit k serverům, úložištím, databázím a dalším aplikačním službám přes internet. Uživatel si tedy zaplatí potřebné zdroje prostřednictvím cloud služeb na internetu a tím získá výpočetní výkon, úložiště databází a další zdroje na požadované úrovni. Podle článku Saas vs Paas vs IaaS: What's The Difference and How To Choose [9] se Cloud Computing dělí na tři základní modely.

- Infrastruktura jako služba (IaaS)
- Platforma jako služba (PaaS)
- Software jako služba (SaaS)

Každý z nich představuje samostatnou část.

**Infrastruktura jako služba (IaaS)**, přináší organizacím Cloud Computing strukturu, včetně serverů, operačních systému a úložišť, které jsou poskytovány pomocí virtualizace. Klienti tak mají kontrolu nad celou infrastrukturou a aplikacemi.

**Platforma jako služba (PaaS)**, poskytují komponenty pro cloud určitého softwaru. Nejčastěji se tedy využívá pro vývoj různých aplikací, kdy se firma nemusí starat o údržbu softwaru a soustředí se pouze na vývoj vlastní aplikace. Zdroje lze jednoduše měnit podle využití. Dále platforma poskytuje služby, které mohou pomoci při vývoji, testování a nasazování aplikací.

**Software jako služba (SaaS)**, představuje nejčastěji využívanou volbu pro firmy na trhu cloudu. Společnost využívá internet k poskytování aplikací svým uživatelům, které jsou spravovány třetí stranou. Nejčastěji jsou tyto aplikace spuštěny přímo přes webový prohlížeč a nevyžadují žádné stahování nebo instalace na straně klienta.

Cloud Computing a Big Data jsou tedy ideální kombinací. Poskytují řešení, které je škálovatelné a lze to přizpůsobit každému na míru. Díky tomu mohou být všechny informační zdroje snadno dostupné a data mohou být analyzována. Hlavní výhody spočívají v:

**Agility (Hbitost)** – Tradiční způsob ukládání a správy dat je značně nevyhovující. Nastavení síťové infrastruktury není jen drahé, ale také časově náročné. Instalace serverů a jejich následné spuštění může trvat i několik týdnů. Pomocí Cloud Computingu lze poskytnout libovolnou infrastrukturu se všemi potřebnými zdroji během krátkého času. [10]

**Affordability (Dostupnost)** – Cloud Computing je pro společnosti nové řešení, jak pracovat s daty. Společnosti si mohou vybrat, co potřebují a za poplatek mohou využívat zdroje pro správu velkých dat, kde i ušetří. Před Cloud Computingem společnosti investovaly nemalé peníze do zřízení a následně i do provozu IT oddělení infrastruktury. [11]

**Data processing (Zpracování dat)** – Množství dat vede k problémům se zpracováním. Například sociální media vytvářejí obrovské množství nestrukturovaných a chaotických dat, které se nedají zpracovat v rámci jedné kategorie. V kombinaci s Big Data a Cloud Computingem, lze snadněji zpracovat nestrukturovaná data a tím otevírá možnosti malým, středním ale i velkým společnostem. [11]

**Reduced complexity (snižování složitosti)** – implementace velkého datového řešení vyžaduje nemalé finanční prostředky na nákup velkého množství zařízení, komponentů a jejich softwarovou integraci. Cloud Computing poskytuje možnost automatizovat tyto požadavky, čímž snižuje složitost a zvyšuje produktivitu týmu Big Data Analysis. [10]

Dnešní největší úskalí cloudu je bezpečnost. Pro poskytovatele a firmy to je problém, který nelze snadno vyřešit. Útočníci stále vyvíjejí nové způsoby, jak najít slabinu, systém napadnout a najít v něm vstupní body. Mezi další komplikace patří ransomware, který hluboce ovlivňuje reputaci a zdroje společnosti, útoky typu Denial of Service, phishingové útoky a zneužívání cloudu.

## **5.5 Správa internetových služeb**

Správa internetových služeb je vrstva obsahující všechna zařízení a součásti potřebné k ověřování uživatelů, které spravuje provozovatel sítě, poskytovatel služby (SP – service provider), nebo partner spravovaných služeb (MSP – managed service partner) a také poskytovatel služby cloud [2].

Mezi další komponenty internetových služeb patří

- Server zásad sítě (NPS)
- SMS brána (SMS Gateway)
- Připojení k internetu (Internet connectivity)
- Aplikace řízení snímačů (Management application of Sensors)

### **Inteligentní brána služeb (ISG)**

Podle Overview of ISG [12] je inteligentní brána služeb strukturovaný rámec, ve kterém zařízení pro přístup ke koncovým zařízením poskytuje předplatitelům flexibilní a škálovatelné služby. Zpracovává následující klíčové aspekty předplatitelů:

- Identifikace účastníka
- Určení služeb a přístupu
- Účtování při přístupu a využití služby
- Monitorování stavu návštěvy
- Řízení životního cyklu relace

Inteligentní brána služeb zavádí dynamický prvek pro poskytování a aktivaci služeb, pomocí řídicích politik a rozšíření změn autorizace (CoA) na protokol RADIUS.

Zařízení s podporou inteligentní brány služeb, může být nasazeno na hranici přístupu a na okraji sítě. Služba je použitelná pro celou řadu prostředí, jako je drátová, veřejná bezdrátová anebo mobilní bezdrátová síť. Služba byla navíc navržena tak, aby byla co nejvíce flexibilní.

### **5.5.1 Zabezpečená překryvná síť (Secure Overlay Network) pro služby Chytrého města**

Podle článku Network Infrastructure Smart Cities [2] Městská Wi-Fi síť funguje jako překryvná síť IP, která umožňuje hostit a využívat různé případy, služby a zařízení pro použití v městských oblastech. Následující vlastnosti městské sítě umožňují městu a jeho prodejčům nasadit zařízení ve městě a hostit více služeb pomocí městské Wi-Fi.

Na ulici by měla být dostupná bezdrátová síť, kabelová síť a to spolu s možnostmi PoE a PoE+ (Power over Ethernet), což umožňuje napájet zařízení pomocí datového kabelu. Rozdílem mezi těmito dvěma druhy je množství dodané energie. Zatímco PoE dokáže dodat pouze okolo 15 Wattů přes kroucenou dvojlunku kategorie 5, PoE+ dokáže přes stejný kabel 30 Wattů s 25 Watty pro zařízení a tím umožňuje napájet i složitější zařízení jako jsou například IP kamery nebo různé alarmy. [13]

Dále by měla být dynamické poskytování VLAN bezpečně na základě 802.1x nebo MAB. Jedná se o dva způsoby ověřování identity, zatímco v prvním zmiňovaném se ověření provádí například na základě jména a hesla případně bezpečnostního certifikátu, jsou zde i zařízení, která protokol 802.1x nemají implementovaný. U těchto zařízení využíváme druhý zmiňovaný protokol MAB, který provádí ověřování pomocí MAC adresy.

Na základě funkcí městské sítě Wi-Fi, ji můžeme využít k různým inteligentním službám po celém městě, jako je inteligentní osvětlení, inteligentní parkování, správa provozu, nakládání s odpady atd.

## 6 Fungování chytrého města

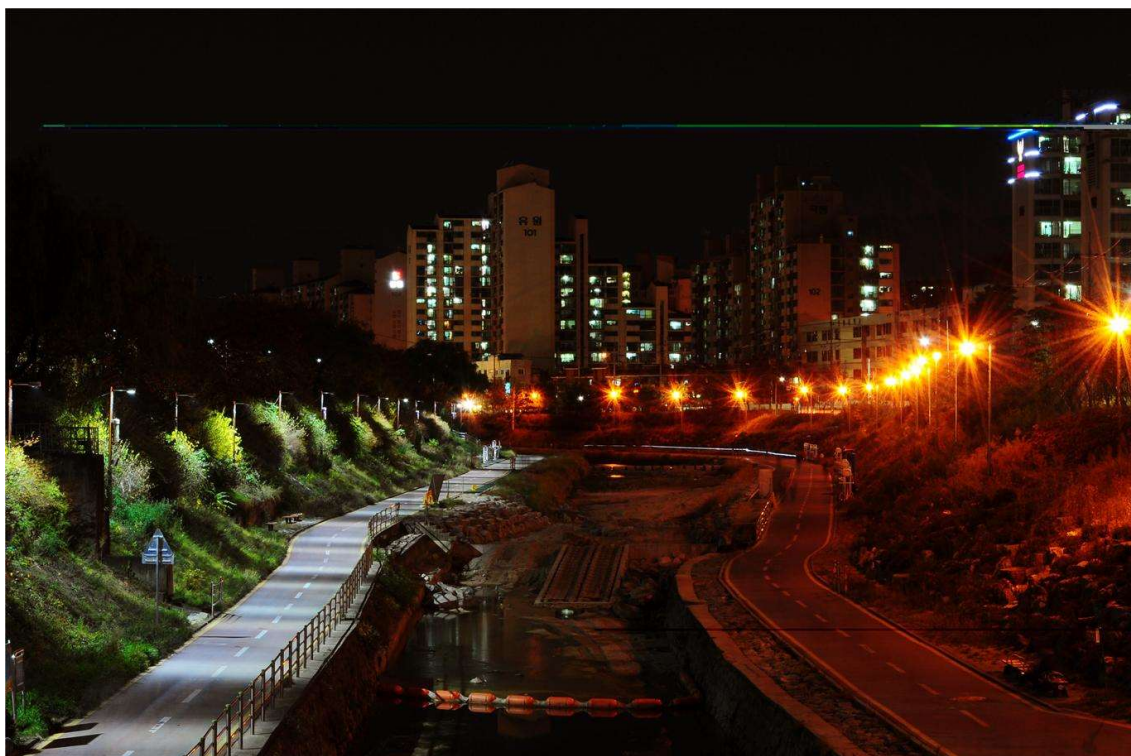
Podle Smart Cities: Whys, hows and whats [14] k fungování chytrého města je potřeba efektivní využití všech dostupných zdrojů, což znamená sdílení informací v propojené a integrované síti. Tyto data pomáhají městu při poskytování nových služeb, ale také optimalizaci stávajících. Díky tomu se mohou snížit výdaje města za služby na minimum a tím dosáhnout snížení výdajů města za služby. Dále se nabízí občanům možnost zapojit se k akcím města na zvýšení kvality života a celkové životní úrovně.

Klíčovým termínem je inteligentní město. [15] Je to město, které využívá data a technologie ke zlepšení života občanů a podniků. Klíčovou technologií za úspěchem Smart City je IoT. Výměna množství dat pomocí internetové sítě mezi zařízeními, jako jsou vozidla, domácí spotřebiče a senzory sledující situaci ve městě, budou moci umožnit nepřehledné množství výhod a zefektivnit tak život všech lidí ve městě. Pomocí dat budeme moci v budoucnu analyzovat situace, které by mohli nastat.

Pokud spojíme práci IoT a Big Data v dopravě chytrého města, můžeme získat dostatek dat k tomu, abychom mohli odhadnout, jaké množství dopravy může být únosné v určitém časovém okamžiku. Informace půjdou shromažďovat v reálném čase pomocí senzorů, kamer, nositelných zařízení a chytrých zařízení. V rámci rozsáhlé analýzy dat se rodí základní systém rozhodování. Systém rozdělí informace a sdílí tyto informace mezi více systémů. [16] Toto řešení shromažďuje všechny formy dopravních informací pomocí snímačů pro usnadnění sledování a kontroly provozu v reálném čase. Navíc tato řešení představují schopnost předvídat dopravní trendy založené na realistických simulacích a matematickém modelu.

Tato analýza dat se používá k tomu, aby pomohla identifikovat příčiny přetížení městské dopravní sítě, mohou za tím stát přírodní podmínky, snížená viditelnost ale například i školní prázdniny. Správci se tedy mohou snadno podívat na příčiny vzniku problému, případně těmto problémům předcházet díky datům z většího časového úseku. Poté by systém v případě problému mohl odeslat řidičům informaci o alternativních cestách městem, a tím ulevit přetíženým silnicím. Jelikož má doprava významný dopad na životaschopnost a efektivitu města, můžou senzory a zařízení pomoci řídit dopravu efektivně bez ohledu na nárůst počtu obyvatel. Dále mohou senzory pomáhat s parkováním. Aplikace města by mohla navést člověka na volné parkovací místo a umožnit toto parkovací místo zarezervovat a rovnou zaplatit parkovný. Tím by se ušetřilo spoustu času lidem ve městě a množství aut v ulicích.

Další možností je například Chytré osvětlení, kdy lampy rozpoznají, pokud pod nimi prochází člověk a v tu chvíli se rozsvítí, případně zvýší intenzitu. Toto řešení může pomoci snížit světelné znečištění, ušetřit náklady a usnadnit správu veřejného osvětlení ve městě. Velmi důležité je také usměrnit tok světla správným směrem, aby svítilo, jen tam, kam má svítit, viz obrázek číslo 1, kde máme porovnání dvou technologií osvětlení.



**Obrázek 1: Vylepšené veřejné osvětlení vlevo, vpravo současné osvětlení.**  
[17]

## **7 Současný stav Smart Cities**

V současné době je Smart Cities na vzestupu, mnoho měst po celém světě ale i v České Republice, vypracovává různé projekty, jak město zmodernizovat a získat představu o investicích. Zatím žádné město není vyloženě Smart City, ale mnoho světových metropolí se k tomu blíží.

Například Barcelona [18] patří mezi jedno z nejvyspělejších měst, kde funguje chytré parkování a celkově chytrá doprava. Dále se zde sleduje kvalita ovzduší v ulicích, využívá se chytré osvětlení, pláže a parky jsou pokryty wifi signálem tak, aby se mohl kdokoliv připojit na internet. Město, které začíná zapadat do konceptu Smart City je Malmö. Toto Švédské město se podle Smart City: Cesta za lepším životem ve městě [19] řadí mezi nejvyspělejší města na světě, díky výstavbě čtvrtí, které splňují vizi Smart City a využívají energii z obnovitelných zdrojů.

Dalším městem, které dělá velké pokroky je Německé město Heildenberg, podobně jako Malmö staví novou čtvrť Bahnstadt na 116 hektarech. Cílem je mít budovy s tzv. pasivním standartem. To znamená, že budovy produkují minimální emise a snaží se o co největší úsporu energie. Součástí budou i vertikální zahrady a veškerá energie k napájení by měla jít z obnovitelných zdrojů, jako jsou větrné elektrárny, solární panely a vodní energie.

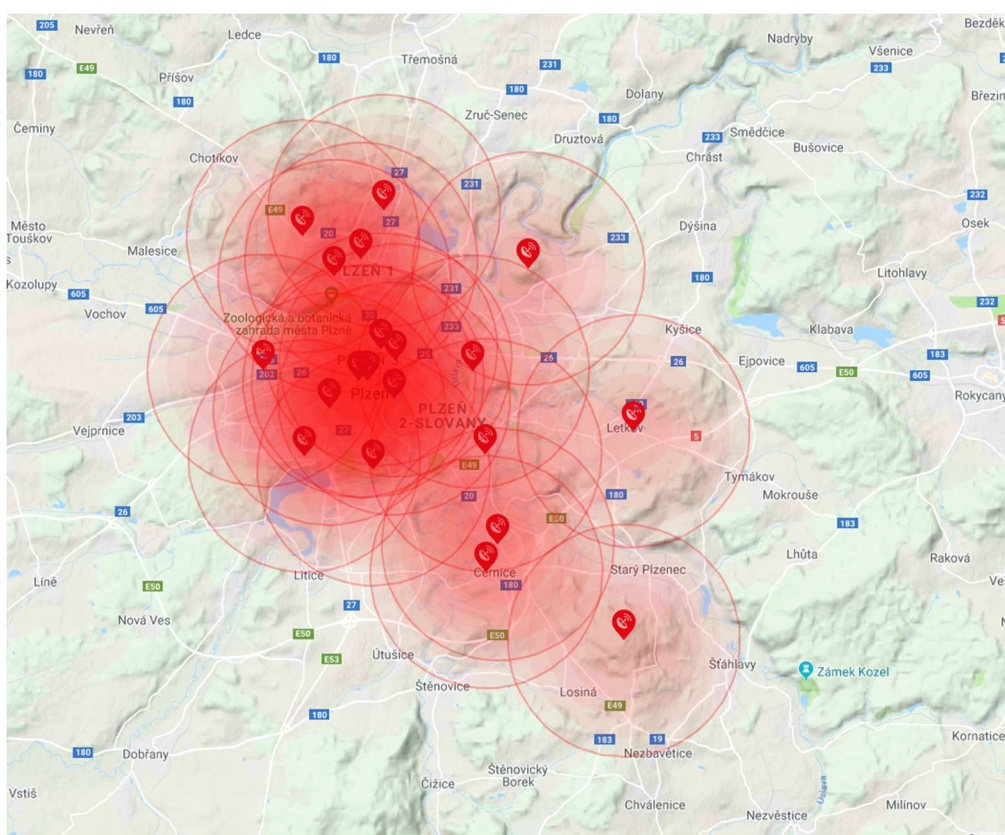
V České Republice se veškeré projekty okolo Smart City teprve rozbíhají, hodně měst si nechává vypracovávat studie a projekty na míru a snaží se získat vizi do budoucna. Například Brno má vizi až do roku 2050, s jasnými plány do roku 2028.

I Praha má vytvořenou vlastní vizi do roku 2030. Podporuje především rozvoj turistiky, kdy chce vytvořit aplikace pro turisty, která by měla zlepšit orientaci po městě a případně je přivést i na méně známá místa. Dále se zvyšují počty dobíjecích stanic pro elektromobily. Do roku 2030 by mělo být v Praze více jak 100 dobíjecích stanic.

Svůj koncept Smart City má i Hradec Králové. Tento koncept vytvořila firma GIST s.r.o., kde je popsáno, jak se bude postupovat při realizaci a také ustanovili řídicí strukturu.



První město v republice, které má vybudovanou komunikační síť pro IoT zařízení je Plzeň. Podařilo se jim během roku 2017 vybudovat IoT síť LoRaWAN. Využívají pásmo 868 MHz a je to založené na technologii LPWAN, jedná se o bezdrátovou síť, na energeticky nenáročnou komunikaci s velkým dosahem. [20] Přístupové body jsou již rozmístěny po celé Plzni a okolí, přičemž přístup k nim je bez omezení pro studenty a městské organizace. Město jim poskytuje možnost zapojit a otestovat vlastní senzory a nabízí jim i možnost zpracování získaných dat. Na obrázku číslo 2 máme názorně zobrazené pokrytí.



**Obrázek 2: Mapa pokrytí města Plzně přístupovými body sítě PLZEN LoRa [20]**

## 8 Napojení Smart technologií do současné sítě

Pro praktickou část byla vytvořena modelová situace, která se soustředí na běžnou situaci, kdy města v zimních měsících řeší problémy související se změnami povětrnostní situace. Vysoká atmosférická vlhkost nebo srážky společně s nízkými teplotami komplikují dopravu a pohyb chodců po komunikacích. Na frekventovaných místech může včasný zásah odvrátit krizovou situaci. Města běžně reagují na hlášení občanů z daných lokalit nebo konají preventivní ošetření, což není zcela efektivní. Výjezd na základě hlášení o úrazu způsobeného pádem osoby na zledovatěném povrchu znamená, že informace přišla pozdě. Preventivní ošetření povrchu chodníků a vozovek je spojený s potenciálně zbytečnými výdaji i zbytečnou ekologickou zátěží. Pokud dochází k dynamickému růstu města, je složitější informovat obyvatele.

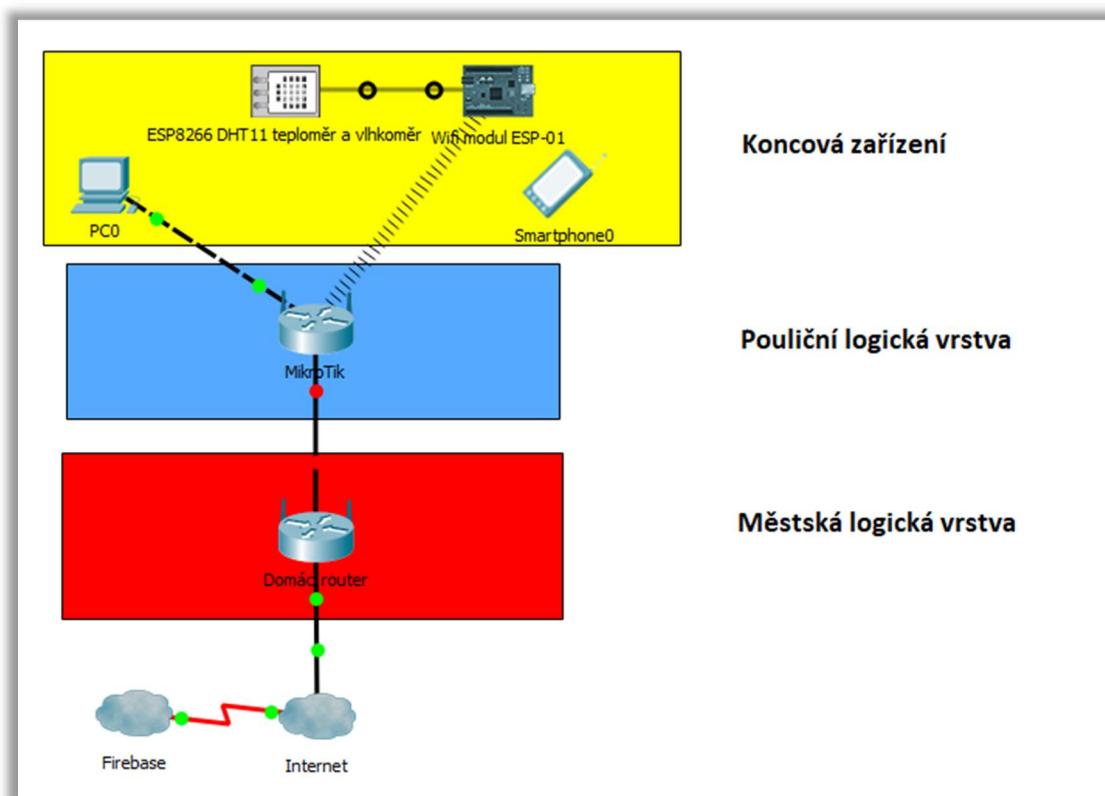
Snahou v praktické části je proto vytvořit modul, který by byl základem pro prezentaci aktuální situace. Výstupem je veřejné pasivní poskytnutí relevantní informace s možností nechat si zaslat upozornění na potenciální riziko vzniku náledí na konkrétním místě města. Toho může být dosaženo napojením monitorovacího zařízení do datové sítě obce a prezentace jeho dat v rámci informačního systému.

Dále uvedený model má prokázat možnost realizace a odhalit možná úskalí implementace. Při návrhu je třeba se zaměřit na technické parametry a splnit jisté požadavky. Monitorovací zařízení musí být miniaturní a autonomní, nezávislé na externím zdroji napájení. Zapojení do datové sítě města by nemělo vyžadovat výrazné investice do její modifikace. Při budování moderních měst se počítá s bezdrátovou sítí pro pokrytí potřeb různých systémů. Zařízení proto primárně využívají bezdrátovou komunikaci. Jako vhodné se jeví doplnění informace o teplotě a vlhkosti v daném místě o informaci obrazovou. Ideální je spojení s IP kamerou, které v mnoha případech již město využívá pro zajištění bezpečnosti.

Při realizaci bylo připojení senzorů teploty a vlhkosti realizováno přes bezdrátový přístupový bod, který je napojen na domácí síť. Jejím prostřednictvím jsou data odesílána ze senzorů do databáze. Dále byla vytvořena webová aplikace, která zobrazuje aktuální data a aktuální obrázek IP kamery. V aplikaci je také možnost pro přihlášení administrátora, který touto cestou může zadat upozornění pro obyvatele, jako reakci na naměřené hodnoty, případně na nenadálé události ve městě.

Dnes již nedílným doplňkem k webové aplikaci je aplikace pro mobilní platformu. Zde byla pro otestování zvolena podpora pro Android. Podobně jako webová aplikace ukazuje aktuální teplotu a vlhkost ze senzoru a aktuální upozornění.

Na obrázku číslo 3. je zobrazen návrh řešení naší situace, kde jsou zobrazeny naše testovací zařízení a co by měli představovat v rámci městské datové sítě.



Obrázek 3 Návrh řešení situace, zdroj: vlastní

## 8.1 **Použitá zařízení a služby**

### 8.1.1 **Mikrotik RB493G**

Jedná se o síťový prvek, který v našem případě poskytuje spojení mezi naším zařízením a vnějším světem. Zařízení RB493G má ve výbavě procesor Atheros AR7161 taktovaný na 680 MHz a operační paměť o velikosti 256 MB typu DDR. Máme zde i velké možnosti, co se týče rozhraní. Zařízení poskytuje devět LAN portů RJ-45, tři sloty miniPCI pro připojení Wi-Fi karty, jeden sériový port RS-232 a USB port 2.0, díky kterému lze připojit externí paměť a využít zařízení jako FTP úložiště. Společně k našemu síťovému prvku je dodáván systém RouterOS v3 L5, který je uložen v paměti NAND o velikosti 64 MB. Napájet lze buď DC adaptérem nebo pomocí PoE (Power over Ethernet). Poslední zmíněné je napájení datovým kabelem. Uvedené zařízení lze nakonfigurovat do několika operačních módů, například Switch, Router nebo Bridge.

### 8.1.2 **Mikrotik miniPCI R52nM**

Pro možnosti Wi-Fi jsme zařízení RB493G rozšířili o kartu R52nM, která umožňuje využívat bezdrátové připojení standardem 802.11n. Samozřejmě umožňuje použít i standardy 802.11a/b/g. Tato síťová karta nabízí rychlost přenosu až 300 Mb/s možností zabezpečení WPA, WPA2, WEP a 802.1X.

### 8.1.3 **ESP8266 DHT11 teploměr a vlhkoměr + Wi-Fi modul ESP-01**

Pro měření a odesílání dat byla použita zařízení od společnosti Arduino. Jde o open-source hardwarovou a softwarovou platformu podpořenou projektovou a uživatelskou komunitou, která navrhuje a vyrábí sady mikrokontrolerů pro budování digitálních zařízení a interaktivních objektů, které lze ovládat fyzicky či naprogramovanými příkazy.

Pro realizaci byly použity dva moduly, ESP8266 DHT11 a ESP-01. První jmenovaný dovoluje monitorovat aktuální teplotu v rozsahu 0 až 50 °C s přesností  $\pm 2$  °C a vlhkost vzduchu od 20 až 90 RH% s přesností  $\pm 5$  %. Druhý modul pak slouží k bezdrátovému připojení k datové síti. Podporuje Wi-Fi standardy 802.11b/g/n při výkonu vysílače 19,5 dBm a zabezpečením WPA a WPA2.

Za doplnění stojí také EPS-01 USB ESP8266 Seriál Wifi Adaptér. Do tohoto zařízení se připojuje náš programovatelný modul ESP-01 a pomocí USB dojde ke spojení s počítačem. Díky tomu můžeme náš modul bez větších problémů programovat, případně zachytávat různé chyby v programu.

#### **8.1.4 Napájení monitorovací jednotky**

Napájecí napětí celé sestavy je 3,3 až 3,6 VDC. Aby mohlo zařízení fungovat samostatně a nemuselo být připojeno do elektrické sítě, je nutné použít správnou baterii. Pro naše účely byla vybrána Li-Ion Sony 18650 30A 3000mAh. Technologie Li-Ion baterií nabízí nominální napětí článku 3,60V a hustotu energie v rozmezí 255-676 Wh/L (watt hodin na litr). [21]

Akumulátory NiMH jsou oproti Li-ion bateriím o něco těžší a větší. Při přibližně stejné velikosti článku nabízí pouze napětí 1,5V s maximální kapacitou 2700 mAh. Bohužel takto vysoká kapacita po krátkém čase rychle ubývá. Hlavním problémem akumulátoru je takzvaný paměťový efekt, kdy během několika nesprávných nabíjecích cyklů, může dojít k výraznému poklesu kapacity akumulátoru. Tím Li-Ion baterie téměř netrpí, navíc mohou být nabity a vybity rychleji než NiMH. [22]

#### **8.1.5 Firebase**

Jedná se o vývojovou platformu pro webové a mobilní aplikace, která je dostupná online pomocí webového prohlížeče. Tato platforma byla vyvinuta společností Firebase roku 2011 a odkoupena roku 2014 společností Google. Platforma nabízí několik produktů, které dnes využívá přes 1,5 milionu aplikací.

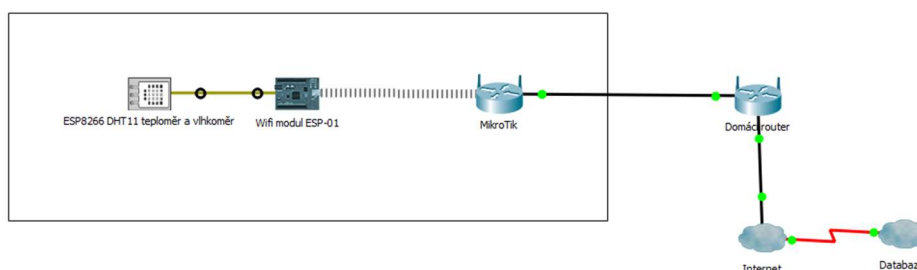
Mezi hlavní produkty patří Firebase Real-time Database, což je NoSQL databázový systém, který zpracovává data v reálném čase. Hodí se pro sledování aktuálních hodnot, které se neustále mění a případně lze na ně rychle reagovat. Dalším produktem, který zde zájemci Google nabízí je "hosting". Jedná se o jednoduchý produkční webhosting, ve kterém lze velice snadno nasadit webové aplikace.

Všechny služby potřebné pro popisovaný projekt byly obsaženy v takzvaném Spark plánu, který je dostupný zdarma. Samozřejmě jsou zde určitá omezení, která se dají za příplatek eliminovat, případně zakoupit celé plány, které obsahují další služby. [23]

## 8.2 Příprava a programování

### 8.2.1 Připojení do sítě

Pro připojení našeho modulu ESP-01 od Arduina, bylo zapotřebí provést konfiguraci síťového zařízení RB493 s implementovanou Wi-Fi kartou R52nM. Bylo využito jako spoj mezi senzorem a Internetem, kde se nachází databáze pro ukládání dat, viz obrázek číslo 4. Router byl přepnut do operačního modu Bridge na němž byla nastavena služba dynamického přidělování IP adres (DHCP). Rozsah adres byl přemostěn z domácího routeru v rozsahu 192.168.1.200-255. Pro Wi-Fi, bylo zvoleno SSID jako „MikroTik-Arduino“ s heslem „arduinoTest“. Jako zkouška naší konfigurace byl test připojení notebooku k Internetu.



Obrázek 4 Schéma zapojení, zdroj: vlastní

## 8.2.2 Příprava databáze

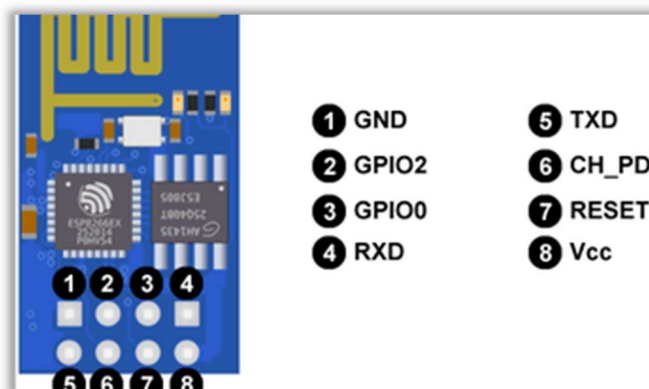
Abychom mohli ukládat data ze senzorů a mít k nim přístup pro webovou a mobilní aplikaci, bylo nutné založit projekt ve Firebase. Po vytvoření projektu jsme si vygenerovali a otevřeli naši databázi. V rozhraní jsme zvolili možnost zabezpečení. Zde jsme upravili pravidla, abychom mohli číst data bez omezení a zapisovat pouze pokud budeme přihlášení, viz obrázek číslo 5.

```
1 {
2   "rules": {
3     ".read": "true",
4     ".write": "auth != null"
5   }
6 }
```

Obrázek 5 Oprávnění Firebase Realtime Database, zdroj: vlastní

## 8.2.3 Programování ESP-01 se senzorem ESP8266 DHT11

Pro programování ESP-01 bylo použito vývojové prostředí Arduino IDE. Abychom mohli začít programovat, bylo nutné zapojit modul do USB adaptéru a propojit piny 1 a 3 dle přiloženého obrázku číslo 6. Je na něm zobrazeno schéma modulu ESP-01 s popisem pinů. Tím jsme přepnuli naše zařízení do programovacího módu, který nám umožnil nahrát náš kód. Po nahrání stačilo USB adaptér s připojeným ESP-01 odpojit od počítače a přerušit spojení pinů. Takto jsme dosáhli přepnutí zpět do klasického módu.



Obrázek 6 ESP-01 pin diagram [24]

Po zapnutí vývojového prostředí Arduino IDE, bylo nutné nainstalovat knihovny, které nám umožnili využívat veškeré potřebné nástroje. Nejdůležitější knihovna byla esp8266, která umožnila ovládání modulů a Wi-Fi a dále knihovna DHT sensor pro senzory.

Pro správné fungování navrhovaného systému, bylo nutné získat aktuální datum a čas. Existuje několik způsobů, jak tyto údaje obdržet. Jedna z možností je zapojení hardwarového modulu se systémovým časem do obvodu, což by znamenalo jednak se starat o aktuálnost jeho nastavení, chceme-li synchronizaci, a také vyšší spotřebu elektrické energie. Druhá možnost byla využít NTP Client knihovnu, která nám umožní získat aktuální datum a čas z Internetu. Samotný protokol NTP (Network Time Protocol) se stará o to, aby všechna zařízení, které jsou připojená do sítě, měla stejný a pokud možno přesný čas.

Po dokončení instalace všech potřebných knihoven, bylo zapotřebí stáhnout poslední aktualizaci takzvané „vývojové desky“. V našem případě se jednalo o Generic ESP8266 Module. Po splnění všech výše uvedených požadavků a postupů, bylo možné přistoupit k samotnému programování.

Nejdříve bylo nutné připojit modul na naší Wi-Fi, tím jsme získali přístup do datové sítě a díky tomu bylo možné data ze senzoru odeslat do databáze. Jelikož mezi databází a naším modulem bude probíhat komunikace, bylo zapotřebí zabezpečit toto spojení, aby nemohlo být narušeno nebo odposloucháváno. K tomuto účelu jsme využili TLS (Transport Layer Security). Pomocí `WifiClientSecure` byla nadefinována proměnná `client` a pomocí `client.connect` byl určen cíl, ke kterému se chceme připojit, v našem případě odkaz na databázi a port. Byl použit port 443, který využívá šifrovaný přenos http protokolu přes TLS (HTTPS). Po úspěšném navázání spojení, bylo potřeba porovnat otisk certifikátu.

Otisk lze získat pomocí prohlížeče, kde pod zabezpečením stránky lze najít certifikát, kde se nachází i jeho kryptografický otisk, který je nutné zkopírovat do našeho kódu. Po připojení se vezme otisk certifikátu na serveru a porovná se s naším otiskem, který jsme zadali. Tím bylo zajištěno, že jsme připojeni ke správnému serveru a že máme bezpečné připojení. Problém u této metody nastává ve chvíli, kdy platnost certifikátu skončí, případně je vytvořen nový. Poté nám nezbyvá nic jiného než, otisk obnovit v samotném kódu.



Po tomto porovnání jsme se mohli připojit do databáze a začít odesílat data. Bylo tedy potřeba získat aktuální čas a datum ve správném formátu. K tomuto účelu nám posloužil již avizovaný NTP client. Jako poslední krok nám chyběla samotná data ze senzoru, zde našel využití náš modul ESP8266 DHT11.

Po získání těchto dat a připojení do databáze, nám stačilo data odeslat. Jelikož ukládáme teplotu a vlhkost, bylo nutné tyto data rozlišit. Byl nastaven cyklus na dva průchody, kdy poprvé ukládáme vlhkost a při druhém průchodu ukládáme teplotu. Vždy bylo potřeba nastavit pomocí `client.print` cestu a časový otisk. Dále proběhlo ověření, s nastavením http verze 1.1. Poté proběhlo nastavení adresy databáze a připojení. To bylo nastaveno na `keep-alive`, tím jsme zajistili, že můžeme přenést více dat pomocí jediného spojení a tím ušetřit zatížení CPU a paměti modulu. V poslední části jsme nastavili počet znaků odesílané informace, a nakonec samotnou zprávu, chceme-li data. Celý proces lze vidět na přiloženém obrázku číslo 7.

```
if (client.connect(FIREBASE_HOST, 443)) {
  getTime();           // získání aktuálního data a času
  getData();           // získání dat ze senzoru ESP8266 DHT 11
  String type;
  float value;
  int i = 0;
  while(i != 2){
    if(i == 0){
      type = "/Humidity/"; // při prvním průchodu ukládáme vkhkost
      value = h;           // uložení aktuální hodnoty vkhkosti
    }
    else{
      type = "/Temperature/"; // při druhém průchodu ukládáme teplotu
      value = t;           // uložení aktuální hodnoty teploty
    }
    client.print("PUT /DHT11_" + dayStamp + "" + type + "" + timeStamp + ".json?auth="); //nastavení cesty
    client.print(FIREBASE_AUTH); // ověření
    client.print(" HTTP/1.1\r\n"); // nastavení HTTP verze 1.1
    client.print("Host: "); // nastavení hosta
    client.print(FIREBASE_HOST);
    client.print("\r\n");
    client.print("Connection: keep-alive\r\n"); // nastavení spojení na keep-alive
    client.print("Keep-Alive: timeout=30, max=100\r\n");
    client.print("Content-Length: 5\r\n\r\n"); // nastavení délky obsahu zprávy
    //getData();
    client.print(value); // zpráva
    delay(1000); // časová prodleva
    i++;
  }
}
else {
  Serial.println("Connect failed!");
}
ESP.deepSleep(sleepTime * 1000000); // uvedení zařízení do režimu spánku
```

Obrázek 7 - Zabezpečení a odeslání dat do databáze, zdroj: vlastní

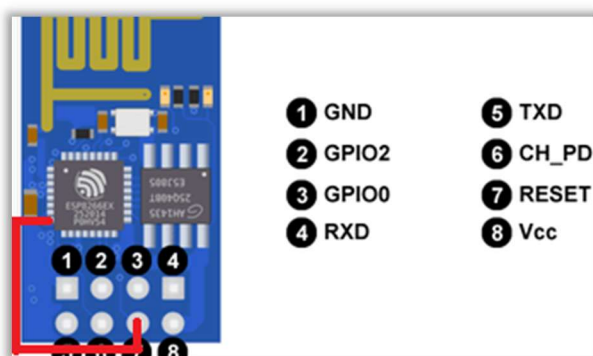
Po odeslání dat bylo potřeba nastavit, jak často chceme, aby se tato část opakovala. Pokud bychom chtěli posílat data například každou minutu, lze jednoduše nastavit zpoždění na jednu minutu `delay(60000)`. Při této metodě nám ale modul stále bere energii mezi 50 mA až 170 mA, podle toho zde právě modul něco dělá. Pokud bychom nechali zařízení v tomto nastavení, byla by odhadovaná výdrž okolo 25 hodin, což je vzhledem k plánovanému nasazení nevyhovující. Pro úsporu energie v baterii, nám modul nabízí hned několik módu, které se dají využít, viz obrázek číslo sedm. Prvním je Modem-sleep, ten nám vypíná pouze Wi-Fi a spotřeba zařízení klesne zhruba na 15 mA. V tomto módu je možná doba provozu až osm dní.

Lepší variantu nabízí tzv. Light-sleep, zde mimo vypnutí Wi-Fi a dalších částí, přepne procesor do režimu „čekající“. Modul pak má spotřebu okolo 0,4 mA, což prodlouží výdrž až na 312 dní. Třetím módem je takzvaný Deep-sleep, který vypíná téměř všechny funkce včetně procesoru a nechává spuštěný jen RTC (Real-time clock). Tato funkce nám udržuje odpočet, za jak dlouho se modul spustí. V tomto režimu modul spotřebovává okolo 20  $\mu$ A a tím prodlužuje životnost naší baterie. Při použití tohoto modu by baterie vydržela celých 17 let. Ale je samozřejmě nutné započítat i čas kdy zařízení odesílá data a odběr systémových led žárovek.

V naší konfiguraci s módem Deep sleep, se svítícími systémovými led žárovkami a s frekvencí odeslání jednou za deset minut, by nám energie měla vydržet až 1000 hodin. Vše bylo změřeno multimetrem Emos EM391. Výpočet byl jednoduchý. Při odeslání jsme naměřili průměrnou hodnotu 75 mA během 8 vteřin. Dalších 10 minut, v přepočtu 600 vteřin, nám modul odebírá 1,9 mA v módu Deep sleep, zde narážíme na problém systémových led žárovek. Průměrná hodnota odběru byla vypočítána:  $(8 \cdot 75 + 600 \cdot 1,9) / 608 = 2,861$  mA. Náš akumulátor má při plném nabití 3000 mAh, to vydělíme naším výsledkem.  $3000 / 2,861 = 1048$  hodin, to znamená že zařízení může teoreticky vydržet zapnuté až 43 a půl dne.

Deep sleep lze aktivovat jednoduchým způsobem, napsat příkaz `ESP.deepSleep()` a jako parametr uvést čas na jak dlouho chceme, aby zařízení bylo v tomto režimu.

Poté co bylo zařízení v hlubokém spánku, ubíhal námi nastavený čas. Po odpočtu nám zařízení pouze probliklo bez jakékoliv reakci, ač mělo odeslat aktuální data. Po hlubším prostudování postupů bylo zjištěno, že je nutné propojit krajní pin procesoru s pinem RESET(7), viz obrázek číslo sedm. Tím jsme zajistili, že pokud modul přepneme do stavu Deep sleep, tak se z něho může i probudit.



Obrázek 8 Propojení krajního pinu CPU a pinu RESET [24]

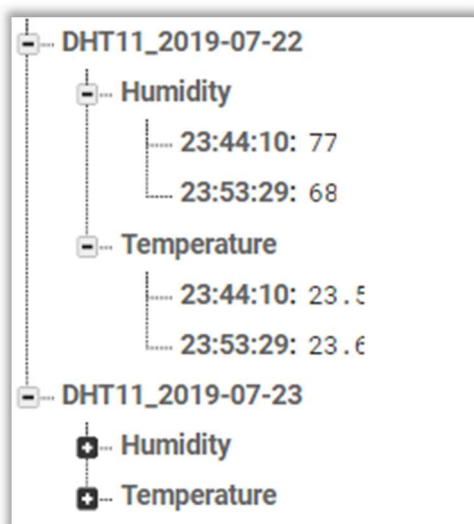
## 8.2.4 Programování webové aplikace

Naše webová aplikace měla sloužit k zobrazení aktuální teploty a vlhkosti senzoru, případně by měla zobrazovat aktuální upozornění. S tím souvisí přihlášení do systému a úprava těchto upozornění, jako je vytváření nových nebo mazání starých oznámení. Bylo tedy potřeba vytvořit html stránku, která by nám tyto data ukázala kdykoliv a kdekoliv, pokud bychom měli přístup k Internetu. V reálu pak může být informace doplněná o snímek z IP kamery sledovaného prostoru pro dokreslení situace. Tento detail zde nebyl řešený z důvodu absence IP kamery. Pro naše účely byl snímek z kamery nahrazen aktuálním snímkem z náměstí ve městě Velké Meziříčí.

Pomocí tabulkových html tagů a využitím CSS (Cascading Style Sheets) byla vytvořena přehledná webová aplikace, která umožňuje zobrazit zaznamenané hodnoty.

Pro samotné získávání dat bylo využito JavaScriptu. Ale ještě předtím jsme museli naimportovat Firebase JS SDK [25], abychom získali veškeré přístupy k balíčku Firebase, který umožňuje využívat další funkce, jako je přihlášení a databázi samotnou. Poté se musel inicializovat náš projekt. Po inicializaci jsme mohli přejít k avizovanému získávání dat.

Aby se vše zobrazilo, jak potřebujeme, bylo zapotřebí vzít do úvahy formát dat uložených v naší NoSQL databázi a zjistit možnosti získání a zobrazení našich hodnot. Jak můžeme vidět na obrázku číslo osm, první část cesty je `DHT11_YYYY-mm-dd`. Další krok se dělí na `Humidity` a `Temperature`. Kde máme naše hodnoty uloženy s jejich klíčem, který je zároveň čas kdy byly hodnoty pořízeny. Struktura uložení dat je zobrazena na obrázku číslo 9.



**Obrázek 9 Uložení dat v databázi, zdroj: vlastní**

V našem kódu jsme tedy nejdříve museli získat poslední odkaz DHT11 s nejbližším datem. Použili jsme tedy vestavěný příkaz:

```
firebase.database().ref().orderByKey();
```

Tento krok nám získal a seřadil veškeré odkazy. Nyní nám stačilo vzít poslední hodnotu. K tomu byl použitý příkaz `limitToLast(1)` a pomocí `forEach` získali poslední `childSnapshot`, ze kterého jsme získali konečný klíč. V našem případě první část cesty.

Po tomto kroku jsme mohli přejít k získávání samotných dat, jelikož první část jsme získali a další část cesty k datům známe. Postup byl ale podobný jako v předchozím případě. Do příkazu jsme doplnili referenci, kdy naše proměnná `journeKey` je ta část, kterou jsme získali v předchozím případě a další část je buď odkaz na teplotu nebo na vlhkost:

```
firebase.database().ref("" + journeyKey + "/Temperature").orderByKey();
```

Po získání `childSnapShot` jsme mohli získali výslednou hodnotu, v tomto případě teplotu. Ale abychom věděli, časový údaj, kdy tyto data pořízena, použili jsme klíč našeho výsledku.

Jelikož chceme čas a datum zobrazit ve formátu `dd.mm.yyyy hh:mm:ss` bylo nutné první řetězec rozdělit pomocí funkce `substring` na roky, měsíce a dny. Poté už nám stačilo pouze řetězec spojit do formátu podle našich představ.

K zobrazení dat bylo potřeba zvolit správný ID html tag a poté jsme příkazem `replaceWith` zajistili, že budeme zobrazovat pouze poslední získanou hodnotu.

Dále jsme přidali do webové aplikace zobrazení upozornění. Veškerá upozornění se získávají podobně jako data v předchozích případech. Vytvořili jsme metodu `getNotification()`, zde jsme zadali referenci do databáze, kde jsou uloženy všechny naše upozornění. Poté nám cyklus `forEach` získal při každém průchodu takzvaný `childSnapshot`, z něhož jsme podle názvu získali potřebná data. Naše upozornění má nadpis `title` a zprávu `message`. V každém takovém cyklu jsme tedy využili příkazu `childSnapshot.val().title/message` a data uložili do proměnných. Poté jsme data odeslali, aby se výsledně zobrazila na stránce. K tomu jsme využili odkaz na id „upozorneni“ s kombinací `after`, kdy se zapíšu data z každého cyklu za poslední záznam. Výpis byl modifikován o html prvky, abychom od sebe oddělili různé upozornění ale také nadpis a samotnou zprávu. Příkaz byl výsledně zapsán takto: `$("#upozorneni").after("<tr><td style='text-align: left; border-bottom: 1px solid black;'><h6>" + title + "</h6></br>" + message + "</td></tr>");`

Pro modifikaci upozornění jsme dále vytvořili novou html stránku s přihlášením a možností přidat či odebrat upozornění. Přihlášení může proběhnout buď přes email a heslo, nebo můžeme využít přihlášení přes Google účet. Po přihlášení se objeví možnosti přidat nadpis a samotnou zprávu. Pod tímto oknem máme zobrazené veškerá upozornění, co byla vytvořena s možností odstranění. Přidání upozornění probíhá pomocí metody `sendNotification()`, ta je vyvolána tlačítkem odeslat upozornění. V metodě získáme zprávu, nadpis, jméno uživatele, který upozornění odesílá a datum s časem. Poté vytvoříme referenci do databáze a využitím předdefinované metody `push()` odešleme všechna data do databáze.

Odstranění proběhne po interakci s tlačítkem odstranit. Tím je vyvolána metoda `deleteNotification()`. Parametrem této metody je ID zprávy, v našem případě `clicked_id`, abychom věděli, které upozornění má být odstraněno. Samotné odstranění probíhá podobně jako vytvoření. Vytvoříme referenci do databáze s adresou `"/0notifications/" + clicked_id` a poté je zavolána metoda `remove()`, ta zajistí samotné odstranění vybraného upozornění z databáze. Po odstranění se naše webová aplikace aktualizuje společně s výpisem upozornění.

Po dokončení naší webové aplikace a její otestování jsme mohli přejít na konečné nasazení. K tomuto účelu jsme využili avizovaný Firebase Hosting, který nám pomůže zpřístupnit všem naši aplikaci na Internetu. Postup pro vytvoření hostingu byl velice jednoduchý v několika málo krocích. [26] Nyní nám chybělo pouze nasadit naši webovou aplikaci. K tomuto účelu vytvořili Firebase CLI, který funguje jako rozhraní příkazového řádku. Tento program vyžaduje také nainstalovaný Node.js a npm jako správce nainstalovatelných balíčků. Po spuštění CLI jsme příkazem `npm install -g firebase-tools` stáhli poslední verzi naší aplikace. Po tomto kroku jsme se mohli přihlásit k našemu účtu na Firebase. Poté jsme zvolili náš aktuální projekt. V tomto bodě jsme museli inicializovat náš projekt s nastavením cesty k naší webové aplikaci. S dokončením si Firebase automaticky vytvořil dva soubory, první `firebase.json`, který uchovává konfiguraci projektu a `.firebaserc`, kde je uložen takzvaný alias projektu.

Nyní už zbývalo jen naši webovou aplikaci pomocí příkazu firebase deploy nasadit do provozu. Po krátké chvíli jsme se mohli podívat na výsledek naší práce, viz obrázek číslo 10. V prohlížeči na vygenerované adrese pro náš projekt: <https://dht11-25f6d.firebaseio.com>



**Obrázek 10** Výsledná webová aplikace, zdroj: vlastní

### 8.2.5 Programování Android aplikace

Reálná potřeba Android aplikace coby informační systému města či obce, může lidem pomoci získat potřebné informace o dění ve městě.

Naše aplikace by měla informovat obyvatele o teplotě a vlhkosti ve městě, případně přijmout a upozornit na zprávu odeslanou obyvatelům města.

Pro programování mobilní aplikace použito Android Studio. Jedná se o vývojové prostředí pro návrh a samotné programování aplikací pro mobilní a operační systém Android. Jeho použití předcházelo jeho instalace na lokální počítač. Důležitým krokem bylo napojení aplikace k našemu Firebase projektu. Pomocí webového managementu jsme zvolili přidání Android aplikace, po otevření dialogového okna jsme zadali název balíčku, v tomto případě `com.example.arduino_control`. V dalším kroku jsme do našeho balíčku stáhli `google-services.json`, který obsahuje informace o našem Firebase projektu a zprostředkovává veškeré potřebné moduly (pluginy).

Do projektového `build.gradle` jsme doplnili takzvanou závislost (dependencies), pro využití nainstalovaného modulu

```
classpath 'com.google.gms:google-services:4.2.0'
```

Poté jsme do aplikačního `build.gradle` naimplementovali jádro

```
implementation 'com.google.firebase:firebase-core:17.0.0',
```

a databázi

```
implementation 'com.google.firebase:firebase-database:16.0.4'
```

Na úplný konec jsme aplikovali modul

```
apply plugin: 'com.google.gms.google-services'
```

Po synchronizaci jsme zapnutím projektu ověřili funkčnost a tím bylo přidání aplikace hotové a my mohli přejít na samotné programování.



Android studio nabízí jednoduchou možnost vytvoření layoutu na zvolený mobil a verzi mobilního Androidu. Aplikace také nabízí možnost vytvářet aplikace na tablety, hodinky anebo třeba na televizi. Jak můžeme vidět na obrázku číslo 11, je důležité mít rozmyšleno, jak naše aplikace má vypadat. Velmi důležité je samotné rozmístění komponent po virtuální obrazovce mobilu tak, aby se nám veškeré údaje příjemně četli, nebyli do sebe vnořené případně mimo obrazovku. Celý tento pohled jsme uložili do souboru pod názvem `aktivity_main.xml`.



**Obrázek 11** Návrh aplikace a rozmístění komponent, zdroj: vlastní

Velice důležitým krokem bylo také přiřazení ID komponentů, abychom je mohli později přidat pohled k proměnným a příkazem `setText()` nahrát data k zobrazení. Aby naše aplikace mohla fungovat, bylo zapotřebí vytvořit třídu `FireApp`, která je rozšířena o třídu `Application`. Máme zde jedinou metodu `onCreate()` ve které bylo zapotřebí inicializovat knihovnu Firebase v kontextu s Androidem, před vytvořením nebo použitím jakékoliv reference na Firebase.

Poté jsme si otevřeli třídu `MainActivity`, zde jsme vytvořili několik proměnných, pro teplotu, vlhkost, poslední aktualizaci, pro referenci na databázi a cestu. Podobně jako v předchozí třídě, jsme zde měli předvytvořenou metodu `onCreate()`. V té bylo nutné nastavit podobu aplikace kterou jsme vytvořily příkazem `setContentView(R.layout.activity_main);`.

Poté jsme přiřadili k proměnným pohledy komponentů pomocí příkazu `findViewById`. Tím jsme zajistili že až budeme mít výsledné hodnoty a uložíme je do proměnné, zobrazí se ve výsledném pohledu.

Vytvořili instance tříd `Function` a `Firestore`, kde jsme nastavili referenci do naší databáze. Jelikož známe pouze URL, bylo nutné získat další část cesty. Seřadili jsme si veškeré hodnoty dle klíče, získali poslední hodnotu a přidali listener. Vygenerovala se nám override metoda `onDataChange()`, která zajistí změnu cesty pokaždé když se změní. Zde jsme získali takzvaný `DataSnapshot`, kde je uložena relativní cesta ve tvaru `DHT11_yyyy_mm_dd` obdobně jako u webové aplikace. Tuto část cesty jsme si uložili do naší proměnné a mohli využít naší instanci na třídu `Function`, kam jsme odeslali naše nadefinované proměnné a zavolali metodu `setTemp()`.

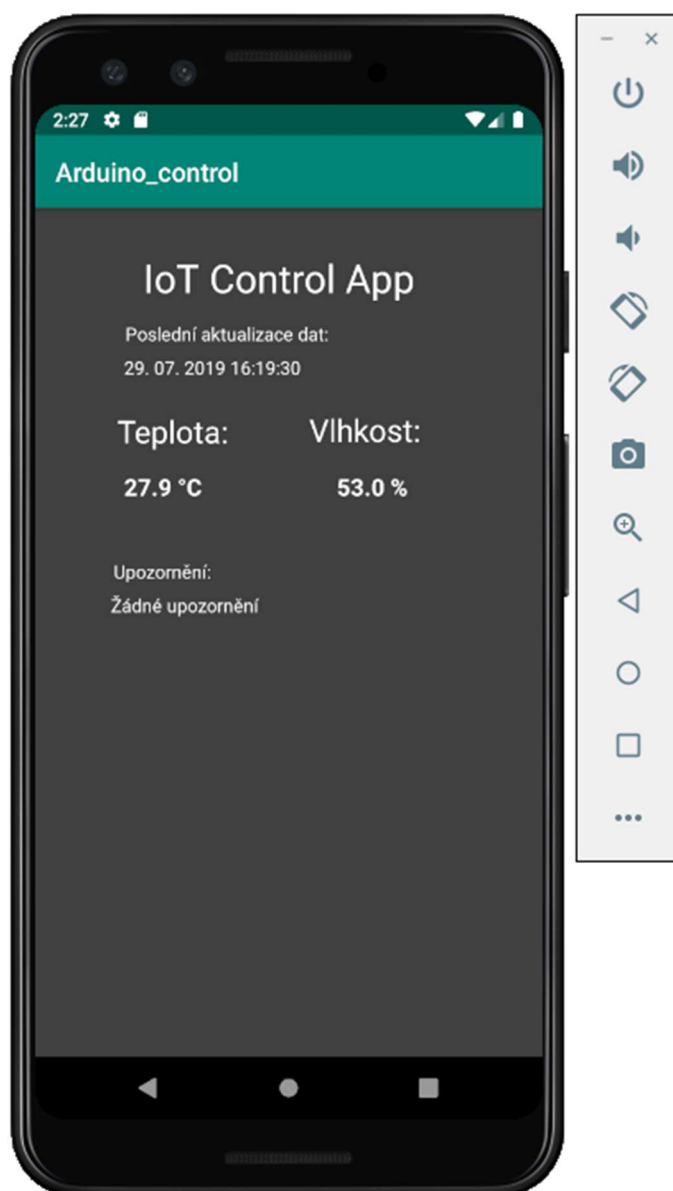
Zde jsme vytvořili novou referenci na naší databázi, kdy k URL adrese byla přidána získaná část cesty. Znovu jsme vytvořili listener na referenci a nechali vytvořit metodu `onDataChange`, kde jsme získali další `DataSnapshot`, který tentokrát obsahoval více hodnot, proto jsme vytvořili for cyklus a prošli veškeré záznamy. V prvním průchodu jsme našli `Humidity`, odkud jsme vytvořili novou referenci s listenerem, vytvořili nový `onDataChange` a s tím i nový `DataSnapshot` tentokrát již s jednou poslední hodnotou. Tím jsme získali vlhkost ve tvaru `{hh:mm:ss=vlhkost}`, zde nám opět pomohl `substring` a získali vlhkost. Nyní už jen stačilo uložit hodnotu pomocí `setText` do naší proměnné a tím zajistit zobrazení v našem návrhu aplikace. Stejný postup následoval při druhém průchodu pro `Temperature`, s rozdílem že po nastavení teploty jsme ještě nastavili datum a čas poslední aktualizace dat.

Aby naše aplikace plnila funkci informačního systému, museli jsme nastavit zobrazení upozornění. Podobně jako u webové aplikace, jsme museli zprávy získat z naší databáze. Proto byla vytvořena ve třídě `Function` metoda `getNotifications()`. Zde jsme zadali referenci databáze a takzvaný `child` na `Notifications`, kde jsou uloženy veškerá upozornění. Poté jsme vytvořili `addChildEventListener`, ve kterém se nám vygenerovali `override` metody. Nás zajímala hlavně metoda `onChildAdded()`, díky které získáme všechny záznamy. Nyní stačilo je všechna upozornění vypsat.

Do objektu `Map map` jsme uložili aktuální oznámení. Po vytvoření proměnné `titleAndMessage`, jsme do ní mohli uložit název a samotnou zprávu upozornění řetězcem:

```
map.get("title") + "\n" + map.get("message") + "\n \n";
```

Nyní stačilo zprávu pouze vypsat v našem `TextView`, ale abychom měli všechny upozornění, bylo potřeba použít místo metody `setText()` metodu `append()`. Pokud bychom použili první jmenovanou, bylo by zobrazeno pouze poslední upozornění. V Případě metody `append()` se nám vloží vždy za poslední záznam. Tím jsme dokončili veškeré úpravy ve třídě `Function` a stačilo zavolat naši metodu `getNotifications()` ve třídě `MainActivity`. Výslednou aplikaci můžeme vidět na obrázku číslo 12.



**Obrázek 12** Výsledná Android aplikace, zdroj: vlastní

### 8.3 *Shrnutí výsledků*

V průběhu práce byly vysvětleny základní koncepty struktur Smart city a technické možnosti datové sítě města. V praktické části bylo využito těchto znalostí, aby byl vytvořen funkční prvek, který lze snadno připojit do uvažovaného datového systému obce a efektivně ho využívat jak technickou službou města, tak samotnými obyvateli. Výstižněji jsme použili senzor ESP-8266 na měření teploty a vlhkosti v kombinaci s Wi-Fi modulem ESP-01, díky kterému se naše zařízení mohlo připojit k přístupovému bodu. K napájení bylo využito Li-Ion baterie Sony 18650. Pro připojení senzoru k Internetu bylo nutné vytvořit přístupový bod, pomocí zařízení Mikrotik RB493G s miniPCI kartou R52nM a jeho nastavením. Po připojení našeho přístupového bodu k internetu, jsme vytvořili projekt ve službě Firebase a nastavili databázi pro data ze senzoru. Poté jsme mohli přistoupit k samotnému programování našeho senzoru v rozhraní Arduino IDE. S tím souvisela i samotná úprava zařízení s využitím ESP-01 USB ESP8266. Pro zabezpečení našich dat ze senzoru, byl využit protokol TLS a data byla poslána do naší připravené databáze. Pro účely využití senzoru byla vytvořena webová a mobilní aplikace. Ty zobrazují nejaktuálnější informaci o teplotě a vlhkosti a zadané upozornění či výstrahy. Webová aplikace je navíc rozšířena o přihlášení pro administrátora a možnosti správu zobrazovaných upozornění a také o snímek IP kamery. Je nutné doplnit že, snímek je pouze ilustrační, a pouze dokresluje možnosti informačního systému pro obyvatele a technickou službu, kteří díky tomu mohou zhodnotit situaci.

## 9 Závěry a doporučení

V bakalářské práci byly představeny koncepty Smart city a jejich problematika. Dále byla rozebrána problematika síťové struktury města a napojení Smart technologií v kombinaci s databázovými a internetovými službami. To si dává za cíl zlepšit životní podmínky ve městě a zjednodušit každodenní činnosti obyvatel. V kapitole 6 bylo popsáno celkové fungování chytrého města, ve kterém senzory získávají data, ty jsou ukládány a dále zpracovávány. V kombinaci s dnešní moderní technologií jako jsou smartphony a další přenosná zařízení, lze snadno skrz mobilní či webové aplikace informovat obyvatele města, případně pomoci vyhledat jakoukoliv městskou službu.

Ale jak bylo pospáno v kapitole 7, je ještě předčasné mluvit o komplexním informačním systému a celkovém fungování Smart city, vzhledem k tomu, že většina měst je na samém počátku vytvoření Smart city, kdy ani nemají vybudovanou žádnou městskou datovou síť. Přesto dochází k určitému vývinu a většina měst nyní provádí studie a snaží se investovat do tohoto konceptu.

V praktické části jsme se tedy zaměřili na samotnou problematiku implementace senzoru s příslušným zpracováním dat a jejich reprezentací ve webové ale i mobilní aplikaci, jenž slouží jako informační systém pro obyvatele. Díky tomu jsme mohli dospět k názoru, že implementace může být v určitých aspektech náročná. Na úrovni hardwaru, kdy je potřeba vyřešit především napájení pro senzory a jejich implementaci do prostředí. Ale pak také na úrovni softwaru, kdy každá představená část konceptu vyžaduje jiné znalosti. Například můžeme mluvit o programování samotného senzoru, nebo o vytvoření mobilní aplikace. Velmi důležitý je také výběr technologií, které budou použity, protože každá nabízí své výhody a nevýhody a v neposlední řadě udržitelnost. S dnešním vývojem technologií narážíme na problémy, jako je rychlé stárnutí, kdy implementované řešení již není aktuální a v důsledku ani tak efektivní.

Přesto si myslím že budoucnost měst a jejich plynulé fungování závisí na představených konceptech a jejich implementace je v rámci udržitelnosti nevyhnutelná.

## 10 Seznam použité literatury

### Internetové zdroje:

[1] Mapping Smart cities in the EU [online]. Copyright © [cit. 06.05.2018]. Dostupné z:

[http://www.europarl.europa.eu/RegData/etudes/etudes/join/2014/507480/IPO-L-ITRE\\_ET\(2014\)507480\\_EN.pdf](http://www.europarl.europa.eu/RegData/etudes/etudes/join/2014/507480/IPO-L-ITRE_ET(2014)507480_EN.pdf)

[2] Network Infrastructure for Smart Cities [online]. Copyright © [cit. 05.06.2018]. Dostupné z: <http://quantela.com/blog/network-infrastructure-smart-cities>

[3] IBM Industries. [online]. Copyright © Copyright IBM Corp. 2018 [cit. 11.12.2018]. Dostupné z: <https://www.ibm.com/blogs/industries/little-known-story-first-iot-device/>

[4] Why it is called Internet of Things: Definition, history, disambiguation. IoT Analytics - Market insights for the Internet of Things [online]. Dostupné z: <https://iot-analytics.com/internet-of-things-definition/>

[5] A Simple Explanation Of 'The Internet Of Things'. Forbes [online]. Copyright ©2018 Forbes Media LLC. All Rights Reserved. [cit. 06.11.2018]. Dostupné z: <https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#36803b011d09>

[6] Internet of Things | NCTA — The Internet & Television Association. NCTA — The Internet & Television Association [online]. Copyright © NCTA [cit. 06.11.2018]. Dostupné z: <https://www.ncta.com/positions/internet-of-things>

[7] Understanding the relationship between IoT and Big Data - JAXenter. JAXenter - News, Articles, Code. [online]. Copyright © Shutterstock [cit. 06.11.2018]. Dostupné z: <https://jaxenter.com/relationship-between-iot-big-data-138220.html>

[8] What is Cloud Computing? - Amazon Web Services. Amazon Web Services (AWS) - Cloud Computing Services [online]. Copyright © 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. [cit. 11.12.2018]. Dostupné z: <https://aws.amazon.com/what-is-cloud-computing/>

[9] SaaS vs PaaS vs IaaS: What's The Difference and How To Choose – BMC Blogs. BMC Software – The Multi-Cloud Management Company [online]. Copyright © Copyright 2005 [cit. 11.12.2018]. Dostupné z: <https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/>

[10] Big Data and Cloud Computing: A Perfect Combination - VEXXHOST. High Performance OpenStack Cloud Computing Solutions | Vexxhost [online]. Copyright © 2005 [cit. 06.11.2018]. Dostupné z: <https://vexxhost.com/blog/big-data-cloud-computing/>

[11] Big Data and cloud computing – challenges and opportunities – Big Data Made Simple. Big Data Made Simple – One source. Many perspectives. [online]. Copyright © 2018 Crayon Data. All rights reserved. [cit. 06.11.2018]. Dostupné z: <https://bigdata-madesimple.com/big-data-and-cloud-computing-challenges-and-opportunities/>

[12] Overview of ISG [online]. Copyright ©k [cit. 11.12.2018]. Dostupné z: <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/isg/configuration/xen3s/isg-xe-3s-book/isg-overview.pdf>

[13] PoE vs. PoE+: What's the Difference?. Tech Topics Blog [online]. Copyright ©2019 Twinstate Technologies. All Rights Reserved. [cit. 03.08.2019]. Dostupné z: <https://blog.twinstate.com/poe-vs-poe-plus>

[14] Smart Cities: Whys, hows and whats [online]. Copyright © [cit. 05.06.2018]. Dostupné z: <http://quantela.com/blog/smartcities/>



[15] Smart city technology: It's all about the Internet of Things. Information Age - For CTOs and Technology Leaders [online]. Copyright © Bonhill Group Plc, [cit. 06.11.2018]. Dostupné z: <https://www.information-age.com/smart-city-technology-123473905/>

[16] How IoT and big data are driving smart traffic management and smart cities – Big Data Made Simple. Big Data Made Simple – One source. Many perspectives. [online]. Copyright © 2018 Crayon Data. All rights reserved. [cit. 06.11.2018]. Dostupné z: <https://bigdata-madesimple.com/iot-big-data-driving-smart-traffic-management-smart-cities/>

[17] They are replacing street lights with new LED lamps. It is supposed to reduce the electricity bill and the light pollution. I don't know about the pollution, but the street looks better. : pics. reddit: the front page of the internet [online]. Copyright © 2018 Reddit, Inc. All rights reserved [cit. 12.01.2019]. Dostupné z: [https://www.reddit.com/r/pics/comments/2irgeu/they\\_are\\_replacing\\_street\\_lights\\_with\\_new\\_led/](https://www.reddit.com/r/pics/comments/2irgeu/they_are_replacing_street_lights_with_new_led/)

[18] Barcelona: smart city revolution in progress | Financial Times. Financial Times [online]. Copyright © [cit. 12.01.2019]. Dostupné z: <https://www.ft.com/content/6d2fe2a8-722c-11e7-93ff-99f383b09ff9>

[19] SMART CITY: Cesta za lepším životem ve městě.. 403 Forbidden [online]. Dostupné z: <https://service.ihned.cz/smartycity/>

[20] IoT Plzeň. IoT Plzeň [online]. Copyright © 2017 [cit. 11.12.2018]. Dostupné z: <https://iot.plzen.eu>

[21] Lithium Ion Battery | Clean Energy Institute. Clean Energy Institute | University of Washington [online]. Copyright © 2019 [cit. 28.07.2019]. Dostupné z: <https://www.cei.washington.edu/education/science-of-solar/battery-technology/>

[22] Lithium-ion Vs. Nickel Metal Hydride Batteries | HowStuffWorks. Auto | HowStuffWorks [online]. Copyright © 2019 [cit. 28.07.2019]. Dostupné z: <https://auto.howstuffworks.com/lithium-ion-batteries-improve-hybrids1.htm>

[23] Firebase. Firebase [online]. Dostupné z: <https://firebase.google.com>

[24] Top 5 Ways to Wire up ESP8266 Module to your Arduino Uno Board – Online Shouter. Online Shouter – Premier website for SEO, WordPress and Blogging [online]. Dostupné z: <http://onlineshouter.com/top-5-ways-wire-esp8266-module-arduino-uno-board/>

[25] firebase | JavaScript SDK | Firebase. Firebase [online]. Dostupné z: <https://firebase.google.com/docs/reference/js/firebase>

[26] Get started with Firebase Hosting | Firebase. Firebase [online]. Dostupné z: <https://firebase.google.com/docs/hosting/quickstar>

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Tomíček Václav	Paseky nad Jizerou 98, Paseky nad Jizerou	I1600622

**TÉMA ČESKY:**

Implementace SMART technologií do počítačové sítě obce

**TÉMA ANGLICKY:**

Implementation of SMART technologies into the city's computer network

**VEDOUcí PRÁCE:**

Ing. Pavel Blažek, Ph.D. - KIT

**ZÁSADY PRO VYPRACOVÁNÍ:**

Cílem bakalářské práce je seznámit se se základními infrastrukturními prvky města a navrhnout do nich zasazení SMART technologií s využitím počítačové sítě obce.


- 1) Chytré město
- 2) Síťová struktura chytrého města
- 3) Fungování chytrého města
- 4) Současný stav
- 5) Napojení Smart technologií do současné sítě
- 6) Závěr

**SEZNAM DOPORUČENÉ LITERATURY:**

- 1, Města budoucnosti; Miroslav Svítek, Michal Postránecký; ISBN 978-80-72700-58-5
- 2, Smart city v praxi; Jakub Slavík; ISBN: 978-80-86726-80-9
- 3, Smart Cities: magazín o chytrých technologiích pro efektivnější správu měst a obcí ISSN: 2336-1786
- 4, Technická infrastruktura měst a sídel, Zuzana Vyoralová, Petr Hrdlička; ISBN: 978-80-01-05202-0
- 5, Smart cities and homes: key enabling technologies; Cambridge, MA : Morgan Kaufmann, (2016). ISBN: 978-0-12-803454-5

Podpis studenta:  .....

Datum: 11.10.2018

Podpis vedoucího práce:  .....

Datum: 11.10.2018