



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

LEVNÁ STANICE PRO CHYTROU
DOMÁCNOST

ECONOMY STATION FOR SMART HOME

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Sebastián Mach

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Michal Matějáško

BRNO 2018

Zadání bakalářské práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky
Student: **Sebastián Mach**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Mechatronika
Vedoucí práce: **Ing. Michal Matějáško**
Akademický rok: 2017/18

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Levná stanice pro chytrou domácnost

Stručná charakteristika problematiky úkolu:

Práce se bude zabývat možnostmi, které skýtají dnes dostupné nástroje pro zařízení spadajících do kategorií Internetu věcí, Domácí automatizace a případně "Velkých dat". Na příkladu levné domácí stanice některé dostupné možnosti demonstruje.

Cíle bakalářské práce:

1. Seznamte se s počítačem Raspberry Pi 3 a s platformou ThingSpeak.
2. Vyberte vhodné součásti a sestavte domácí stanici tak, aby alespoň monitorovala základní meteorologické veličiny. Funkcionalitu rozšiřte dle uvážení.
3. Vytvořte obslužný software zajišťující požadovanou funkcionalitu zařízení. Ověřte jeho funkčnost v prostorách Mechatlabu.
4. Propojte domácí stanici s ThingSpeak. Využijte ji pro záznam a analýzu měřených dat.
5. Analyzujte měřená data a vhodným způsobem informujte uživatele.

Seznam doporučené literatury:

VALÁŠEK, M.: Mechatronika, Vydavatelství ČVUT, 1995

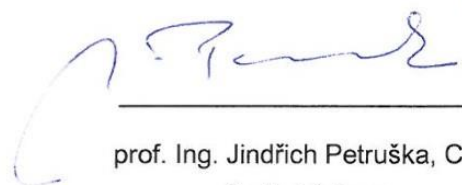
HEROUT, Pavel. Učebnice jazyka C. 5. vyd. České Budějovice: Kopp nakladatelství, 2008. ISBN 978-80-7232-351-7.

SUMMERFIELD, Mark. Python 3: výukový kurz. Brno: Computer Press, 2010. ISBN 978-80-251-2737-7.

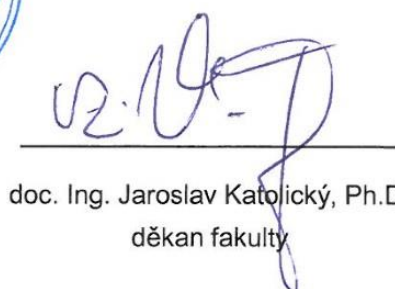
FEYNMAN, Richard Phillips, Robert B. LEIGHTON a Matthew SANDS. The Feynman lectures on physics. 6th print. Reading, Massachusetts: Addison-Wesley Publishing Company, 1977. ISBN 0-201-02116-1. <http://www.feynmanlectures.caltech.edu/>, California Institute of Technology

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2017/18.

V Brně, dne 27. 10. 2017



prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu



doc. Ing. Jaroslav Katoický, Ph.D.
děkan fakulty

ABSTRAKT

Táto práca sa zaoberá vytvorením stanice pre inteligentnú domácnosť, ktorá bude ovládať jej externé periférie. Stanica bude komunikovať s kamerou a senzormi, sťahovať z internetu potrebné informácie a komunikovať s cloudom, na ktorom prebiehajú výpočty.

ABSTRACT

This project is about creating a smart home station that will control its external peripherals. The station will communicate with the camera and sensors, download the necessary data from internet, and communicate with the cloud, where are the calculations made.

KLÍČOVÁ SLOVA

Chytrá domácnosť, Raspberry, ThingSpeak, Node.js, React.js, Python OpenCV, MATLAB

KEYWORDS

Smart home, Raspberry, ThingSpeak, Node.js, React.js, Python OpenCV, MATLAB

BIBLIOGRAFICKÁ CITACE

MACH, S. *Levná stanice pro chytrou domácnost*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 43 s. Vedoucí bakalářské práce Ing. Michal Matějáško.

POĎAKOVANIE

Rád by som poďakoval svojmu vedúcemu práce Ing. Michalovi Matějáskovi, za jeho vedenie, konzultácie a cenné rady pri tvorbe tejto práce.

ČESTNÉ PREHLÁSENIE

Prehlasujem, že táto práca je mojím pôvodným dielom, spracoval som ju samostatne pod vedením Ing. Michala Matějáska a s použitím literatúry uvedenej v zozname literatúry.

V Brne dňa 23.5. 2018

.....

Sebastián Mach

OBSAH

1	ÚVOD.....	6
2	CIEĽ PRÁCE	7
3	ZÁKLADNÉ INFORMÁCIE	8
3.1	RASPBERRY A OPERAČNÝ SYSTÉM.....	8
3.2	THINGSPEAK	9
3.3	PROGRAMY PRE OBSLUHU RASPBERRY	10
4	SPUSTENIE SERVERA	12
5	SPUSTENIE GUI.....	14
6	METEOSTANICA.....	16
6.1	INFORMÁCIE O SENZORE	16
6.2	ZAPOJENIE SENZORA.....	17
6.3	PRIPOJENIE NA THINGSPEAK	17
6.4	PRINCÍP KOMUNIKÁCIE THINGSPEAK – SERVER.....	18
6.5	ANALÝZA DÁT NA THINGSPEAKU A PREDPOVEĎ POČASIA	19
6.6	PREDPOVEĎ Z OFICIÁLNEJ WEBOVEJ APLIKÁCIE.....	22
7	ZVONČEK S BEZPEČNOSTNOU KAMEROU	24
7.1	STREAM VIDEO A ZACHYTENIE FOTOGRAFIE	24
7.2	TLAČIDLO	26
7.3	NÁVRH KRABÍČKY ZVONČEKA	28
7.4	ROZPOZNÁVANIE TVÁRE S PYTHON OPENCV.....	29
7.5	SQLite DATABÁZA.....	31
7.6	PRINCÍP FUNGOVANIA ZVONČEKA	32
7.6.1	STLAČENIE TLAČIDLA	32
7.6.2	PODRŽANIE TLAČIDLA.....	33
7.6.3	PRIDANIE NOVÉHO UŽÍVATEĽA DO DATABÁZY	34
8	API DOKUMENTÁCIA	35
9	UKÁŽKY Z UŽÍVATEĽSKÉHO ROZHRAŇIA	39
10	ZÁVER	41
11	ZOZNAM POUŽITEJ LITERATÚRY	42
12	ZOZNAM OBRÁZKOV A TABULIEK	43

1 ÚVOD

Svet technológií pre domácnosť už nie je budúcnosť, ale prítomnosť. Definícia inteligentnej domácnosti spočíva v tom, že využíva zariadenia pripojené na internet, čo umožňuje diaľkové ovládanie a správu domácnosti. Takáto domácnosť by mala poskytovať komfort, bezpečnosť, šetrenie energie a pomoc pri chode domácnosti.

Vyšší stupeň inteligentnej domácnosti je autonómnosť. To znamená, že systém koná na základe údajov z rôznych senzorov, užívateľských nastavení a výpočtov umelej inteligencie. Takéto systémy ale nie sú lacná záležitosť a vybudovanie takéhoto komplexného systému vyžaduje mnoho prostriedkov a znalosť technológií. Výhodou niektorých inteligentných domácností je modulárny systém, čo znamená, že jednotlivé zariadenia sa dajú dopĺňať postupne. Na tomto princípe je postavená aj táto práca.

Koncept inteligentnej domácnosti tejto práce je postavený na cenovo dostupnom mikropočítači Raspberry Pi, ktorý predstavuje jadro domácnosti, s ktorým spolupracujú ostatné zariadenia. Na tento počítač je možné napojiť rôzne prvky predstavujúce zariadenia inteligentného systému. V nasledujúcich kapitolách bude podrobne opísaný vývoj dvoch zariadení – domového zvončeka s bezpečnostnou kamerou s rozpoznávaním tváre a senzora na meranie základných meteorologických údajov ako teplota, vlhkosť a tlak vzduchu.

2 CIEĽ PRÁCE

Cieľom tejto práce je návrh stanice pre inteligentnú domácnosť na mikropočítači Raspberry Pi s hardvérovými prvkami, a to konkrétne domový zvonček s kamerou a senzor na meranie meteorologických údajov. Medzi ďalšie hlavné ciele tejto práce patrí:

- prepojenie stanice s platformou ThingSpeak,
- záznam a analýza nameraných dát zo senzora,
- vytvorenie užívateľského rozhrania pre kompletné automatické ovládanie stanice.

Medzi vedľajšie ciele, ktoré bolo treba splniť patrí:

- vytvorenie softvéru pre rozpoznanie tváre návštevníka z kamery na zvončeku,
- návrh systému pre otváranie dverí (databáza tvár užívateľov, pridávanie užívateľov, štart analýzy tváre),
- návrh dizajnu a 3D tlač zvončeka.

3 ZÁKLADNÉ INFORMÁCIE

3.1 RASPBERRY A OPERAČNÝ SYSTÉM

Raspberry je malý jednodoskový počítač s doskou plošných spojov. V roku 2012 bol vyvinutý primárne pre výučbu informatiky na školách. Tento počítač obsahuje všetky potrebné porty a periférie ako HDMI, USB, GPIO piny a ďalšie, čím je vhodným kandidátom pre stanicu inteligentnej domácnosti. Na Raspberry je možné spustiť rôzne distribúcie Linuxu ale aj Microsoft Windows 10 IoT Core [1].



Obr.1: Raspberry Pi 3 Model B.

Architektúra	ARMv8-A (64/32-bit)
Procesor (CPU)	1.4 GHz 64-bit quad-core ARM Cortex-A53
Pamäť (SDRAM)	1 GB
Integrovaná sieť	Gigabitový Ethernet + WiFi 802.11ac, Bluetooth 4.2 BLE

Tabuľka 1: Základné parametre Raspberry Pi 3 Model B.

Táto stanica je navrhnutá na model Raspberry Pi 3 Model B. Ako operačný systém bol zvolený *RASPBIAN STRETCH LITE*, ktorý je založený na linuxovej distribúcii Debian. Tento systém je oproti klasickému desktopovému *Raspbianu* odľahčený o grafické rozhranie a jeho veľkosť je oveľa menšia. To znamená, že stanica bola pri vývoji ovládaná cez terminálové príkazy.

Zvolený operačný systém má v sebe predinštalované programy ako Python 2, Scratch, Javu a ďalšie ovládače potrebné pre správny chod.

Postup pri inštalácii OS Raspbian Stretch Lite :

1. stiahnuť inštalačný obraz Raspbianu z oficiálnej stránky [2],
2. tento obraz zapísať na sformátovanú Micro SD kartu a vložiť do Raspberry,
3. na Micro SD karte bolo treba vytvoriť prázdny textový súbor s názvom “ssh” pre umožnenie komunikácie cez SSH.

Secure Shell (SSH) je protokol, ktorý umožňuje zabezpečenú vzdialenú komunikáciu medzi počítačmi pripojenými na jednu sieť. Tento protokol sa používa pre prístup k terminálu *Raspbianu* a takisto pre prenos súborov z osobného počítača na *Raspberry Pi*.

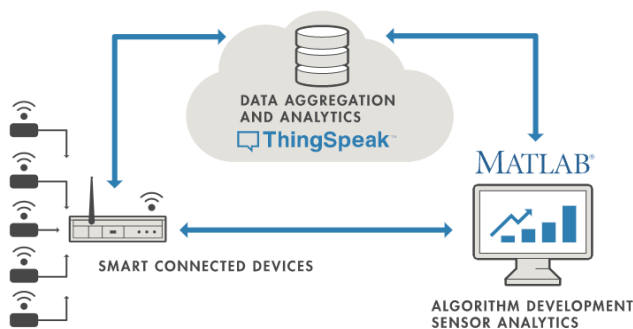
3.2 THINGSPEAK

ThingSpeak je IoT platforma, ktorá umožňuje ukladanie, vizualizáciu a analýzu dát v reálnom čase. Všetky tieto procesy prebiehajú na cloude ThingSpeak. Na cloude má funkciu spúšťať kód napísaný v MATLAB, ktorý môže vykonávať analýzu a kolekciu dát.

IoT – skratka pre internet vecí. Znamená to, že na internet je pripojených veľa zariadení ako sú napríklad autá, spotrebiče alebo senzory. Tieto zariadenia zberajú údaje a ukladajú ich na cloud ako v tomto prípade ThingSpeak. Tu prebieha analýza dát a následne sa na základe jej výsledku spúšťajú procesy v infraštruktúre zariadení. IoT sa dnes používa vo veľa aplikáciách ako sú napríklad domáca automatizácia, monitorovanie dopravy alebo počasia.

Medzi hlavné výhody platformy ThingSpeak patrí:

- jednoduché nastavenie zariadení na odosielanie dát na ThingSpeak kanál,
- vizualizácia dát v reálnom čase,
- MATLAB pre analýzu dát,
- automatizácia analýzy v pravidelnom časovom intervale alebo na základe splnených potrebných podmienok,
- vlastná RESTful API,
- odpadá nutnosť vlastného servera pre tieto činnosti, čím je inteligentná stanica odľahčená o tieto úlohy.



Obr. 2: Blokový diagram ThingSpeak [3].

3.3 PROGRAMY PRE OBSLUHU RASPBERRY

Ďalším krokom po inštalácii operačného systému a spustení SSH komunikácie bolo pripojenie Raspberry na internet a inštalácia programov potrebných na vývoj stanice. Raspberry Pi sa na WiFi pripája nasledovne:

- príkaz pre nájdenie všetkých dostupných sietí:

```
$ sudo iwlist wlan0 scan
```

- príkaz pre úpravu súboru s nastaveniami siete:

```
$ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

- v tomto súbore treba pridať položku podľa nasledujúceho vzoru:

```
network={
    ssid="meno siete"
    psk="heslo"
}
```

- po úprave a uložení súboru treba Raspberry reštartovať. Po reštarte sa automaticky pripojí k WiFi sieti ak je v jej dosahu

```
$ sudo reboot
```

Zoznam programov, ktoré boli použité pri vývoji stanice:

MobaXterm – nástroj pre vzdialenú správu počítačov. Poskytuje všetky dôležité sieťové nástroje ako SSH, X11, RDP a Unixové príkazy (bash, ls, cat, sed) v jednom programe. Cez tento program bol ovládaný terminál *Raspbianu* a zobrazovali sa v ňom výstupné správy zo servera [4].

WinSCP – open-source nástroj pre Windows pre bezpečné prenášanie súborov medzi počítačom a vzdialeným serverom. Je ním možné sa pripojiť k SSH serveru so službou SFTP (SSH File Transfer Protocol). Tento program slúžil na prenášanie skriptov a zdrojových kódov do internej pamäte Raspberry Pi [5].

Advanced IP Scanner – sieťový skener, ktorý zobrazuje všetky pripojené sieťové zariadenia. Tento program bol využívaný pre identifikáciu IPv4 adresy Raspberry Pi.

4 SPUSTENIE SERVERA

Po pripojení stanice na internet a inštalácii všetkých potrebných programov na obsluhu Raspberry nasledoval výber systému, na ktorom bude bežať server stanice (backendová časť). Ako najvhodnejší bol zvolený Node.js [6].

Je to softvérový systém určený pre škálovateľné internetové aplikácie a v oblasti IoT je jeden z najpoužívanejších. Jeho primárnym cieľom je tvorba serverovej časti webových aplikácií.

Node.js sa skladá z V8 JavaScript enginu, na ktorom beží napríklad Chrome od spoločnosti Google. Node.js využíva model I/O s neblokujúcim prístupom k udalostiam. To znamená, že úlohy sa v ňom spracúvajú asynchrónne. Node.js je postavený na tzv. event modele. Existuje event loop, slučka udalostí, ktorá sa zapne pri spustení skriptu a vypne po vykonaní všetkých inštrukcií. Do tejto slučky prichádzajú požiadavky od užívateľa na vykonanie, ktoré sú pridelené jednotlivým nezávislým vláknam, a vystupujú v poradí v závislosti od toho, ako rýchlo boli vykonané. Toto zabezpečuje asynchrónnosť a neblokujúce vykonávanie úloh [6].

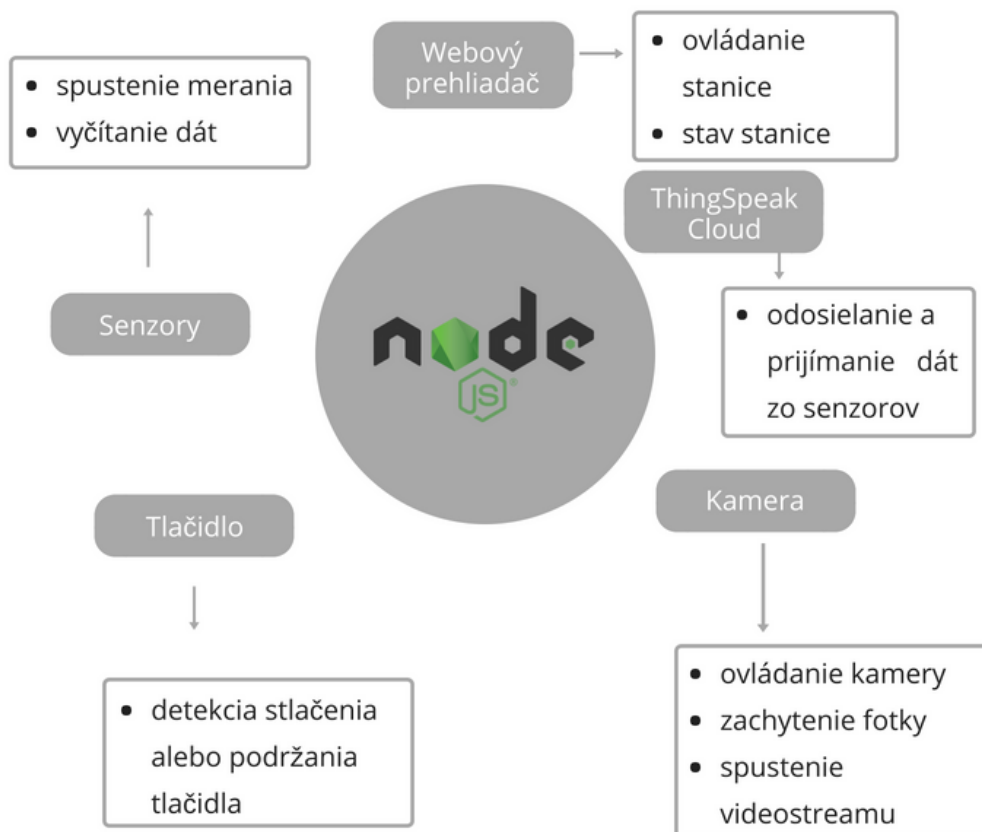
Ďalšou výhodou, prečo bol zvolený práve tento systém je, že obsahuje ekosystém balíčkov npm. Sú to open-source knižnice, ktoré sa do projektu pridávajú jednoduchým príkazom „*npm install [názov knižnice]*“.

Node.js sa dá použiť aj k spúšťaniu procesov cez terminál. Predinštalovaná knižnica s názvom *spawn* dokáže spustiť celú radu skriptov a komunikovať medzi nimi.

Na tomto princípe bude postavená celá inteligentná stanica. Na Raspberry bude bežať server v Node.js, ktorý bude:

- ovládať všetky periférie stanice – kameru (zachytenie fotky alebo spustenie streamu), senzory (vyčítavanie hodnoty z pinov),
- komunikovať s užívateľským rozhraním, ktoré bude bežať vo webovom prehliadači. Cez toto užívateľské rozhranie bude komunikovať užívateľ s Node.js, ktorý bude vykonávať zadané úlohy,
- odosielanie dát zo senzorov na cloud ThingSpeak a prijímanie dát z analýzy predpovede počasia,
- detekovať interakciu s tlačidlom, na základe ktorej bude vykonávať ďalšie úlohy.

Bloková schéma serverových úloh

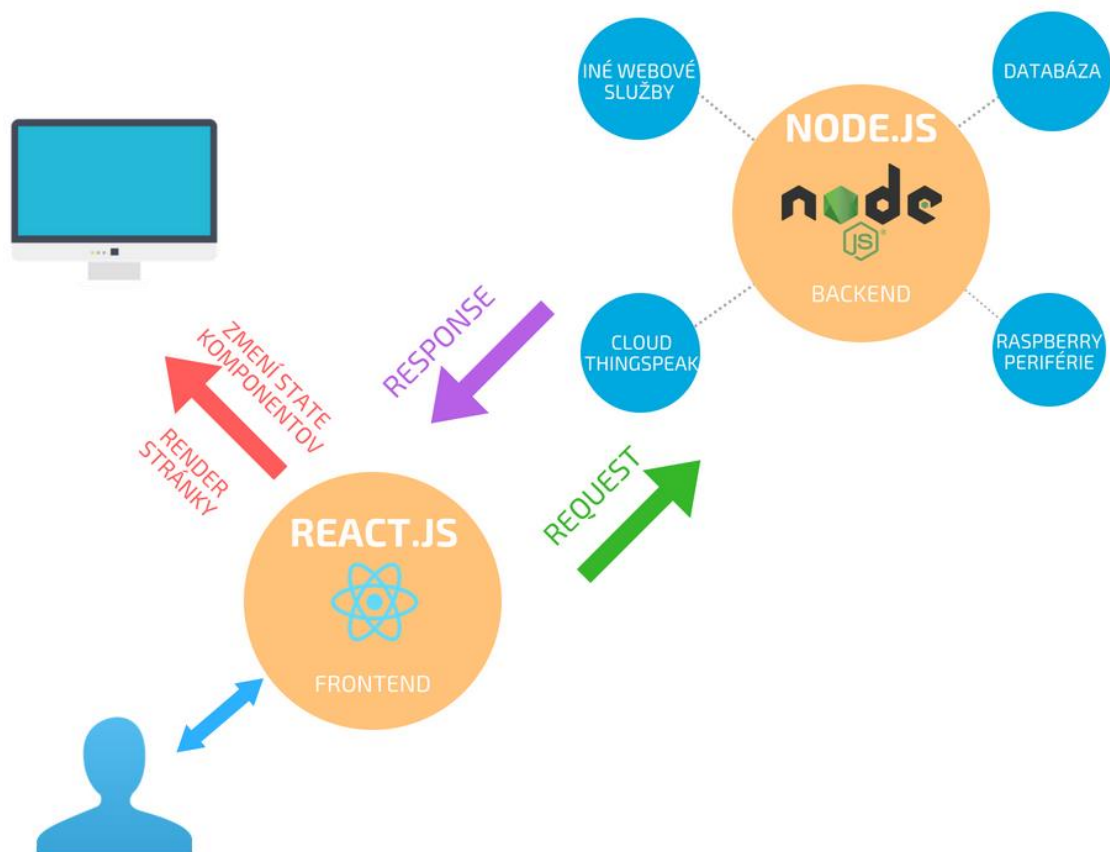


Obr. 3: Bloková schéma serveru.

5 SPUSTENIE GUI

Grafické rozhranie bolo vyvíjané v React.js [7]. React.js je JavaScript knižnica pre tvorbu interaktívnych užívateľských rozhraní. Je založený na princípe skladania komponentov na stránku v závislosti na stave aplikácie. Každý komponent si manažuje svoj vlastný stav. Tieto komponenty sú napísané v jazyku JavaScript, sú zložené z HTML elementov, ktoré sú štýlované pomocou CSS tak, aby spolu vytvárali komplexné užívateľské rozhranie. UI bolo vyvíjané tak, aby bolo zhodné vo všetkých moderných webových prehliadačoch a responzívne pre mobily a tablety.

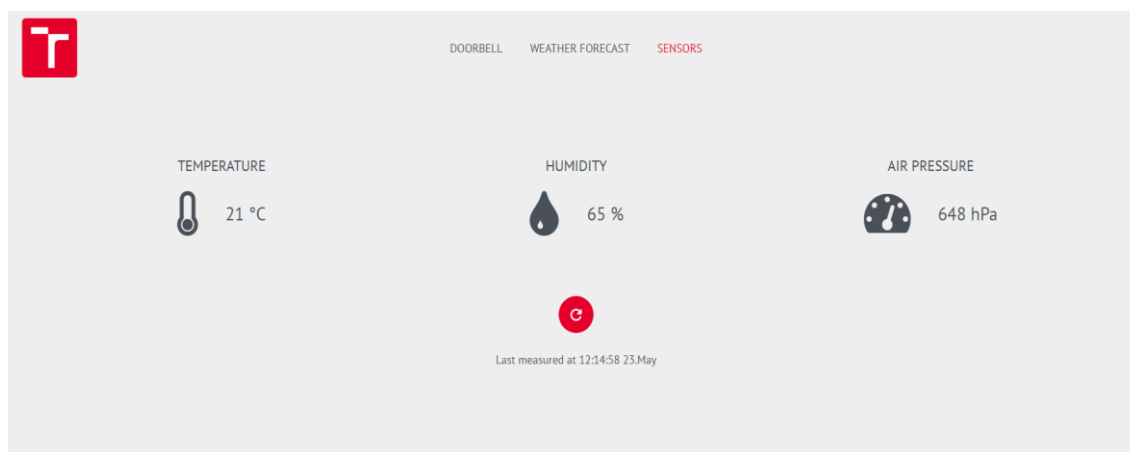
Model komunikácie užívateľského prostredia a serveru, ktorý ovláda stanicu je zobrazený v nasledujúcej blokovej schéme.



Obr. 4: Blokova schéma UI + server.

UI bude obsahovať 3 hlavné sekcie:

- a) zvonček – v tomto komponente bude možné spustiť video stream z kamery, zachytiť aktuálnu fotku z kamery a spravovať systém rozpoznávania. Tento systém obsahuje správu databázy uložených fotiek užívateľov, metódu uloženia nového užívateľa a náhľad na ukážku z poslednej analýzy tváre.
- b) meteostanica – tento komponent je rozdelený na dve časti. Jedna je oficiálna predpoveď z webovej služby, ktorá poskytuje údaje o aktuálnom počasí s predpoveďou na najbližšie dni. Druhou je vlastná predpoveď počasia z výpočtov z ThingSpeak analýzy.
- c) senzor – v komponente sú zobrazené aktuálne údaje namerané senzorom pripojeným na stanicu.



Obr. 5: Horná lišta UI s údajmi zo senzora BME280.

6 METEOSTANICA

V nasledujúcich podkapitolách bude opísaný vývoj prvej funkcie inteligentnej stanice – meteostanica.

6.1 INFORMÁCIE O SENZORE

Pre meranie meteorologických údajov bolo na výber z viacerých variant. Senzory sa líšia tým, čo všetko sú schopné merať, zapojením, rozsahom a presnosťou merania. Medzi vhodné senzory pre túto aplikáciu patria :

- DHT11 ,DHT12 – senzory pre meranie teploty a vlhkosti, používajú 1 GPIO pin, rozdiel medzi nimi je v presnosti a rozsahu merania,
- DHT22 senzor pre meranie vlhkosti v rozsahu od 0 až po 100 percent s presnosťou 2 %,
- DS18B20 – senzor pre meranie vonkajšej teploty vzduchu, je vode odolný, rozsah merania od -55 do 125 °C,
- BMP180 – senzor pre meranie tlaku vzduchu,
- BME280 – senzor, ktorý meria teplotu, vlhkosť aj tlak.



Pre jednoduchosť merania všetkých veličín v jednom senzore bol zvolený BME280. Medzi jeho výhody patrí cena (3 doláre) a to, že po objednaní senzor prichádza skalibrovaný.

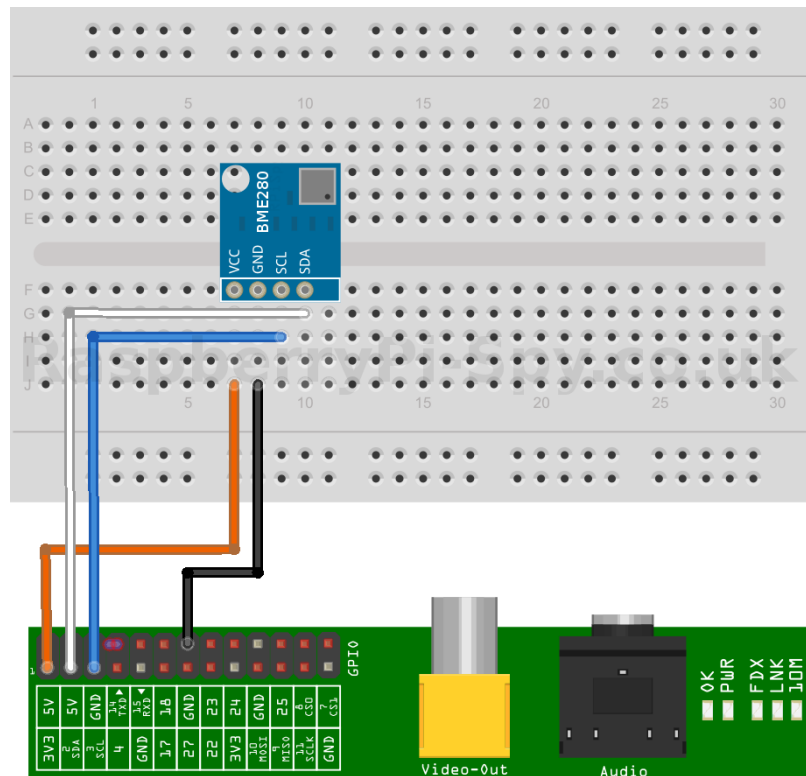
Obr. 6: BME280.

Napájanie	1,8 – 5 V DC		
Interface	I ² C (nad 3,4 MHz), SPI (nad 10 MHz)		
	Teplota	Vlhkosť	Tlak vzduchu
Rozsah merania	-40 až 85 °C	0 až 100 %	300 až 1100 hPa
Presnosť	±1 °C	±3 %	±1 Pa

Tabuľka 2: Údaje o senzore BME280 [8].

6.2 ZAPOJENIE SENZORA

Senzor má 4 piny a jeho pripojenie je znázornené na obrázku. Vyčítavanie údajov prebieha pomocou rozhrania I²C. Skript pre vyčítanie údajov je napísaný v jazyku C a skompilovaný kompilátorom GCC. Spustenie procesu merania zabezpečuje automaticky Node.js a trvá od požiadavky na údaje zo senzora, spustenia merania a odpoveďou servera 120 ms.



Obr. 7: Schéma zapojenia senzora [9].

6.3 PRIPOJENIE NA THINGSPEAK

Na začiatku sa vytvoril ThingSpeak kanál pre ukladanie nameraných dát s tromi poľami pre teplotu, vlhkosť a tlak vzduchu. Po vytvorení kanála boli vygenerované API kľúče pre zapisovanie a vyčítavanie dát. Pre komunikáciu medzi Node.js a ThingSpeak sa používa npm balíček s názvom *Thingspeakclient*. V nasledujúcom odstavci je zobrazený postup pripojenia servera na cloud.

- inicializácia údajov potrebných na prepojenie Node.js a ThingSpeak:

```
const ThingSpeakClient = require('thingspeakclient');
const client = new ThingSpeakClient();
const writeKey: 'C508KWFR1R01PEDZ';
const readKey: 'C6087XH9Q9WXBV50';
const channelId: 432138;
```

- pripojenie na kanál a po úspešnom pripojení odoslanie nameraných dát:

```
client.attachChannel(channelId, {writeKey: writeKey, readKey: readKey},
  (errAttachChannel,resAttachChannel) => {
    if(!errAttachChannel){
      client.updateChannel(channelId,
        {field1: temp,
         field2: humidity,
         field3: pressure}, (updateErr,updateRes) => {
        console.log('<ThingSpeak> Data posted to ThingSpeak
at', new Date().toLocaleString());
        });
      }
    });
```

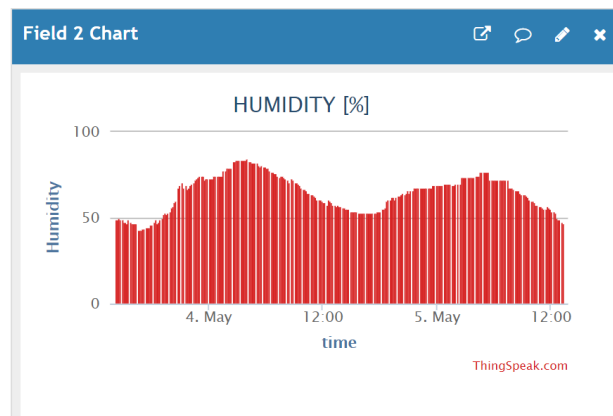
6.4 PRINCÍP KOMUNIKÁCIE THINGSPEAK – SERVER

Komunikácia medzi serverom domácej stanice a cloudom ThingSpeak je ovládaná pomocou npm knižnice *node-scheduler*. Táto knižnica sa stará o opakujúce sa vykonávanie definovanej funkcie.

Nasledujúca ukážka zo skriptu zobrazuje, ako bol nastavený časový interval 10 minút pre odosielanie dát na cloud. V tele funkcie prebieha odmeranie dát, a po úspešnom odmeraní odošle údaje na zvolený ThingSpeak kanál.

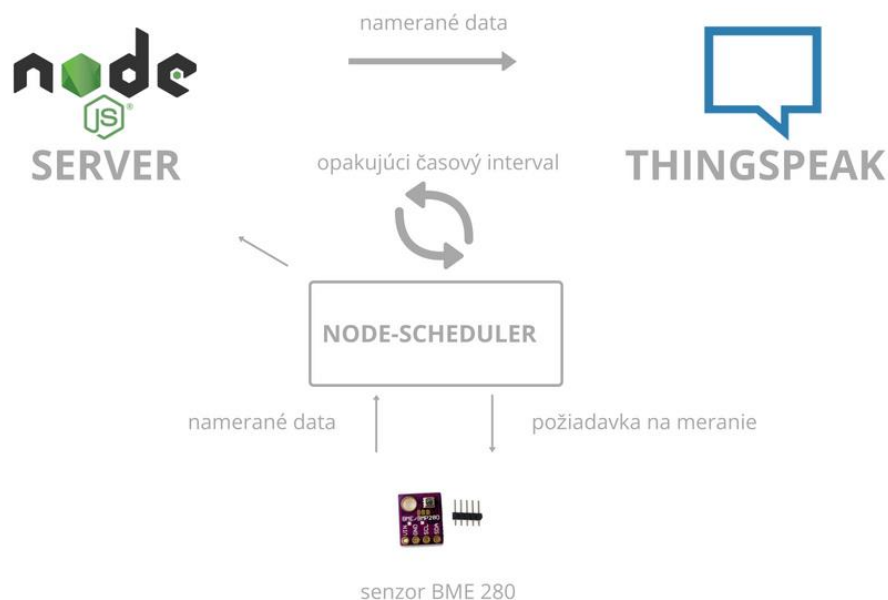
```
const scheduler = schedule.scheduleJob('*/*10 * * * *', function(){
  /*
    odmeranie a odoslanie dát zo senzora na ThingSpeak
  */
});
```

Po spustení servera bude stanica pravidelne odosielať namerané údaje na ThingSpeak, kde sa budú ukladať a vyhodnocovať. Príklad vizualizácie nameraných údajov je na ďalšom obrázku:



Obr. 8: ThingSpeak graf vlhkosti vzduchu.

Nasledujúca bloková schéma vysvetľuje princíp merania dát s ich odosielaním na ThingSpeak:



Obr.9: Bloková schéma TS-server.

6.5 ANALÝZA DÁT NA THINGSPEAKU A PREDPOVEĎ POČASIA

Ďalšou funkciou inteligentnej domácej stanice je výpočet vlastnej predpovede počasia zo základných meteorologických údajov – teplota, vlhkosť a tlak vzduchu. Táto predpoveď sa vytvára z uložených dát na ThingSpeak. ThingSpeak umožňuje jednoduchý prístup k uloženým údajom z kanála a spustenie MATLAB skriptov priamo vo webovom prehliadači. V nastaveniach je možné tento skript spúšťať v pravidelnom časovom intervale nastavením „TimeControl“.

Predpoveď počasia sa počíta z údajov za posledných 24 hodín. Základ výpočtu predpovede je postavený na priebehoch dát, aktuálnych hodnotách a súvislosti medzi teplotou, vlhkosťou a tlakom. Predpoveď je vyhodnocovaná v štyroch kategóriách:

- tlak
- rosný bod a tlak vzduchu
- teplota
- rosný bod a teplota

Apps / TimeControl / New

Name

Time Zone

Frequency ☐ One Time ☒ Recurring

Recurrence ☐ Week ☐ Day ☐ Hour ☒ Minute

Every minutes

Start Time

Fuzzy Time

Action

Code to execute

Obr. 10: Nastavenie TimeControl na platforme ThingSpeak.

Každá táto kategória obsahuje isté predpovedné statusy, z ktorých každý má určité podmienky. Ak sú tieto podmienky splnené a dáta o počasí ich spĺňajú, stáva sa tento status pravdivý a zobrazí sa v užívateľskom rozhraní jeho hláška aj s podrobným vysvetlením. Pre predpoveď boli vytvorené dva kanály. Prvý zbieral potrebné údaje pre predpoveď a druhý ukladal informácie o predpovedi. Tieto dáta sú formátované do JSON syntaxe. JSON (JavaScript Object Notation) je jednoduchý formát pre výmenu údajov. Je zložený z dvojíc názov – hodnota. Princíp ukladania predpovedných statusov na kanál je nasledovný:

- vykonajú sa funkcie, ktoré overujú pravdivosť podmienok pre statusy o počasí,
- tieto výsledky sú vstupom pre funkcie, ktoré vytvárajú JSON objekty zvlášť pre každú kategóriu,
- po tomto procese sa tieto objekty sformátujú do jedného JSON objektu a ten sa zapíše na ThingSpeak kanál.

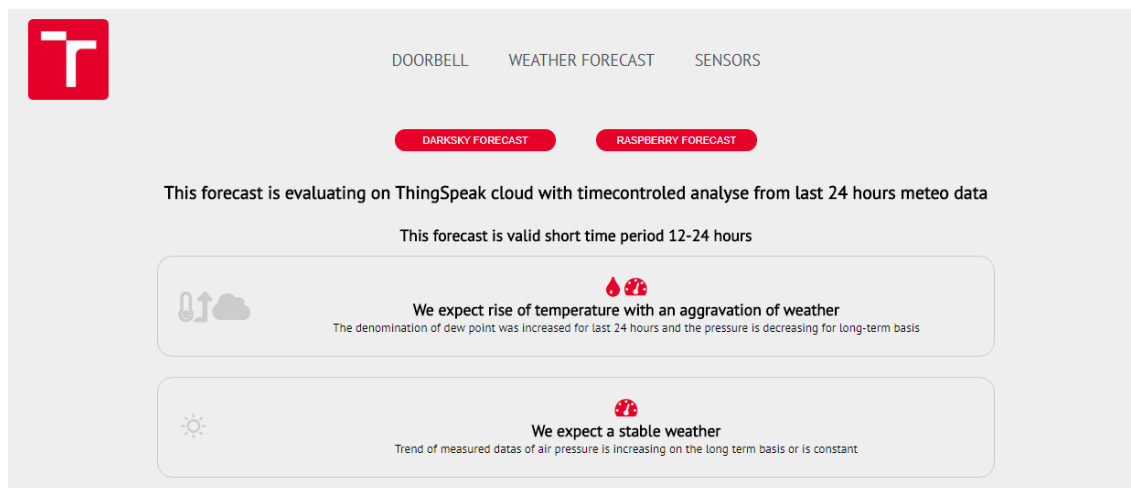
Toto členenie údajov o predpovedi bolo vytvorené pre jednoduchšie pridelovanie ikon v užívateľskom rozhraní a špeciálny ThingSpeak kanál pre JSON objekty s informáciami o predpovedi bol vytvorený pre jednoduchšie rozlíšenie vstupov zapisovania údajov o počasí od informácií o predpovedi.

V tomto odseku sú uvedené predpovedné statusy s ich podmienkami, ktoré musia byť splnené pre zobrazenie:

- 1) TLAK
 - a) Očakávame stabilné počasie – tlak vzduchu stúpa alebo je konštantný (± 3 hPa)
 - b) Očakávame nepriaznivé počasie – tlak vzduchu klesol pod 1000 hPa
 - c) Očakávame zrážky prípadne búrku – tlak vzduchu klesá alebo sa pohybuje okolo hodnoty ($1013,3 \pm 0,1$) hPa
- 2) ROSNÝ BOD A TLAK VZDUCHU
 - a) Očakávame mierne ochladenie so zlepšeným počasím – tlak vzduchu stúpa a hodnota rosného bodu klesá
 - b) Predpokladáme oteplenie – tlak vzduchu aj hodnota rosného bodu stúpa
 - c) Predpokladáme mierne ochladenie – tlak vzduchu aj hodnota rosného bodu klesá
 - d) Predpokladáme oteplenie so zhoršeným počasím – tlak vzduchu klesá a hodnota rosného bodu stúpa
- 3) TEPLOTA
 - a) Dnes bude horúci deň – teplota v ranných hodinách prekročila hranicu 20 °C
- 4) ROSNÝ BOD A TEPLOTA
 - a) Pravdepodobnosť búrky – hodnota rosného bodu stúpla nad 17 °C
 - b) Vysoká pravdepodobnosť búrky – hodnota rosného bodu stúpla za posledných 12 hodín o 6 °C
 - c) Nebezpečenstvo nočného mrazu – rosný bod klesol pod 0 °C

Príklad funkcie na overenie statusu o vysokej pravdepodobnosti búrky (rosný bod musí stúpnúť o 6 °C za posledných 12 hodín):

```
function dPIB6Ci12hrs = dewPointIncreaseBy6CIn12Hours(dewpoints)
    l = length(dewpoints);
    dewPMax = max (dewpoints(l/2:l));
    dewPMin = min (dewpoints(l/2:l));
    if( (dewPMax - dewPMin) >= 6 )
        dPIB6Ci12hrs = 1;
    else
        dPIB6Ci12hrs = 0;
    end
end
```



Obr. 11: UI ukážka vizualizácie predpovede vypočítanej na ThingSpeak.

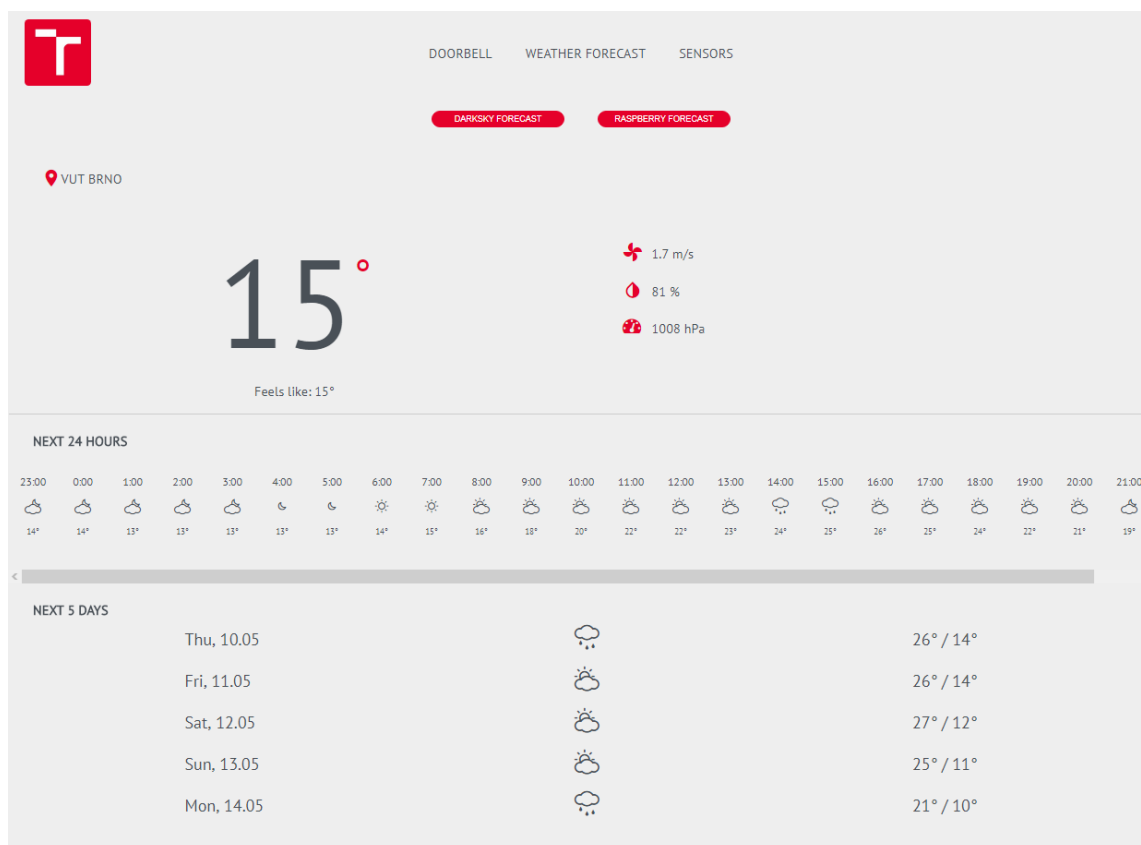
Táto predpoveď bola vytvorená ako ilustrácia príkladu analýzy údajov na ThingSpeak cloude a aktuálnosť sa pohybuje v rozsahu niekoľko hodín. Nezastupuje klasickú predpoveď počasia, ale iba informuje o základných možných stavoch počasia na základe elementárnych poznatkov o predikcii počasia.

6.6 PREDPOVEĎ Z OFICIÁLNEJ WEBOVEJ APLIKÁCIE

Ako ďalšiu funkciu poskytuje domáca stanica jednoduchý prístup k aktuálnym údajom o počasí s predpoveďou na najbližšie dni. Tieto údaje sú stiahnuté z webovej aplikácie *Darksky*. *Darksky* je spoločnosť, ktorá sa špecializuje na predpovedanie a vizualizáciu počasia. Poskytuje neplatenú API pre získanie údajov o predikcii počasia, aktuálnom stave a histórii dát. Po zaregistrovaní aplikácia vygeneruje API kľúč, ktorý treba pridať do URL pre získanie dát. Táto URL musí obsahovať zemepisnú šírku a dĺžku daného miesta, pre ktoré chceme získať údaje. Poloha bola nastavená na adresu fakulty FSI VUT. Ďalej bolo potrebné zadať jednotky SI sústavy. Pre HTTP požiadavku bol použitý npm balíček „*request*“. Nasledujúci kód zobrazuje nastavenie všetkých údajov pre získanie dát.

```
// default address - Brno, Technicka, 612 00, Czech R.
const address = {
  latitude: 49.224317,
  longitude: 16.577616
};
//API key from Darksky account
const apiKey = '077e9a5e77f09466a7e8a6e1bc0cffa5';
//URL to request to get weather data
const weatherURL =
`https://api.darksky.net/forecast/${apiKey}/${address.latitude},${address
.longitude}?units=si`;
```

Po HTTP požiadavke na zadanú URL adresu skript vracia telo odpovede zo servera DarkSky s údajmi. Tieto údaje sú filtrované a v užívateľskom rozhraní sa zobrazujú základné aktuálne údaje ako teplota, pocitová teplota, rýchlosť vetra, tlak a vlhkosť vzduchu. Ďalej je zobrazená hodinová predpoveď pre najbližších 24 hodín a denná pre nasledujúcich 5 dní.



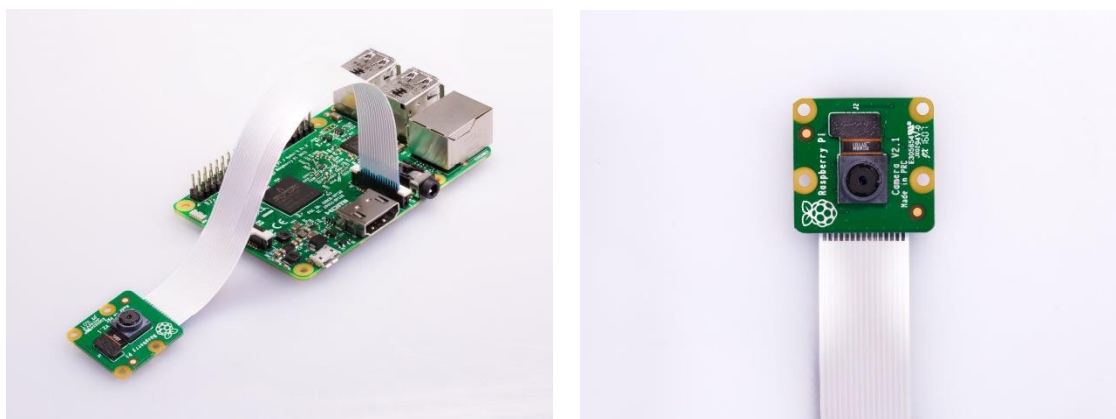
Obr. 12: UI - údaje z DarkSky.

7 ZVONČEK S BEZPEČNOSTNOU KAMEROU

Ďalším modulom domácej stanice je zvonček, ktorý pozostáva z tlačidla a kamery. Funguje ako klasický zvonček, ale je rozšírený o možnosť video streamu z kamery na zvončeku. Tlačidlo bude mať dve funkcie – stlačenie = v danom momente zachytí fotku návštevníka a analyzuje tvár z fotky OPENCV softvérom na rozpoznávanie tváre. Ak sa podoba návštevníka nachádza v kolekcii užívateľov, identifikuje ho. Druhá funkcia – podržanie tlačidla = v užívateľskom rozhraní sa spustí žiadosť o video hovor. Kamera na zvončeku slúži aj ako bezpečnostná kamera – pohľad sa dá spustiť z UI aj bez manuálneho stlačenia tlačidla. Zvonček obsahuje originálnu kameru pre Raspberry Pi model V2.

Čip	8Mpx Sony IMX219 CCD
Foto	až 3280x2464 px
Video	1080p @ 30fps; 720p @ 60 fps; 640x480p @ 90 fps
Konektor	CSI
Rozmery	25 × 20 × 9 mm
IR filter	áno

Tabuľka 3: Parametre kamery V2.



Obr. 13: Raspberry kamera V2.

7.1 STREAM VIDEO A ZACHYTENIE FOTOGRAFIE

Stream videa z kamery umožňuje ovládač UV4L. UV4L je ovládač kamery špeciálne navrhnutý pre Raspberry Pi. Omeškanie streamu je 100-700 ms v závislosti na rýchlosti siete a sile signálu. Pri spustení streamu zaznamenaná najpomalšia odozva, potom sa video ustáli. Tento stream spotrebuje približne 20 % výkonu procesora Raspberry, preto sa spúšťa iba v prípade zapnutia cez užívateľské rozhranie alebo po podržaní tlačidla.

Postup inštalácie UV4L na Raspberry pre operačný systém Raspbian Stretch Lite [10]:

```
$ curl http://www.linux-projects.org/listing/uv4l_repo/lpkey.asc | sudo  
apt-key add -  
$ sudo nano /etc/apt/sources.list -pridať link: deb http://www.linux-  
projects.org/listing/uv4l_repo/raspbian/stretch stretch main  
$ sudo apt-get update  
$ sudo apt-get install uv4l uv4l-raspicam  
$ sudo raspi-config - povoliť v interface kameru  
$ sudo rpi-update  
$ sudo apt-get install uv4l-webRTC  
$ sudo apt-get install uv4l-server uv4l-uvic uv4l-xscreen uv4l-mjpegstream  
uv4l-dummy uv4l-raspidisp
```

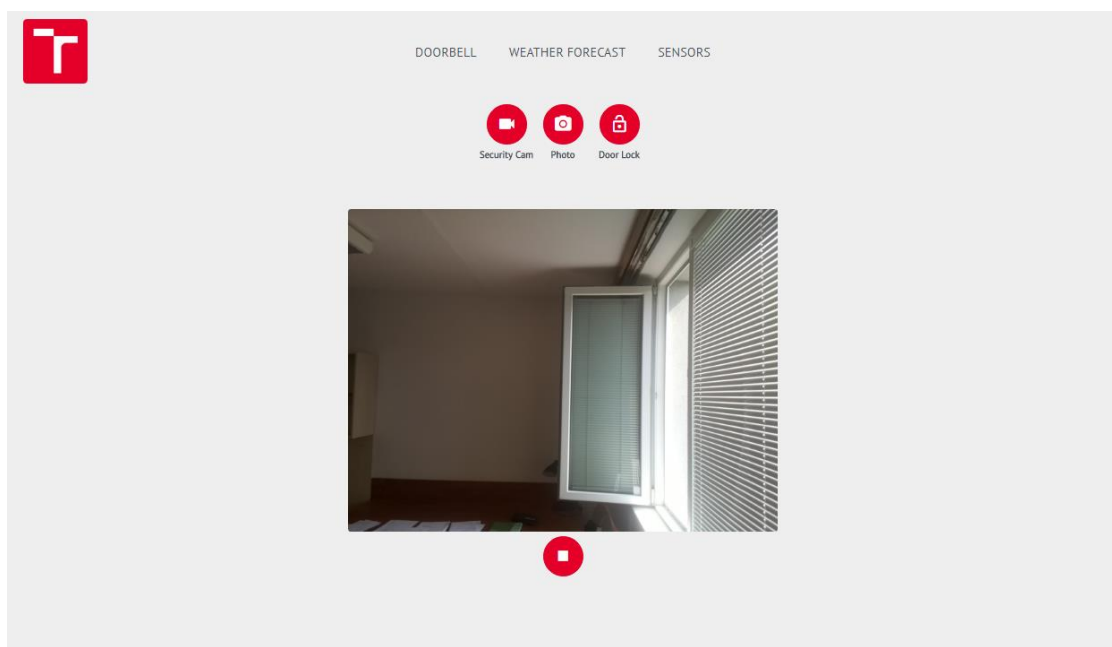
Spustenie servera pre stream videa z kamery zabezpečuje Node.js vykonaním príkazu

```
$ uv4l --driver raspicam --auto-video_nr --width 640 --height 480 --  
encoding h264
```

Stream sa zastavuje príkazom

```
$ pkill uv4l
```

Ovládanie streamu funguje tak, že po stlačení tlačidla v užívateľskom rozhraní s ikonou kamery sa pošle požiadavka na server, ktorý spustí terminál s príkazom pre zapnutie servera.

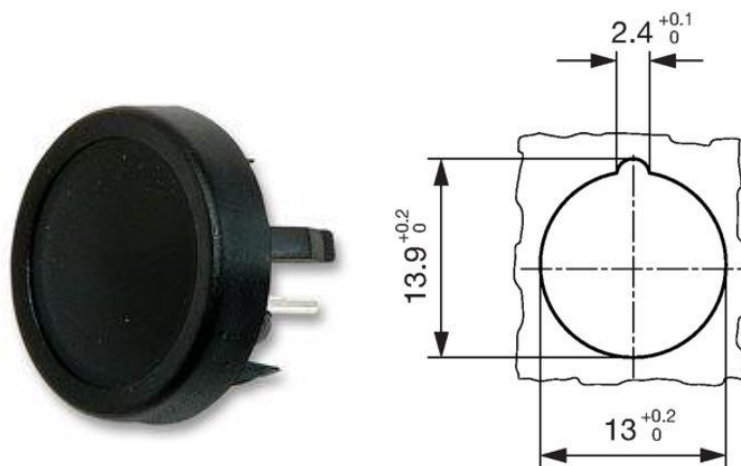


Obr. 14: UI - stream videa.

Pre zachytenie fotky z kamery používam npm balíček „*node-raspistill*“, ktorý zachytí fotku so zadanými parametrami bez nutnosti uloženia a pošle ju na ukážku do užívateľského rozhrania.

7.2 TLAČIDLO

Pre ovládanie zvončeka sa používa tlačítkový spínač značky Schurter. Jedná sa o 1-pólový spínač, ktorý má dve polohy on alebo off. V tlačidle je umiestnená pružinka, vďaka ktorej sa vždy vracia do vypnutej polohy.



Obr. 15: Tlačidlo Schurter [11].

Tlačidlo bude pripojené na Raspberry zapojením “pull-down resistor”. Vstup logickej brány je viazaný na zem. Z úrovne logickej nuly sa môže dostať na logickú jednotku pomocou ovládania mechanického tlačidla.

Pre vyčítanie hodnoty logickej nuly prípadne jednotky z pinu Raspberry používa knižnicu *bcm2835*. Jedná sa o C knižnicu pre Raspberry Pi, ktorá poskytuje prístup k funkciám GPIO a ostatným IO na čipe Broadcom BCM 2835, ktorý sa používa v Raspberry Pi. Umožňuje ovládanie a prepojenie rôznych externých zariadení. Poskytuje funkcie na vyčítanie digitálnych vstupov a nastavenie digitálnych výstupov pomocou SPI, I²C a na prístup k systémovým časovačom.

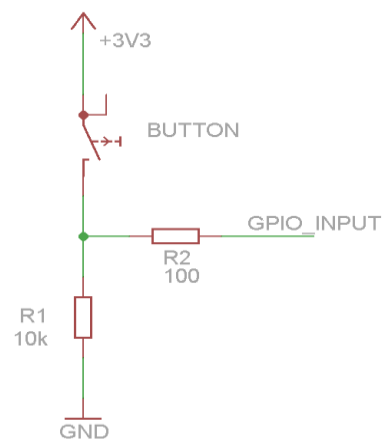
Nasledujúca ukážka kódu napísaného v jazyku C zobrazuje vyčítanie hodnoty logickej nuly alebo jednotky. Rozlíšenie stlačenia a podržania je vytvorené pomocou časovača, ktorý sa spustí po stlačení tlačidla a inkrementuje sa každých 200 milisekúnd. Ak je po odľahčení tlačidla hodnota počítadla väčšia ako 10, tlačidlo bolo podržané. Pri vyčítaní hodnoty je odstránený šum a po rozlíšení stlačenia alebo podržania tlačidla nastane 5 sekundová pauza nutná pre ošetrenie opakovaného stlačenia.

```

while (1)
{
    // read data from Raspi PIN
    uint8_t value = bcm2835_gpio_lev(PIN);

    // logical 1 from PIN
    if(value)
    {
        delay(50); // noise remove
        if(bcm2835_gpio_lev(PIN))
        {
            // init counter for differentiate hold/push button
            int counter = 0;
            while(bcm2835_gpio_lev(PIN))
            {
                delay(200);
                counter++;
            }
            if(counter >= 10)
            {
                printf("btn_hold\t");    // button holded 2 seconds
            }
            else
            {
                printf("btn_push\t");    // button pushed
            }
        }
        fflush(stdout);
        //delay to ignore repeating button push in short time
        delay(5000);
    }
}
}

```



Obr. 16: Schéma zapojenia tlačidla s pull-down rezistorom.

7.3 NÁVRH KRABÍČKY ZVONČEKA

Pre vývoj prototypu krabičky pre zvonček bol zvolený dizajnový 3D CAD/CAM softvér Fusion 360 [12]. Na obrázkoch je zobrazený prototyp krabičky pre 3D tlač. Obsahuje dolný kruhový výrez pre pridanie tlačidla a horný výrez pre objektív kamery.



Obr. 17: Prototyp krabičky - horný pohľad.



Obr. 18: Prototyp krabičky - spodný pohľad.

7.4 ROZPOZNÁVANIE TVÁRE S PYTHON OPENCV

Ľudská tvár hrá podstatnú rolu v sociálnej interakcii a dokresľuje osobnosť človeka. Využívanie podoby tváre ako istého bezpečnostného kľúča získava v posledných rokoch čoraz väčšiu popularitu a pozornosť. Rozpoznávanie tváre má v porovnaní s inými biometrickými systémami ako napríklad otláčok prsta alebo podoba dúhovky oka viacero výhod, pretože snímanie identifikačných znakov tváre môže prebiehať aj z veľkej vzdialenosti od snímacieho zariadenia.

Technické podmienky pre rozpoznávanie tváre:

- a) viditeľnosť tváre – tvár by nemala byť zakrytá a musí byť dobre osvetlená,
- b) rozlíšenie kamery – optimálna vzdialenosť medzi zorničkami očí je 80 až 100 pixelov, pokiaľ je hodnota menšia než 40 pixelov, prudko klesá úspešnosť rozpoznania,
- c) uhol natočenia tváre – softvér pre rozpoznávanie tváre musí rozpoznávať kontúry tváre ako oči, pery, nos a ich vzájomné vzdialenosti. Pri natočení tváre o $\pm 15^\circ$ vo vertikálnom alebo horizontálnom smere dochádza k deformácii pomeru vzdialeností medzi kontúrami tváre a tým k neúspešnej analýze [13].

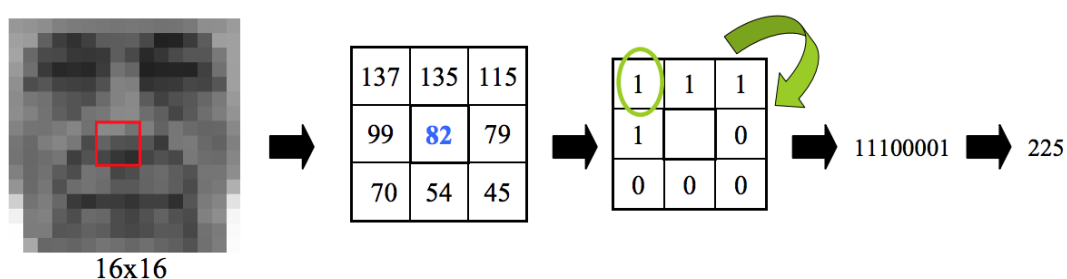
Rozpoznávanie tváre s OpenCV sa delí na viacero krokov. V prvom kroku sa program pripojí k databáze známych užívateľov a načíta z nej ich mená a príslušné ID. Tieto údaje sa uložia do poľa s užívateľskými menami a poľa s identifikačnými číslami užívateľov. Následne po tom program začne prípravu pre tréning známych užívateľov. Fotky známych užívateľov majú štruktúru subpriechinkov obsahujúcich v názve ID konkrétneho užívateľa, ktorého fotky sú v ňom uložené. To zabezpečuje prehľadnú štruktúru uložených fotiek, ktoré možno editovať manuálne aj bez použitia užívateľského prostredia. Príprava pre tréning známych fotiek užívateľov začína konvertovaním fotiek do monochromatických farieb. Ďalej nasleduje načítanie kvalifikátora z OpenCV. Je to program, ktorý rozhoduje o tom, či daná fotka obsahuje tvár alebo nie. Je vyladený na mnohých fotkách, aby dokázal správne analyzovať novú fotografiu. OpenCV nám ponúka dva kvalifikátory pre detekciu tváre:

- a) „*Haar Classifier*“,
- b) „*LBP Classifier*“.

Funkcia prípravy pre tréning známych užívateľov vráti tváre a ID užívateľov. Po úspešnej príprave fotiek nasleduje tréning tvárí. Pre túto funkciu je nutné inicializovať identifikátor tváre. OpenCV má zabudované tieto tri identifikátory tváre:

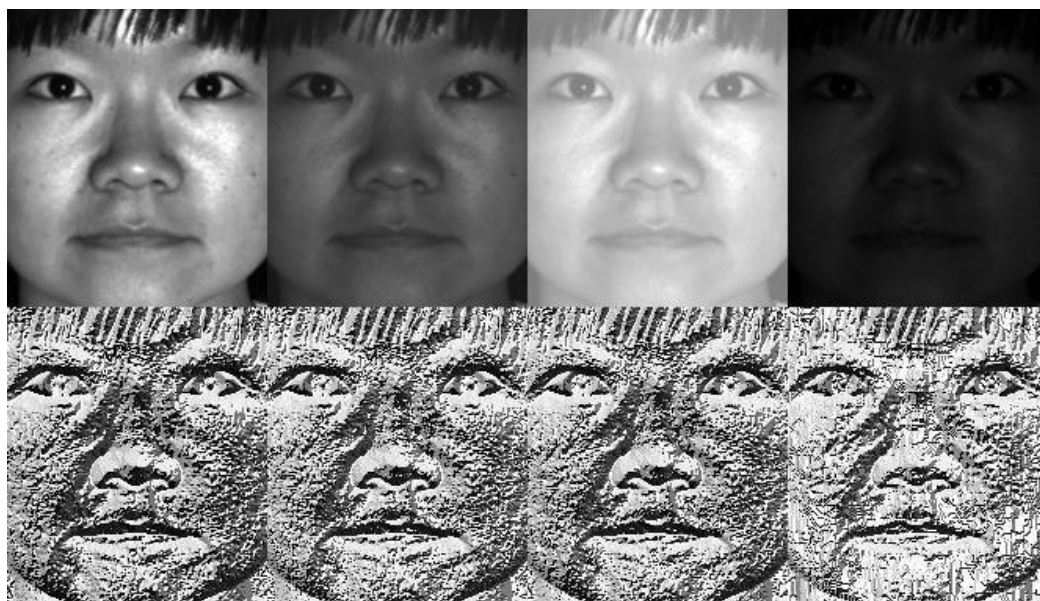
- a) *EigenFaces Face Recognizer Recognizer*,
- b) *FisherFaces Face Recognizer Recognizer*,
- c) *Local Binary Patterns Histograms (LBPH) Face Recognizer*.

Tento softvér využíva LBPH rozpoznávanie tváre. Prvé dva rozpoznávacie algoritmy závisia na svetelných podmienkach pri zachytávaní fotografie. LBPH algoritmus je voči svetelným podmienkam menej závislý. Jeho princíp je postavený na tom, že neberie celú fotografiu ako celok, ale rozdelí ju na lokálne oblasti, ktoré porovnáva so susednými. Vysekne okno 3×3 pixely, presunie ho na jeden obrázok pri každom pohybe po lokálnych častiach. Ku každému pixelu priradí jeho hodnotu intenzity, a od stredového pixelu odčítava hodnoty susedných pixelov. Pixely s menšou alebo rovnakou intenzitou sú označené 0 a s väčšou hodnotou 1. Po tomto procese tieto hodnoty zoradí za sebou v smere hodinových ručičiek. Keďže týchto čísel je 8, vznikne z tejto rady binárne číslo, ktoré sa prevedie na decimálne.



Obr. 19: Princíp fungovania LBPH algoritmu [14].

Z týchto decimálnych čísel sa vytvorí histogram ich výskytu vo fotografii. Pre každú známu tvár sa generuje takýto histogram, ktorý sa uloží pre neskoršie rozpoznávanie tváre. Pri spustení analýzy tváre sa z fotografie návštevníka vygeneruje takýto histogram, ktorý sa porovnáva so známymi a hľadá sa najlepšia zhoda [14].



Obr. 20: Vytvorenie histogramu z fotiek s odlišnou kvalitou osvetlenia [14].

Vstupnými parametrami identifikátoru tváre sú samotné podoby tváre z kvalifikátoru pre detekciu tváre a príslušné ID užívateľa. Po ukončení procesu tréningu tváre je softvér pripravený pre rozpoznávanie užívateľov. Tento program sa spúšťa automaticky po spustení servera na domácej stanici, začne s tréningom známych užívateľov a po pár desiatkach sekundách až minútach v závislosti od veľkosti databázy a kvality fotiek je pripravený na rozpoznávanie.

Funguje na princípe, že v nekonečnej while slučke číta vstup zo servera a na základe tohto vstupu ďalej uskutočňuje úlohy. Existujú dva možné vstupy zo servera. V prvom prípade sa jedná o pokyn na začatie rozpoznávania. Server stanice zachytí fotku po stlačení tlačidla a pošle príkaz pre spustenie analýzy tváre návštevníka. Softvér pre rozpoznávanie tváre načíta aktuálnu fotku, nájde na nej tvár a hľadá zhodu so známymi užívateľmi. Následne program pošle na server ID užívateľa s najlepšou zhodou tváre, server ho nájde v databáze a upraví čas jeho poslednej návštevy na aktuálny. V druhom prípade sa jedná o pridanie nového užívateľa do databázy. Softvér začne s tréningom uložených fotiek nového užívateľa a pre ďalšiu analýzu je prístupná už aj jeho nová podoba.

7.5 SQLite DATABÁZA

Ako nástroj pre ukladanie informácií o užívateľoch bol zvolený databázový systém SQLite. Je to nezávislý, jednoduchý, rýchly a bezserverový databázový stroj SQL. Nie je založená na klasickom databázovom mechanizme typu klient-server ako napríklad MySQL ale na princípe lokálneho ukladania dát pre jednotlivé aplikácie a zariadenia. Je to jedna z najrozšírenejších databáz so širokým poľom pôsobenia. Nevyžaduje žiadnu správu, preto sa používa v zariadeniach, ktoré musia fungovať bez odbornej podpory človeka. Sem patria zariadenia internetu vecí. Ďalšou výhodou SQLite databázy je, že sa s ňou dá pracovať po stiahnutí a nainštalovaní potrebných knižníc zároveň v systémoch napísaných v jazykoch Python a Node.js [15].

Informácie o užívateľovi, ktoré domáca stanica ukladá, obsahujú jeho meno a priezvisko, právo na odomknutie dverí, čas poslednej návštevy a identifikačné číslo automaticky vygenerované databázou. SQLite indexuje identifikačné čísla od 1.

Proces uloženia nového užívateľa cez užívateľské rozhranie začína otvorením karty s názvom „Zvonček“ a tlačidlom „Pridať nového užívateľa“. Po kliknutí sa server pripojí k databáze, vyčíta z nej ID posledného pridaného užívateľa a vytvorí subpriečinok s číslom o jedna väčšie ako toto ID. Po kliknutí na tlačidlá zachytenia fotky sa do tohto priečinka ukladajú fotky tváre nového užívateľa podľa indexu stlačeného tlačidla. Server pošle do užívateľského rozhrania zmenšenú ukážku fotky pre overenie kvality zachytenej fotky. Po vyplnení informácie o mene a práve na odomknutie sa po potvrdení uloží užívateľ do databázy. Server pošle príkaz softvéru pre rozpoznávanie tváre, aby začal

s tréningom tváre nového užívateľa. V prípade zrušenia celého procesu pri ukladaní sa odstráni vytvorený priečinok aj s fotkami.

Pre náhľad a správu databázy je v užívateľskom rozhraní vyčlenený špeciálny komponent. Sú v ňom zobrazení všetci uložený užívatelia s ich nastaveniami. Po kliknutí na zvoleného užívateľa sa zobrazí jeho profil s fotkou a konkrétnymi údajmi. V tomto komponente je možné užívateľov takisto vymazať z databázy. Úprava údajov je možná načítaním databázového súboru prehliadačom SQLite. Pre správny chod softvéru pre rozpoznávanie tváre je nutné mať v databáze uložených minimálne dvoch užívateľov.

7.6 PRINCÍP FUNGOVANIA ZVONČEKA

Princíp fungovania zvončeka je založený na subprocessoch, ktoré spúšťa server a tie sa starajú o správne fungovanie tlačidla a kamery na zvončeku. Pre výstup z tlačidla je spustený „buttonlistener“, ktorý spustí skript na vyčítanie hodnoty pinu z Raspberry (kapitola 7.2)

```
// run child process, which is listening to button output
const btnListenerPath = path.join(__dirname, './scripts/button');
const buttonListener = spawn(btnListenerPath)
```

Podľa výstupu z toho subprocessu sa ďalej posielajú inštrukcie do softvéru pre rozpoznávanie tváre alebo priamo do užívateľského rozhrania cez WebSocket. Ďalší spustený subprocess obstaráva analýzu tváre. Server spustí Python skript pomocou knižnice „python-shell“. Táto knižnica bola vybraná pre jednoduchú komunikáciu medzi Node.js a Python.

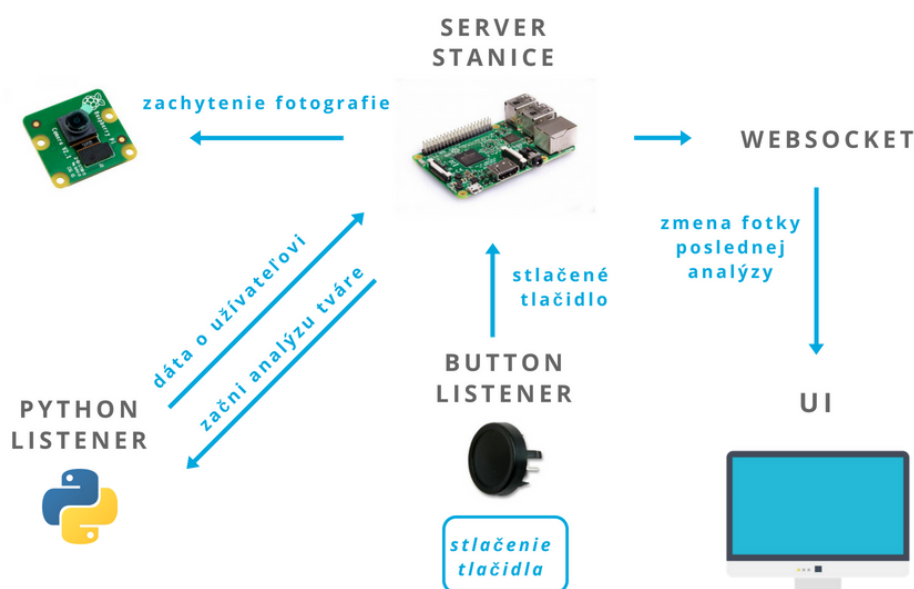
```
// run child process for communicating between NODEJS and Python - stdin
and stdout
const pythonListener = new PythonShell('python_stdin.py');
```

7.6.1 STLAČENIE TLAČIDLA

Po krátkom stlačení tlačidla na zvončeku „buttonlistener“ pošle správu serveru stanice, ktorý po overení výstupu pošle príkaz do terminálu pre zachytenie fotografie. Ak zachytenie fotografie prebehlo úspešne, pošle správu cez PythonShell do softvéru pre rozpoznávanie tváre, aby danú fotografiu načítal, detekoval na nej tvár, prehľadal kolekciu fotiek známych užívateľov a našiel z nich najlepšiu zhodu. Po ukončení analýzy posielala cez PythonShell správu serveru o ukončení analýzy. V prípade úspešného vyhľadania zhody pošle ďalšiu správu, ktorá obsahuje ID identifikovaného užívateľa. Po získaní tejto správy server otvorí databázu, nájde v nej daného užívateľa a upraví čas jeho poslednej návštevy. Asynchrónne s týmto procesom prebieha aktualizácia ukážky fotografie z poslednej analýzy tváre. Dáta medzi serverom a užívateľským rozhraním sa vymieňajú cez WebSocket.

WebSocket je komunikačný protokol, ktorý umožňuje obojsmernú komunikáciu medzi serverom a webovým prehliadačom. Je to nezávislý protokol založený na základe TCP (Transmission Control Protocol). Pred odosielaním dát musí klient a server medzi sebou naviazať spojenie. Výhodou protokolu WebSocket oproti klasickému HTTP (Hypertext Transfer Protocol) je, že umožňuje real-time prenos dát. To znamená, že obsah správ zo serveru do webového prehliadača prebieha bez nutnosti požiadavku od klienta počas otvoreného spojenia. Týmto spôsobom prebieha obojsmerná komunikácia medzi serverom a klientom [16].

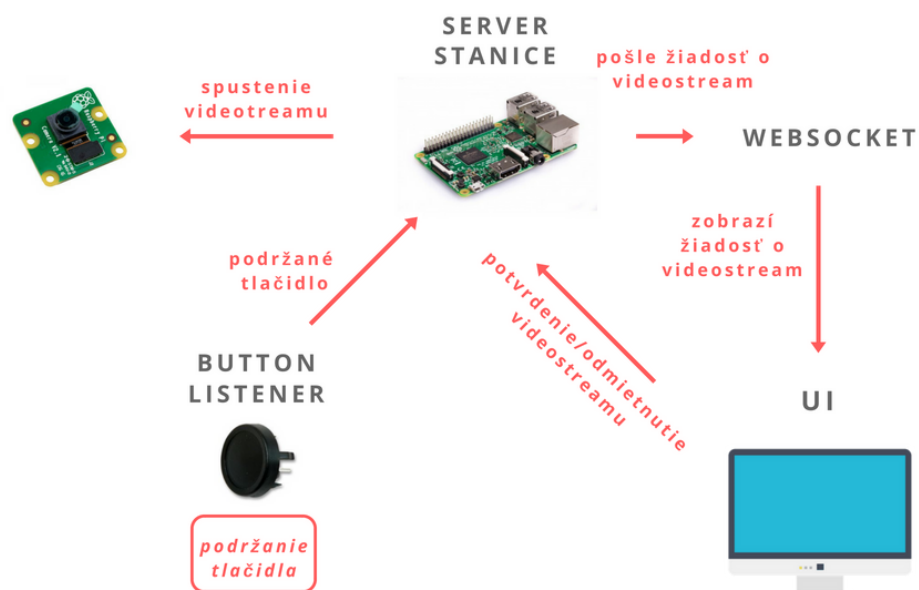
Po vytvorení WebSocketu a prípadnom naviazaní spojenia sa odošle náhľad fotografie. V prípade úspešnej analýzy obsahuje identifikátor tváre s popisom mena užívateľa. Náhľad fotografie je zmenšený pre rýchlejšiu odozvu a konvertovaný do formátu Base64. Base64 je kódovanie, ktoré prevádza binárne dáta na text a umožňuje tak jednoduchší prenos fotografie. Princíp komunikácie je zobrazený na nasledujúcej mape mysle.



Obr. 21: Princíp komunikácie po stlačení tlačidla.

7.6.2 PODRŽANIE TLAČIDLA

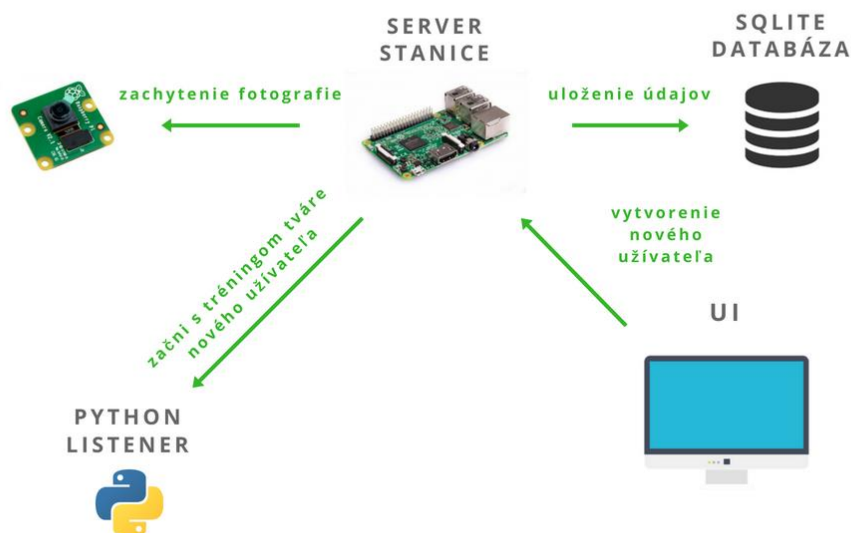
Po podržaní tlačidla na zvončeku „buttonlistener“ pošle správu serveru stanice, ktorý vytvorí WebSocket a po úspešnom pripojení klienta pošle žiadosť o video stream. V užívateľskom rozhraní sa zobrazí okno s tlačidlami prijatie a odmietnutie streamu. Po kliknutí na odmietnutie sa tento komponent vypne. Po stlačení tlačidla s prijatím hovoru server spustí video stream pomocou ovládača UV4L a presmeruje stránku na komponent s video streamom. Princíp fungovania stanice po podržaní tlačidla je vysvetlený na nasledujúcej mape mysle.



Obr. 22: Princíp komunikácie po podržaní tlačidla.

7.6.3 PRIDANIE NOVÉHO UŽÍVATEĽA DO DATABÁZY

Pridávanie nových užívateľov prebieha cez užívateľské rozhranie. Podrobný popis, ako prebieha proces ukladania je opísaný v kapitole 7.5. Na nasledujúcej mape mysle je graficky znázornený priebeh vytvorenia nového užívateľa.



Obr. 23: Princíp fungovania uloženia nového užívateľa.

8 API DOKUMENTÁCIA

V nasledujúcej kapitole je podrobne rozpísaný zoznam API dokumentácie – tzv. „endpointov“ serverovej časti aplikácie domácej stanice. Komunikácia medzi frontendom a backendom prebieha pomocou GET a POST metód.

Popis	Spustí video stream.
URL	/doorbell/startstream
Metóda	GET
Úspešná odpoveď	{“status”:true}
Neúspešná odpoveď	{“status”:false}

Popis	Zastaví video stream.
URL	/doorbell/stopstream
Metóda	GET
Úspešná odpoveď	{“status”:true}
Neúspešná odpoveď	{“status”:false}

Popis	Načíta fotku poslednej analýzy tváre.
URL	/doorbell/lastanalyse
Metóda	GET
Úspešná odpoveď	{“status”:true, “imgBase64”:“ ...” }
Neúspešná odpoveď	{“status”:false}

Popis	Zachytí fotografiu s nastavenými parametrami.
URL	/doorbell/takephoto
Metóda	GET
Úspešná odpoveď	fotka z Raspberry kamery
Neúspešná odpoveď	-

Popis	Získanie dát zo senzora na meranie meteorologických veličín.
URL	/sensors/bme280

Metóda	GET
Úspešná odpoveď	{“status”:true, “sensorData”:“ ...”}
Neúspešná odpoveď	{“status”:false}

Popis	Vytvorenie nového priečinku v tréningových dátach pre rozpoznávanie tváre.
URL	/users/adduser
Metóda	GET
Úspešná odpoveď	{“status”:true, “newUserID”:“ ...”}
Neúspešná odpoveď	{“status”:false}

Popis	Pri prerušení ukladania nového užívateľa zmaže vytvorený priečinok aj s prípadnými už vytvorenými fotkami.
URL	/users/cancelusersaving
Metóda	GET
Úspešná odpoveď	{“status”:true}
Neúspešná odpoveď	{“status”:false}

Popis	Zachytí fotku užívateľa, pomenuje ju podľa zadaného identifikátora a uloží do priečinka správneho užívateľa. V odpovedi vráti náhľad fotky.
URL	/users/takeuserphoto
Metóda	POST
Parametre	photoID
Úspešná odpoveď	{“status”:true, “imgBase64”:“ ...”}
Neúspešná odpoveď	{“status”:false}

Popis	Načíta prvú uloženú fotku užívateľa podľa vstupného parametru
URL	/users/showuser
Metóda	POST
Parametre	userID

Úspešná odpoveď	{“status”:true, “imgBase64”:“ ...”}
Neúspešná odpoveď	{“status”:false}

Popis	Otvorí databázu a vráti všetkých uložených užívateľov aj s príslušnými informáciami.
URL	/users/getallusers
Metóda	GET
Úspešná odpoveď	{“status”:true, “users”:“ ...”}
Neúspešná odpoveď	{“status”:false}

Popis	Otvorí databázu a vráti ID užívateľa, ktorý bol posledný identifikovaný rozpoznávaním tváre.
URL	/users/lastentry
Metóda	GET
Úspešná odpoveď	{“status”:true, “lastEntryID”:“ ...”}
Neúspešná odpoveď	{“status”:false}

Popis	Vymaže z databázy užívateľa so zadaným ID
URL	/users/deleteuser
Metóda	POST
Parametre	id
Úspešná odpoveď	{“status”:true}
Neúspešná odpoveď	{“status”:false}

Popis	Uloží do databázy užívateľa so zadanými parametrami
URL	/users/saveuser
Metóda	POST
Parametre	name, unlock_rights
Úspešná odpoveď	{“status”:true, “user”:“(uložený užívateľ)”}

Neúspešná odpoveď	{“status”:false}
--------------------------	------------------

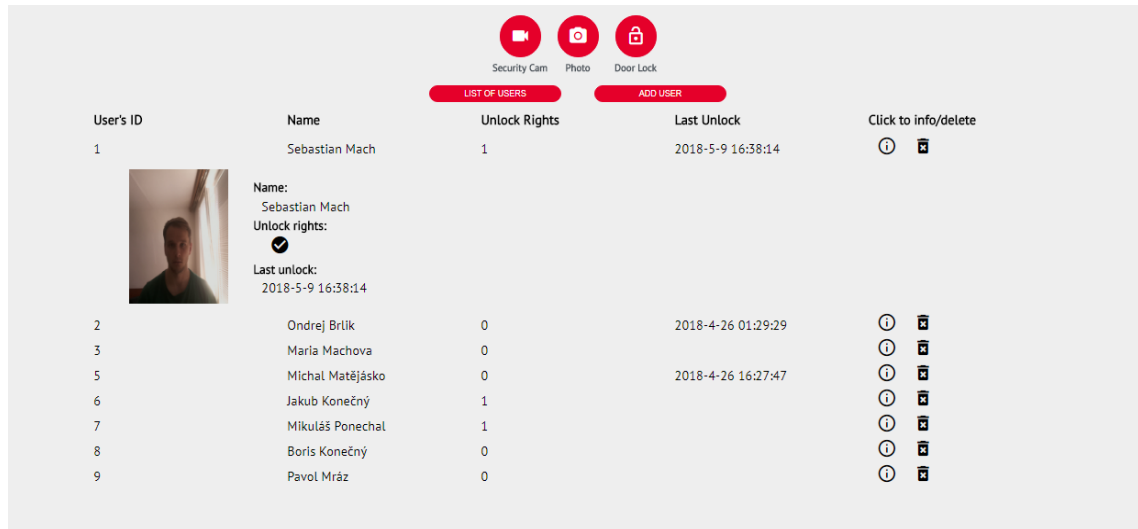
Popis	Pripojí sa na DarkSky a získa údaje o počasí a predpovedi.
URL	/forecast/darkskyforecast
Metóda	GET
Úspešná odpoveď	{“status”:true, “weatherInfo”:“ ...”}
Neúspešná odpoveď	{“status”:false}

Popis	Pripojí sa na ThingSpeak kanál, v ktorom sa ukladajú správy o vypočítanej predpovedi počasia a vráti posledný vstup do tohto kanála
URL	/forecast/raspberryforecast
Metóda	GET
Úspešná odpoveď	{“status”:true, “forecastInfo”:“ ...”}
Neúspešná odpoveď	{“status”:false}

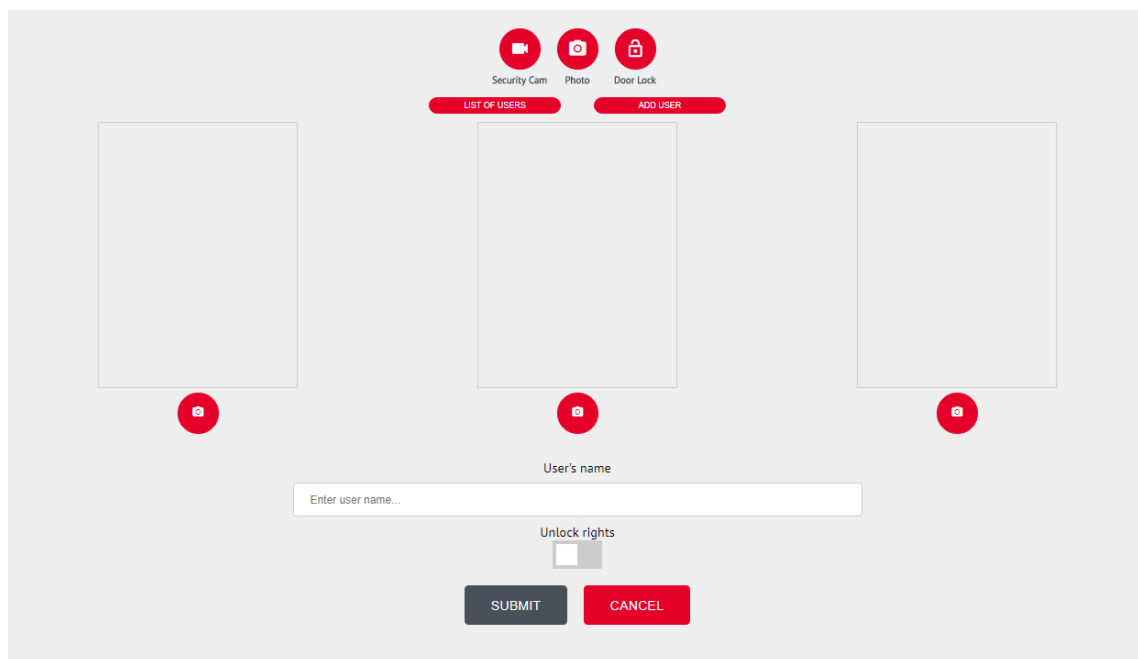
Tabuľka 4: API dokumentácia.

9 UKÁŽKY Z UŽÍVATEĽSKÉHO ROZHRAINIA

V tejto kapitole sú zobrazené ukážky z užívateľského rozhrania vytvoreného v React.js, ktoré ešte neboli zobrazené v predchádzajúcich kapitolách tejto práce.

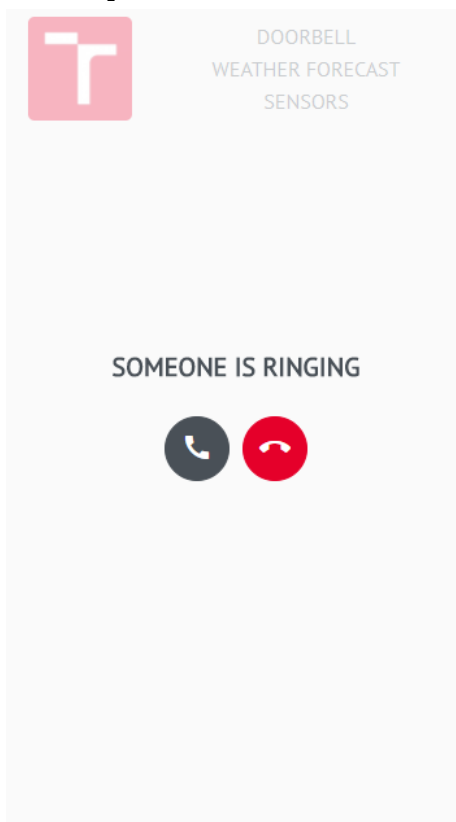


Obr. 24: Ukážka UI - Zoznam uložených užívateľov.

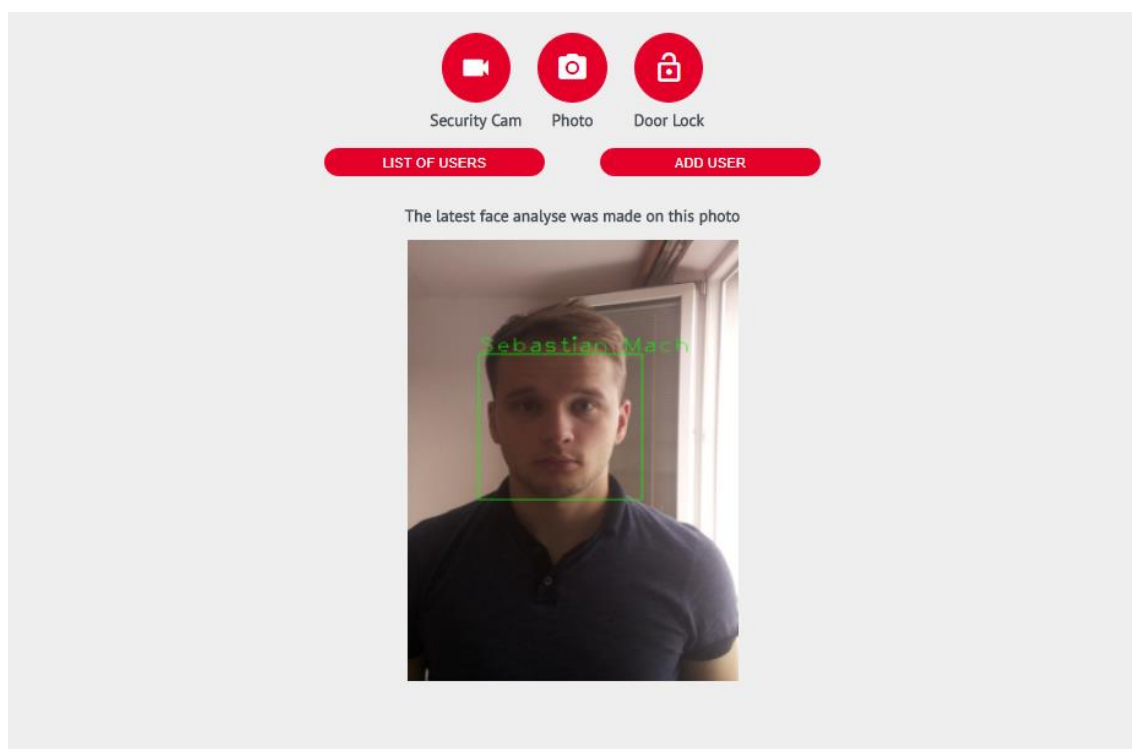


Obr. 25: Ukážka UI - Pridanie nového užívateľa.

Užívateľské prostredie je responzívne pre mobilné zariadenia. Na nasledujúcom obrázku je príklad náhľadu na užívateľské prostredie v mobilnom telefóne.



Obr. 26: Ukážka UI - Žiadosť o video stream.



Obr. 27: Ukážka UI – Analýza tváre.

10 ZÁVER

Hlavným cieľom tejto bakalárskej práce bol návrh domácej inteligentnej stanice, ktorá je založená na modulárnom systéme pripájania nových zariadení. Prvým krokom k funkčnej stanici bolo zoznámenie sa s počítačom Raspberry Pi 3, ktorý tvorí jadro celej domácej stanice. Toto jadro sa stará o spúšťanie všetkých vonkajších periférií a správu senzora. Pre ovládanie tohto hlavného centra bol vybraný systém Node.js, ktorého základný popis fungovania je popísaný v kapitole č.4. Pre optimalizáciu a zjednodušenie ovládania všetkých funkcií stanice bolo vytvorené užívateľské prostredie. Podrobný popis technológie, v ktorej bolo vytvárané UI je v kapitole č.5.

Ako prvá funkcia domácej inteligentnej stanice bola vyvinutá meteostanica s meraním teploty, vlhkosti a tlaku vzduchu. Táto funkcia bola neskôr rozšírená o predpovede počasia. Jedna získaná z oficiálnych zdrojov a druhá, ktorej výpočet prebiehal z dát na ThingSpeak cloude. Ako vlastne táto časť stanice funguje je opísané v kapitole č.6.

Druhý modul, ktorý stanica obsahuje je zvonček s kamerou. Tento zvonček slúži ako bezpečnostná kamera alebo ako klasický zvonček, u ktorého ale odpadá nutnosť fyzickej prítomnosti osoby pri dverách. Kamera umiestnená v tele krabičky zvončeka umožňuje video stream do webového prehliadača. Tento zvonček má pridanú hodnotu v extra funkcii. Podporuje rozpoznávanie tváre návštevníka z databázy známych užívateľov. Tento softvér beží pomocou OpenCV v Pythone. Percento úspešnosti rozpoznávania tváre ešte nebolo presne určené a bude presné až po adekvátnom zostavení a testovaní. Informácie v kapitole č.7 opisujú na čom všetkom závisí presnosť tejto analýzy aj s kompletným popisom tejto časti stanice.

V kapitole č.8 je rozpísaná API dokumentácia serveru, ktorá poskytuje informácie o tom, ako ovládať domácu stanicu.

V poslednej kapitole sú zobrazené ukážky z UI, ktoré je možné spustiť v akomkoľvek modernom webovom prehliadači na počítači alebo mobilnom zariadení.

Ako už bolo spomenuté domáca stanica je navrhnutá ako modulárny systém, čiže je možné ju vylepšiť pomerne jednoducho o ďalšie funkcie a zariadenia. Pre jednoduchšie a presnejšie meranie meteorologických veličín zmenou na bezdrôtové meranie pomocou zariadenia s WiFi alebo Bluetooth. Pridaním nových typov senzorov je možné rozšíriť počet meraných veličín.

Vylepšením zvončeka s pridaním mikrofónu by bola možná obojsmerná komunikácia medzi návštevníkom a správcom. Pre presnejšie rozpoznávanie tváre existuje možnosť neurónovej siete, ale v tomto smere je stanica limitovaná výpočtovým výkonom a ako vhodnejšia alternatíva sa javí analýza fotiek na externých službách. Táto možnosť je zase limitovaná legislatívou o spracovaní osobných údajov. Ďalší vývoj systému pre overenie totožnosti by sa mal vyvíjať viac smerom k viacfaktorovej autentifikácii.

11 ZOZNAM POUŽITEJ LITERATÚRY

- [1] Raspberry Pi. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 20.5.2018 [cit. 2018-05-22]. Dostupné z: https://en.wikipedia.org/wiki/Raspberry_Pi
- [2] *Raspbian Stretch Lite* [online]. [cit. 2018-04-2]. Dostupné z: <https://www.raspberrypi.org/downloads/raspbian/>
- [3] *ThingSpeak* [online]. Natick USA [cit. 2018-05-3]. Dostupné z: <https://thingspeak.com>
- [4] *Mobaxterm* [online]. [cit. 2018-05-3]. Dostupné z: <https://mobaxterm.mobatek.net>
- [5] *Winscp* [online]. [cit. 2018-05-2]. Dostupné z: <https://winscp.net/eng/docs/lang:cs>
- [6] *Node.js* [online]. [cit. 2018-05-4]. Dostupné z: <https://nodejs.org/en/about/>
- [7] *React.js* [online]. [cit. 2018-04-21]. Dostupné z: <https://reactjs.org>
- [8] *BME280* [online]. [cit. 2018-05-4]. Dostupné z: <https://www.aliexpress.com>
- [9] *Using the BME280 I2C Temperature and Pressure Sensor in Python* [online]. In: . [cit. 2018-05-15]. Dostupné z: <https://www.raspberrypi-spy.co.uk/2016/07/using-bme280-i2c-temperature-pressure-sensor-in-python/>
- [10] *Installation for ARM (Raspberry Pi)* [online]. In: . [cit. 2018-05-16]. Dostupné z: <http://www.linux-projects.org/uv4l/installation/>
- [11] *Tlačítkový spínač Schurter* [online]. [cit. 2018-05-1]. Dostupné z: <https://www.gme.cz/tlacitkovy-spinac-schurter-1241-1113-7097-ip65>
- [12] *Fusion 360* [online]. [cit. 2018-05-21]. Dostupné z: <https://www.fusion360.cz>
- [13] SLACH, Martin. *Rozpoznávání obličejů* [online]. 3.4.2016 [cit. 2018-05-22]. Dostupné z: <https://www.securitymagazin.cz/technologie/-1404049866.html>
- [14] *Face Recognition with OpenCV and Python* [online]. [cit. 2018-05-22]. Dostupné z: <https://github.com/informramiz/opencv-face-recognition-python>
- [15] *About SQLite* [online]. [cit. 2018-05-22]. Dostupné z: <https://www.sqlite.org/about.html>
- [16] WebSocket. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-22]. Dostupné z: <https://cs.wikipedia.org/wiki/WebSocket>
- [17] *Darksky* [online]. [cit. 2018-05-22]. Dostupné z: <https://darksky.net/about>
- [18] *Face detection using OpenCV and Python* [online]. [cit. 2018-05-2]. Dostupné z: <https://www.superdatascience.com/opencv-face-detection/>
- [19] *Predpoved' počasia podľa rosného bodu, teploty, vlhkosti a tlaku vzduchu* [online]. [cit. 2018-04-20]. Dostupné z: <http://www.kstst.sk/pages/vht/meteo/pred.htm>
- [20] *BME280 - Zdrojový kód pre získanie údajov zo senzora* [online]. [cit. 2018-04-25]. Dostupné z: <https://github.com/ControlEverythingCommunity/BME280>

12 ZOZNAM OBRÁZKOV A TABULIEK

- Obr. 1: Raspberry Pi 3 Model B
- Obr. 2: Blokový diagram Thingspeak
- Obr. 3: Bloková schéma serveru
- Obr. 4: Bloková schéma UI + server
- Obr. 5: Horná lišta UI s údajmi zo senzora BME280
- Obr. 6: BME280
- Obr. 7: Schéma zapojenia senzora
- Obr. 8: ThingSpeak graf vlhkosti vzduchu
- Obr. 9: Bloková schéma TS-server
- Obr. 10: Nastavenie TimeControl
- Obr. 11: UI ukážka vizualizácie predpovede vypočítanej na TS
- Obr. 12: UI - údaje z DarkSky
- Obr. 13: Raspberry kamera V2
- Obr. 14: UI - stream videa
- Obr. 15: Tlačidlo Schurter
- Obr. 16: Schéma zapojenia tlačidla s pull-down rezistorom
- Obr. 17: Prototyp krabičky - horný pohľad
- Obr. 18: Prototyp krabičky - spodný pohľad
- Obr. 19: Princíp fungovania LBPH algoritmu
- Obr. 20: Vytvorenie histogramu z fotiek s odlišnou kvalitou osvetlenia
- Obr. 21: Princíp komunikácie po stlačení tlačidla
- Obr. 22: Princíp komunikácie po podržaní tlačidla
- Obr. 23: Princíp fungovania uloženia nového užívateľa
- Obr. 24: Ukážka UI - Zoznam uložených užívateľov
- Obr. 25: Ukážka UI - Pridanie nového užívateľa
- Obr. 26: Ukážka UI - Žiadosť o video stream
- Obr. 27: Ukážka UI – Analýza tváre

Tabuľka 1: Základné parametre Raspberry Pi 3 Model B

Tabuľka 2: Údaje o senzore BME280

Tabuľka 3: Parametre kamery V2

Tabuľka 4: API dokumentácia