Slovenská technická univerzita v Bratislave Fakulta informatiky a informačných technológií Ilkovičova 2, 842 16, Bratislava 4

## Internet vecí v našich životoch [IoT]

Inžinierske dielo – moduly systému LS

Tím:	č. 20
Pedagogický vedúci tímu:	Ing. Tomáš Kováčik, PhD
Členovia tímu:	Barbora Čelesová, Tomáš Koreň, Jakub Pullmann, Michal
	Puškáš, Matúš Sosňak, Peter Stofaňák, Jozef Vaľko,
Akademický rok:	2017 / 2018

## Obsah

1	Modul –	Arduino1
	1.1 Ana	lýza1
	1.1.1	Analýza zvuku včiel 1
	1.1.2	Analýza posielania správ - Update
	1.1.3	Analýza úspory batérie
	1.2 Náv	rh
	1.3 Imp	lementácia 4
	1.3.1	Implementácia merania váhy 4
	1.3.2	Implementácia posielania hodnôt po zmene
	1.3.3	Implementácia detegovania hodnôt mimo rozsahu
	1.3.4	Implementácia autokalibrácie akcelerometra
	1.3.5	Implementácia LED diód
	1.4 Test	ovanie
	1.4.1	Testovanie hmotnosti
	1.4.2	Testovanie posielania správy iba po zmene
	1.4.3	Počiatočné testovanie A/D prevodníka HX7118
	1.4.4	Testovanie LED diód9
2	Modul –	Server
	2.1 Ana	lýza10
	2.1.1	Analýza SigFox Callback - update 10
	2.2 Imp	lementácia
	2.2.1	Ukladanie nameraných údajov 10
3	Modul –	Webová stránka produktu 12
	3.1 Náv	rh12
	3.1.1	Podstránka kontakt 12
	3.1.2	Podstránka vytvorenia objednávky
	3.1.3	Podstránka manažmentu objednávok 13
	3.2 Imp	lementácia14
	3.2.1	Architektúra webovej stránky
	3.2.2	Funkcionalita vytvorenia a potvrdenia objednávky15
	3.2.3	Funkcionalita mazania používateľa16
	3.2.4	Notifikácia používateľa 17
	3.2.5	Profil používateľa
	3.2.6	Prezentačná stránka produktu
	3.2.7	Admin portál

	3.2.8	Funkcionalita zobrazenia mapy	24
	3.3 T	Testovanie	25
	3.3.1	Testovanie vytvorenia objednávky	25
	3.3.2	Testovanie zobrazenia novej objednávky v administrátorskom portáli	26
	3.3.3	Testovanie potvrdenia objednávky	27
	3.3.4	Testovanie kontaktnej podstránky	28
4	Modu	ıl – Android	29
	4.1 N	Návrh mobilnej aplikácie	29
	4.1.1	Návrh zobrazenia grafov v android aplikácii	29
	4.1.2	Profil používateľa	29
	4.1.3	Návrh zobrazenia mapy v android aplikácii	29
	4.2 In	Implementácia	30
	4.2.1	Profil používateľa	31
	4.2.2	Prihlasovacia stránka	31
	4.2.3	Registračný formulár	31
	4.2.4	Základné zobrazenie úľov	32
	4.2.5	Zobrazenie detailu úľa	33
	4.2.6	Zobrazenie všetkých nameraných dát pre konkrétny úľ	33
	4.2.7	Objednávka zariadenia	33
	4.2.8	O projekte	34
	4.2.9	Notifikácie	34
	4.2.10	0 Nastavenie hraničných hodnôt	34
	4.2.11	1 Zobrazovanie úľov na mape	34
	4.3 T	Testovanie	35
	4.3.1	Testovanie lokálnej databázy	35
	4.3.2	Testovanie funkčnosti máp	35

## 1 Modul – Arduino

Táto kapitola obsahuje analýzu, návrh, implementáciu a testovanie spojené s modulom Arduino a technológiou Sigfox.

## 1.1 Analýza

## 1.1.1 Analýza zvuku včiel

Na základe zvuku včiel, včelár môže detegovať, Či sú včely v poriadku alebo sa s nimi niečo deje. Včely nemajú sluchové orgány, ale majú 5 očí a výborný čuch. Včelie úle vydavájú zvuky na frekvenciach 20 až 10 000Hz.

Rojenie včiel nastáva vtedy, keď mladušky prestavajú kŕmiť matku. Matka následne prestane klásť vajíčka. Mladušky, ktoré stratili možnosť zbaviť sa materskej kašičky, sú znepokojené a vydávajú intenzívnejší zvuk mávaním krídlami pásme 200-300Hz. Sila tohto zvuku závisí od počtu takýchto mladušiek, ktoré nemôžu kŕmiť larvy. Stúpnutím hlasitosti o 3 dB znamená zdvojnásobenie sily zvuku.

- Včely v kľudovom stave okolo 200 Hz
- 21 dní pred vyrojením narastá intenzita zvuku okolo 240 Hz
- 8 dní pre rojením je intenzita zvuku 270 Hz
- 1 deň pred rojením je intenzita zvuku 300 Hz

Včelstvo taktiež vydáva špeciálne zvuky, keď príde o matku. Včelí v rozsahu polhodiny až dve hodiny bude vydávať zvuk okolo 300 - 400 Hz najčastejšie 350 Hz. Následne sila tónu včelstva bez matky bude po 4-5 hodinách klesať.

#### Dostupné mikrofóny pre Arduino

#### **MAX4466**

Tento modul ma nastaviteľný zisk, ktorý ovládame pomocou malého otočného potenciometra. K dispozícii je pin Vcc, GND pin a analógový pin. Analógový pin vysiela krivku, kde "0" je Vcc/2 a amplitúda závisí od zisku a hlasitosti zvuku. Čip MAX4466 je operačný zosilňovač, ktorý je špeciálne optimalizovaný na použitie ako zosilňovač mikrofónu

## HXJ-17

Senzor má otáčkový potenciometer, žiadny zosilňovač a čip LM393. K dispozícii je pin Vcc, GND pin, analógový pin a digitálny pin. Vzhľadom k tomu, že tento modul nemá zosilňovač, je dobrý iba na snímanie hlasitých zvukov, ako sú napríklad tlieskanie a klepanie. Prítomnosť LM393 nám umožňuje nastaviť prahovú hodnotu, takže doska môže generovať digitálny výstup pre detegovanie prekročenia prahovej hodnoty. Niektorí predajcovia propagujú tento modul ako zosilňovač LM393, ale podrobná analýza ukazuje, že analógový pin nie je zosilnený.

## LM393

Tento je podobný ako HXJ-17, ale ešte jednoduchší. Má jeden otáčací potenciometer a žiadny analógový výstup. Vhodný na detegovanie hlasitého zvuku.

## 1.1.2 Analýza posielania správ - Update

Pre odoslanie údajov nameraných zo senzorov je dostupná správa s maximálnou veľkosťou 12 znakov respektíve bytov. Pre posielanie celých čísel je potrebné minimálne 24 bytov. Preto je potrebné navrhnúť spôsob pre efektívne využitie maximálnej veľkosti posielanej správy.

## Rozsah hodnôt senzorov

- Vlhkosť <0, 100>
- Teplota <-50, 80>
- Stav batérie <0, 100>
- Hmotnosť <0, 100>
- Poloha úľa <0, 1>

## Hodnoty so senzorov zakódujeme do hexadecimálneho tvaru.

- Pridelenie bit-ov hodnotám(Dostupných 12 znakov)
- Vlhkosť 2 x 7 bit-ov
- Teplota 2x 8 bit-ov
- Hmotnosť 7 bit-ov
- Stav batérie 7 bit-ov
- Poloha úľa 1 bit

## Hodnoty indikujúce prevrátenie úľa:

- $\mathbf{0} \mathbf{U}\mathbf{l}$  je v normálnej polohe
- 1 Úľ je prevrátený

## 1.1.3 Analýza úspory batérie

Naše Arduino je napájané z batérie, preto je potrebné uvažovať o zabezpečení čo najdlhšej prevádzke na čo najmenších batériách. Spotrebu je možné na Arduine znížiť programovo, kde si pri nečinnosti uspíme mikroprocesor Arduino. Uspanie sa nie vždy hodí, hlavne pri získavaní dát pri zmene prípadne neustáleho získavania dát. V našom prípade dáta získavame v pravidelných intervaloch a tie odosielame pomocou SigFox.

Pre uspanie Arduina použijeme knižnicu *avr/sleep.h* a pomocou funkcie *sleep\_mode()* zariadenie uspíme. Knižnica nie je potrebné sťahovať. Je možné použiť niekoľko režimov uspania Arduina. Jednotlivé módy sú zobrazené v tab.

Módy	
SLEEP_MODE_IDLE	5 (najmenej úsporný)
SLEEP_MODE_ADC	4
SLEEP_MODE_PWR_SAVE	3
SLEEP_MODE_STANBY	2
SLEEP_MODE_PWR_DOWN	1 (najúspornejší)

Tabuľka 1: Režimy spánku

Režim spánku nastavíme pomocou funkcie *set\_sleep\_mode(mode)* a následne pomocou funkcie *sleep\_enable()* zaistíme, že režim spánku bude dostupný. Následne po uspaní Arduina je ho potrebné opäť zobudiť. Zobudiť Arduino je možné niekoľkými spôsobmi:

- Pomocou externého prerušenia
- Pomocou Serial UART
- Pomocou INTERNAL TIMER-u
- Pomocou WATCHDOG TIMER-u

Aby sme dosiali čo najväčšiu výdrž v našom projekte použijeme ten najúspornejší režim spánku. Ak použijeme tento režim spánku, tak nie je možné použiť všetky spôsoby prebudenia mikroprocesor. V našom prípade použijeme WATCHDOG TIMER, keďže chceme, aby naše zariadenie bolo úplné nezávisle od používateľa. WATCHDOG TIMER je možne nastaviť na prebudenie sa najviac každých osem sekúnd. Vždy po prebudení skontroluje či už je čas niečo vykonať, ak nie Arduino sa opäť uspí. V našom prípade budeme musieť spúšťať častejšie akcelerometer, aby sme identifikovali pohyb úľa a ďalšie senzory iba v pravidelných intervaloch. WATCHDOG TIMER pokračuje aj keď sa program vykonáva, to znamená, že ak sa pri spustení každých 16s vykonáva program 10s, tak Arduino bude spať 6s.

## 1.2 Návrh

Po diskusií sme sa zhodli na premiestnení bitu indikujúceho polohu úľa zo 47 bitu na 44 bit. Hodnota je v binárnej sústave. Taktiež sa hodnota nevyužitých bitov 45, 46, 47 zmení na statickú hodnotu 1 čím sa zabezpečí, aby SigFox Cloud preposlal 12 znakový reťazec ako String.

## Vzorové dáta

Bit	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bin	1	1	1	1	0	1	0	1	1	0	0	1	0	1	0	1
Hex		I	7			4	5			ç	)			4	5	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bin	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	1
Hex		(	)			4	2			Ι	ĨŦ.			ç	)	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bin	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hex		(	)			(	)			(	)			(	)	

Tabuľka 2 Rozloženie bitov správy

Tabuľka 3 Legenda k tabuľke rozloženia bitov správy

Farba	Rozsah bitov	Senzor
	0-7	Teplota vonku
	8-15	Teplota vnútri
	16-22	Vlhkosť vonku
	23-29	Vlhkosť vnútri
	30-36	Stav batérie
	37-43	Hmotnosť
	44	Poloha úľa
	45-47	Bity s hodnotu 1

Výsledná správa: F59502F90000

## Kódovanie a dekódovanie teploty:

Keďže senzor na meranie teploty mieria aj mínusové teploty je potrebné tieto hodnoty reprezentovať vhodným spôsobom. Pre zakódovanie tejto hodnoty máme k dispozícii 8 bitov. Na meraný rozsah teplôt senzora od -50 do +100, nám postačuje 7 bitov. 8 bit reprezentuje kladnú alebo zápornú hodnotu:

- 0 Kladná hodnota
- 1 Záporná hodnota

## 1.3 Implementácia

## 1.3.1 Implementácia merania váhy

Pre meranie váhy používame štandardy A/D prevodník HX711 a knižnicu #include "HX711.h". Pre získanie váhy je potrebné identifikovať kalibračný faktor pre získanie hodnoty váhy v kilogramoch.

Kalibračný kód pri spustení načíta zero\_factor, ktorý je potrebný pre meranie váhy bez nutnosti váhu zo zariadenia odoberať pri zapnutí zariadenia. Pri kalibračnom kóde nesmie byť na váhe žiadna hmotnosť pri zapnutí. Po zapnutí a prečítaní zero\_factor hodnoty sa začne čítanie hodnôt z váhy. V tomto kroku položíme na váhu predmet známej váhy. Pomocou Monitoru seriového portu, je možné meniť pomocou znakov + a – hodnotu kalibračného zariadenia. Kalibrácia spočíva v nastavení kalibračného faktoru, až kým sa nezobrazí požadovaná váha daného predmetu.

Po identifikovaní a nastavení kalibračného faktora môžeme získať váhu zo zariadenia pomocou weight = scale.get\_units();.

Weight
2.6.4
5.0 Kg
WeightWeight

## 1.3.2 Implementácia posielania hodnôt po zmene

Posielanie nameraných hodnôt bolo optimalizované, kvôli zníženiu počtu správ odosielaných na SigFox. Dáta sa posielajú každých 10 minút, avšak iba vtedy, ak dôjde k zmene nameraných údajov na zariadení. Porovnáva sa aktuálna správa na odoslanie s predchádzajúcou odoslanou správou.

```
//sending the message
if (!String(finalMessage).equals(String(lastMessage)))
{
    Serial.println(finalMessage);
    Serial3.print("AT$SF=");
    Serial3.println(finalMessage);
}
```

#### 1.3.3 Implementácia detegovania hodnôt mimo rozsahu

Pri meraní teploty a vlhkosti vonkajšej aj vnútornej sa overuje, či nameraná hodnota nie je mimo rozsahu meracieho zariadenia. Ak je nameraná hodnota mimo rozsahu, vloží sa na miesto nameranej hodnoty, číslo 127(v binárnej sústave 0111 1111), ktoré je mimo meraného rozsahu. Číslo 127 označuje zlú hodnotu pre vonkajšiu/vnútornú teplotu aj vlhkosť. V prípade zlej hodnoty pri stave batérie nie je potrebné ošetrovať, keďže takýto stav nenastane.

```
//checking measuremts from DHT22s
if((humidity < 0) || (humidity > 100)){
   humidity = 127;
}
if((temperature < -50) || (temperature > 80)){
   temperature = 127;
}
if((humidity2 < 0) || (humidity2 > 100)){
   humidity2 = 127;
}
if((temperature2 < -50) || (temperature2 > 80)){
   temperature2 = 127;
}
```

#### 1.3.4 Implementácia autokalibrácie akcelerometra

Pre správnu funkčnosť zariadenia sme implementovali automatickú kalibráciu pri každom spustení zariadenia. Kalibrácia prebieha kvôli prvotnému nastaveniu úľa v prípade žeby úľ nebol umiestený vo vodorovnej polohe. Kalibrácia vypočíta kalibračné offsety pre jednotlivé osi x, y a z, ktoré následne použije.

mpu.setXAccelOffset(ax\_offset); mpu.setYAccelOffset(ay\_offset); mpu.setZAccelOffset(az\_offset);

Obrázok 1 Nastavenie kalibračných offsetov.

## 1.3.5 Implementácia LED diód

Pre zobrazenie funkčnosti zariadenia sa na zariadenie namontovali 3 LED diódy. LED diódy majú označenia **SN(modrá LED dióda)**, ktorá indikuje odoslanie správy, **CL(Červená LED dióda)**, ktorá indikuje kalibráciu zariadenia a **ON(Zelená LED dióda)**, ktorá indikuje zapnuté zariadenie.

CL LED dióda svieti pri zapnutí zariadenia, kedy prebieha kalibrácia zariadenia. Po kalibrácii CL dióda zhasne. Následne zariadenie nameria hodnoty a odošle na SigFox server, pričom blikne SN LED dióda. ON LED dióda sa zapne po kalibrácií a vždy pri aktivite(meranie hodnôt, odosielanie) zhasne.

## 1.4 Testovanie

Táto kapitola opisuje jednotlivé testy funkčných modulov systému. V každom teste je podrobne popísaná akcia a porovnaná reakcia testera s očakávanou reakciou.

## 1.4.1 Testovanie hmotnosti

I D 1 Názov: :		Názov:	Otestovanie merania hmotnosti				
Úroveň splnenia testu:			Musí	Tester:	Peter Štofaňák		
Rozhranie:			Systém/používateľ	Systém/používateľ			
Úč	el:		Overenie funkčnosti merania s	správnej hn	notnosti		
Vs	tupné po	odmienky:	Arduino, váha a ďalší HW				
Výstupné podmienky:			Overenie funkčnosti prebehlo	v poriadku			
K ro k	Akcia		Očakávaná reakcia	Skutočná reakcia			
1	Pou	žívateľ zapojí váhu do Arduina	Správne zapojenie antény	Správne	zapojenie antény		
2	Použív	ateľ pripojí Arduino k PC	Správne pripojenie Arduina k PC	Správne pripojenie Arduin PC			
3	3 Používateľ spustí "Monitor Sériového portu"		Spustenie "Monitor Sériového portu"	Spustenie "Monitor Sériovo portu"			
4	4 Používateľ položí predmet konkrétnej hmotnosti na váhu		Používateľ položí predmet na váhu	Používateľ položil predmet váhu			
<ul> <li>Používateľ overí nameranú</li> <li>hmotnosť predmetu na PC a na mobilnom zariadení</li> </ul>		žívateľ overí nameranú osť predmetu na PC a na nobilnom zariadení	Používateľ overí nameranú hodnotu na PC aj na mobilnom zariadení	Používateľ overil nameran hodnotu na PC aj na mobilnom zariadení			

## 1.4.2 Testovanie posielania správy iba po zmene

ID:	2	Názov:	Otestovanie posielania s	práv pri z	mene	
Úro	veň spl	nenia testu:	Musí	Tester:	Jozef Vaľko	
Roz	hranie		Systém/používateľ			
Úče	l:		Overenie funkčnosti posielania správ po zmene stavu			
Vstu	ıpné po	odmienky:	Arduino, váha, senzory a	ďalší HW	V	
Výs	tupné p	oodmienky:	Overenie funkčnosti preb	ehlo v po	riadku	
Kr ok		Akcia	Očakávaná reakcia	Skuto	očná reakcia	
1	Ро	užívateľ zapojí anténu do Arduina	Správne zapojenie antény	Správ	ne zapojenie antény	
2	]	Používateľ pripojí Arduino k PC	Správne pripojenie Arduina k PC	Správ Arc	ne pripojenie luina k PC	
3	Ροι	užívateľ spustí "Monitor Sériového portu"	Spustenie "Monitor Sériového portu"	Spuste Sério	enie "Monitor ového portu"	
4	Ροι	ižívateľ počká kým sa vytvorí prvá správa	Používateľ čaká	Použ	žívateľ čaká	
5	Pou	žívateľ počká kým sa vytvorí druhá správa	Používateľ čaká	Použ	źívateľ čaká	
6	Ροι	ıžívateľ overí, či sa správy rovnajú	Porovnanie oboch správ	Porovna	nie oboch správ	
6a	Ak sa	správy rovnajú používateľ overí, či sa druhá správa odoslala.	Overenie neodoslania druhej správy	Overen dru	ie neodoslania hej správy	
6b	Ak sa	správy nerovnajú používateľ overí či sa odoslali obe.	Overenie odoslania oboch správ	Overe ob	nie odoslania och správ	

## 1.4.3 Počiatočné testovanie A/D prevodníka HX711

Overil som prepojovacie káble meracím zariadením Mastech MS8268. Pre overenie funkčnosti A/D prevodníka som postupoval nasledovne:

- Vytvoril som môstikové zapojenie, ktoré je znázornené na obrázku nižšie(pre sledovanie malých zmien v jednotlivých odporoch)
- Použil som tri 100k ohmové rezistory a jeden 100k ohmový potenciometer
- Ak bude mať potenciometer inú hodnotu ako ostatné rezistory, môžeme merať rozdiel na mostíku(ak bude mať rovnakú hodnotu, napätie bude nulové na pinoch A+, A-)

Senzor po zapojení do HX711 nejavil žiadne známky funkčnosti.



Obrázok 2 Počiatočné testovanie A/D prevodníka HX711<sup>1</sup>

## 1.4.4 Testovanie LED diód

ID 2 Názov:		Názov:	Otestovanie LED diód				
Úre	oveň splno	enia testu:	Musí <b>Tester:</b> Jozef Val				
Roz	zhranie:		Systém/používateľ				
Úče	el:		Overenie funkčnosti správneho indikovania stavu meracieho zariadenia				
Vst	upné pod	mienky:	Arduino, váha a ďalší HW				
Výs	stupné po	dmienky:	Overenie funkčnosti prebehlo v poriadku				
K ro k		Akcia	Očakávaná reakcia	Skut	očná reakcia		
1	Р	oužívateľ zapne Arduino	Používateľ zapol zariadenie	Pou: z	žívateľ zapol zariadenie		
2	Používa	teľ skontroluje, či zasvietila CL dióda	Používateľ skontroloval stav CL diódy	Používa sta	tteľ skontroloval v CL diódy		
3	Použív	vateľ overí, či zhasla CL dióda a blikla dióda SN	Používateľ skontroloval stav CL a SN diód	Používa stav	iteľ skontroloval CL a SN diód		
4	Používat	eľ overí, či po bliknutí SN diódy zasvietila dióda ON	Používateľ skontroloval stav SN a ON diódy	Používa stav S	ttel' skontroloval SN a ON diódy		
5	Použív a ON di	vateľ overí, či SN dióda blikne óda zhasne pri odoslaní nových hodnôt	Používateľ opätovne skontroluje stav SN a ON diód	Použí skontrol	vateľ opätovne uje stav SN a ON diód		

 $<sup>^{1}\</sup> http://navody.arduino-shop.cz/navody-k-produktum/ad-prevodnik-24-bit-s-2-kanaly.html$ 

## 2 Modul – Server

Táto kapitola obsahuje analýzu, návrh, implementáciu a testovanie spojené so serverom.

## 2.1 Analýza

## 2.1.1 Analýza SigFox Callback - update

Implementované posielanie POST request na špeciálnu URL, ktorá umožňuje špecifikovať parametre. Formát URL:

http://team20-

 $\frac{17.studenti.fiit.stuba.sk/BeeWebpage/public/sigfox?data[value]={data}&id={device}&time={time}&lat}{at={lat}&lng}&rssi={rssi}$ 

## Posielané parametre:

Atribút	Opis
data	Používateľské dáta v hexa tvare
device	Jedinečný identifikátor zariadenia
time	Časová pečiatka udalosti
lat	Zemepisná šírka
lng	Zemepisná dĺžka
rssi	RSSI – sila signálu

## 2.2 Implementácia

Táto kapitola opisuje implementáciu API volaní na našom serveri.

## 2.2.1 Ukladanie nameraných údajov

Sigfox server posiela na náš server namerané údaje v podobe textového reťazca s ciframi v hexa tvare, nameraný čas a idetifikačne číslo používateľa. V prvom kroku transformujeme získaný reťazec na pole nameraných údajov. Následne skontrolujeme získané údaje s hraničnými hodnotami a v prípade prekročenia hranice inicializujeme a následne pošleme používateľovi notifikáciu. V poslednom kroku uložíme záznam do tabuľky measurements.

Pre získanie notifikácií sa vytvorili nasledujúce API volania:

1. Pre získanie aktuálneho záznamu zariadenia

Cesta: user/measurements/actual

## Request

Typ: POST Headers: Content-type: application-json Body: { "user\_id": "45",

"token": "a",

"device\_id": "36B7B7"

}

2. Pre získanie požadovaného počtu meraní

Cesta: user/measurements

## Request

```
Typ: POST
```

Headers: Content-type: application-json

Body: {

"user\_id": "45", "token": "a", "device\_id": "36B7B7", "from" : "2018-04-12 17:34:26" "to" : "2018-04-13 09:46:31"

}

## 3 Modul – Webová stránka produktu

Táto kapitola obsahuje analýzu, návrh, implementáciu a testovanie spojené s webovou stránkou produktu.

## 3.1 Návrh

V nasledujúcej časti je zobrazená základná vizuálna predstava, ako by mala vyzerať stránka pre včelárov. Na jej základe bude postavená následná implementácia.

## 3.1.1 Podstránka kontakt

Účelom tejto podstránky je umožniť používateľom kontaktovať vývojárov produktu. Pre zobrazenie podstránky kontakt nie je potrebné mať žiadne prihlasovacie údaje. Stačí ísť na adresu <u>https://team20-17.studenti.fiit.stuba.sk/BeeWebpage/public</u> a kliknúť na sekciu kontakt. Po tejto interakcii sa zobrazí, ktorý je možné vidieť na obrázku Obrázok 3.

## Kontaktujte nás



Obrázok 3 Kontaktná podstránka

Táto podstránka zobrazuje mapu, kde sa nachádza sídlo vývojárov produktu a taktiež formulár, ktorý je potrebné vypísať v prípade, že chce používateľ vývojárov kontaktovať. Tento formulár sa skladá z nasledujúcich polí:

- Meno meno osoby, ktorá vývojárov kontaktuje.
- E-mail adresa, na ktorú môžu vývojári odpovedať.
- Telefón telefónne číslo v prípade odpovede.
- Správa správa používateľa vývojárom.

Po stlačení tlačidla odoslať sa všetky polia prekontrolujú a následne sa odošlú na e-mail vývojárov respektíve administrátora.

## 3.1.2 Podstránka vytvorenia objednávky

Účelom tejto podstránky je umožniť zákazníkovi vytvoriť si vlastný úľ. Pre úspešné zobrazenie podstránky je potrebné byť prihlásený ako klasický používateľ a v ľavom menu kliknúť na sekciu vytvorenie objednávky. Následne sa zobrazí objednávací formulár, ktorý obsahuje tieto polia:

- Meno úľa názov úľa, ktorý sa bude používateľovi zobrazovať v manažmente úľov.
- Adresa úl'a adresa, kde sa úl' nachádza.
- Checkboxy SMS a E-MAIL spôsob, ktorým chceme dostávať notifikácie.
- Iné požiadavky iné nadštandardné požiadavky, ktoré by chcel zákazník.

Objednávací formulár	
Meno úľa	
Adresa úľa	
Notifikácie chcem dostávať na:	
Požiadavky	
	.:
✓ Odoslať	

## Obrázok 4 Príklad objednávacieho formulára

Na obrázku Obrázok 4 je možné vidieť príklad objednávacieho formulára, ktorý je vyššie popísaný. Po jeho vypísaní sa najprv skontrolujú jednotlivé polia a až potom je formulár odoslaný administrátorovi na potvrdenie. Po jeho úspešnom potvrdení sa v sekcii manažment objednávok vytvorí nový úľ, ktorý si používateľ zadefinoval.

## 3.1.3 Podstránka manažmentu objednávok

Uvedená podstránka je prístupná len administrátorom portálu. Po prihlásení je potrebné kliknúť v ľavom menu na manažment objednávok, čoho výsledok bude prehľadné zobrazenie objednávok vo forme tabuľky. Každá objednávka obsahuje meno zákazníka, meno úľa, adresu úľa, poznámky a stav, čiže či bola objednávka potvrdená alebo nie. Funkcionalita objednávky je postavená na princípe, pri ktorom si zákazník vo svojom portáli vypíše a odošle objednávku, ktorá sa zaeviduje do databázy a teda sa zobrazí

administrátorovi v manažmente objednávok a ten má právu ju buď potvrdiť alebo zmazať. Po potvrdení sa používateľovi vytvorí úľ s menom a adresou, ktorú pri tvorbe objednávky zadal.

Manažmen objednávok	t c				
Meno zákazníka	Meno úľa	Adresa úľa	Poznámky	Stav	
Jozef Vaľko	Záhradný úľ	Záhradná 8, Prešov	Ďakujem za vaše služby.	$\checkmark$	Zmazať
Michal Kovac	Modrý úľ	Ostrova 15, Bratislava	Žiadne	$\checkmark$	Zmazať
Testovací používateľ	Sivý úlik	Zlatá 125, Detva	Žiadne	$\checkmark$	Zmazať

Obrázok 5 Príklad zobrazenia objednávok

Na obrázku Obrázok 5 je možné vidieť prehľadné zobrazenie jednotlivých objednávok, ktoré bolo vyššie popísané. V prípade, že objednávka ešte nie je potvrdená, tak sa v stĺpci stav nezobrazí uvedený obrázok, ale tlačidlo potvrdiť, ktoré môžete vidieť na obrázku Obrázok 6.

Testovací používateľ	Zombov ulik	MLynska dolina	ASAP	Potvrdiť	Zmazať

Obrázok 6 Príklad nepotvrdenej objednávky

## 3.2 Implementácia

Táto kapitola opisuje implementáciu jednotlivých častí webovej stránky produktu.

## 3.2.1 Architektúra webovej stránky

Základ funkcionality je založený na MVC dizajne, ktorý je zrealizovaný pomocou Slim3 frameworku. Výhodou Slim 3 je jeho minimalistická architektúra, ktorá slúži hlavne pri budovaní prostriedku na správu API volaní. V našom prípade tento framework slúži ako platforma na zhotovenie webovej stránky produktu, ale taktiež na správu údajov z meraní a poskytnutie potrebných API volaní, ktoré sú využívané mobilnou aplikáciou.



Obrázok 7 Základný tok požiadavok pri MVC

Jadro aplikácie tvoria cesty, ktoré na základe HTTP požiadavky vytvoria požadovanú odpoveď. Prvým krokom je konfigurácia aplikácie. V tomto kroku sme nastavili základne nastavenia aplikácie v súbore settings.php a definovali kontajner, v ktorom sme uložili cestu ku konfiguračnom súboru a odpoveď pri HTTP odpovedi 404. V konfiguračnom súbore sa nachádzajú autentifikačné údaje.

## 3.2.2 Funkcionalita vytvorenia a potvrdenia objednávky

Do webovej stránky bola pridaná funkcionalita vytvorenia objednávky používateľom a potvrdenie administrátorom. Pred pridaním tejto funkcionality ale bolo potrebné vytvoriť v databáze tabuľku "Orders", ktorá slúži na uloženie vytvorenej objednávky. Táto tabuľka obsahuje nasledujúce údaje:

- Id identifikátor, ktorý určuje, ktorý používateľ objednávku vytvoril.
- Hive\_name meno úľa.
- Hive\_address adresa úľa.
- Notes ďalšie poznámky.
- SMS ak používateľ zaškrtne, že chce dostávať notifikácie pomocou SMS, tak je táto položka nastavená ako true, inak je false.
- E-mail v prípade, že chce používateľ dostávať notifikácie pomocou e-mailu, je táto položka nastavená na true, inak je false.
- Verify\_order potvrdenie objednávky, pokiaľ ju administrátor potvrdí tak je true, pri vytvorení používateľom je automaticky false.

Po vytvorení tejto tabuľky bola vytvorená najprv funkcionalita na strane klasického používateľa, ktorej účelom bolo uložiť vytvorenú objednávku do databázy a následne ju zašle na administrátorsky e-mail, aby administrátor dokázal čím skôr na potvrdenie objednávky reagovať (Obrázok Obrázok 8).

Meno úľa:	Sivý úlik
Adresa úľa:	Zlatá 125, Detva
E-mail:	Nastavený
SMS:	Nenastavené
Odkaz:	Žiadne

#### Obrázok 8 Príklad e-mailu o potvrdení objednávky

Následne bola vytvorená funkcionalita na strane administrátora, kde bolo v tabuľke vytvorenú objednávku vidieť a zároveň ju bolo možné potvrdiť, čiže by sa hodnota vo verify\_order zmenila na true. V prípade, že by sa vytvorená objednávka administrátorovi portálu nepáčila, tak ju nemusel potvrdiť, ale mohol by ju vymazať a práve preto bola pridaná aj funkcionalita vymazania objednávky.

## 3.2.3 Funkcionalita mazania používateľa

Cieľom tejto funkcionality je dať možnosť administrátorovi portálu zmazať jednotlivých používateľov. Môže sa to diať napríklad v prípade, že by zákazník s nami ukončil zmluvu. Na obr. 1 je možné v pravej časti vidieť tlačidlo vymazať, ktorým sa táto funkcionalita vykonáva.



Obrázok 9 Príklad zobrazenia používateľov v administratorskom portáli

Po kliknutí na tlačidlo zmazať sa vykoná príkaz, ktorým sa z tabuľky "users" zmaže uvedený používateľ. Táto tabuľka obsahuje nasledujúce polia:

- Name meno používateľa.
- Email email používateľa.
- Id identifikátor.
- Role\_id rola či sa jedná o administrátorsky účet alebo klasický.
- Password\_hash heslo po výpočte s kryptografickou funkciou.
- Password\_salt -časť hesla pridávaná ku kryptografickej funkcii.
- Phone telefónne číslo používateľa.

Pre úspešné zmazanie používateľa je potrebný len identifikátor, ktorým sa určí o akého používateľa sa jedná.

## 3.2.4 Notifikácia používateľa

V prípade ak po prijatí nameraných údajov konkrétneho zariadenia používateľa nastane prekročenie vopred nastavených hraničných hodnôt, bude používateľ upozornený notifikáciou. Po zaregistrovaní nového zariadenia budú nastavené začiatočné hraničné hodnoty pre každý údaj. V prípade potreby si môže používateľ nastaviť hraničné hodnoty podľa jeho vlastnej vôle.

stĺpec	typ
id	integer
title_text	Character varying(30)
body_text	text
hive_id	integer
hive_name	text
user_id	integer
time	timestamp
seen	boolean
token	Character varying(150)

Na ukladanie notifikácií bola vytvorená v databáze tabuľka notifications.

Kontrola notifikácií prebieha po prijatí nového záznamu o meraní. V prípade prekročenia hodnoty sa následne vygenerujú notifikačné správy, ktoré sa uložia do tabuľky. V tomto kroku sme museli ošetriť problém, ktorý bol zapríčinený opakovaným posielaním rovnakého záznamu zo strany Sigfox. Tento problém sme ošetrili pridaním stĺpca token, ktorý predstavuje unikátny kľúč zložený z času záznamu merania(time), identifikačného čísla zariadenia(hive\_id), identifikačného čísla používateľa(user\_id) a nadpisu správu(title\_text). Vďaka tomuto stĺpcu, ktorý je unikátny je zabezpečené že každá notifikácia sa nachádza v databáze iba raz. Po vygenerovaní správy notifikácie sa pošle dané upozornenie na Firebase server.



Obrázok 10 Notifikácie vo webovej aplikácii

Prihlásený používateľ má možnosť vidieť počet neprečítaných notifikácií, ako aj detail každej z nich. Po zobrazení detailu notifikácie sa farebne odlíši daná notifikácia od tých, ktoré používateľ nevidel. Všetky notifikácie môže používateľ vidieť v samostatnej podstránke, kde môže vidieť všetky jeho notifikácie.

Zobraz 10 V záznamov			
Čas	Upozornenie	Správa	Včelí úľ
2018-04-10 20:54:00	Hodnota vlhkosti	Hodnota vlhkosti dnu má hodnotu 23	Úlik pri Dunaji
2018-04-10 20:54:00	Hodnota teploty	Hodnota teploty von má hodnotu 24°C	Úlik pri Dunaji
2018-04-10 20:54:00	Úľ sa prevrátil		Úlik pri Dunaji
2018-04-10 21:04:02	Hodnota vlhkosti	Hodnota vlhkosti dnu má hodnotu 23	Úlik pri Dunaji
2018-04-10 21:04:02	Hodnota teploty	Hodnota teploty von má hodnotu 24°C	Úlik pri Dunaji
2018-04-10 21:04:02	Úľ sa prevrátil		Úlik pri Dunaji
2018-04-10 21:14:04	Hodnota teploty	Hodnota teploty von má hodnotu 24°C	Úlik pri Dunaji
2018-04-10 21:14:04	Úľ sa prevrátil		Úlik pri Dunaji
2018-04-11 22:17:17	Hodnota vlhkosti	Hodnota vlhkosti von má hodnotu 37	Úlik v Prešove
2018-04-11 22:17:17	Hodnota teploty	Hodnota teploty dnu má hodnotu 41°C	Úlik v Prešove
2018-04-11 22:17:17           Záznamy 1 až 10 z celkom 29           Predchádzajúca         1         2         3	Hodnota teploty Nasledujúca	Hodnota teploty dnu ma nodnotu 41°C	UIIK V Prešove

Obrázok 11 História notikácií vo webovej aplikácii

## 3.2.5 Profil používateľa

Každý používateľ má po registrácií na stránke k dispozícií vlastný profil, ktorý obsahuje jeho osobné údaje zadané pri registrácií s možnosťou ich zmeniť.

<b>a</b> 🔎		Testovací používateľ
Profil		
Meno	Testovací používateľ	
Email	test@test.test	
Telefónne číslo	+421 123 456 800	
Password		
Confirm password		
	Uložiť zmeny	

Obrázok 12 Profil používateľa na webovej stránke

Pri zmene telefónneho čísla alebo mena sa zmeny uložia do tabuľky users. V prípade ak používateľ zmení email alebo heslo, tak sa vytvorí nový používateľ.

## 3.2.6 Prezentačná stránka produktu

Prezentačná stránka produktu poskytuje základné informácie o našom projekte. Je použitá open-source frontend knižnica Bootstrap, verzia 4.0.0.

V hornom menu sa vie používateľ prekliknúť na detailné informácie o produkte – "O produkte", kontaktovať nás – "Kontakt" a tiež zobraziť prihlasovacie okno – "Prihlásiť".



Obrázok 13 Prezentačná stránka produktu

## 3.2.7 Admin portál

Náš systém umožňuje prihlasovanie admina, ktorý má možnosť prezerať si všetkých užívateľov a ich zaznamenané merania.

## Prihlásenie sa a registrácia

Pre prihlásenie sa do systému ako admin, je potrebné prihlásiť sa s užívateľom, ktorého hodnota *role\_id* = 2. Pre vytvorenie administrátora je potrebné zaregistrovať používateľa na webovej stránke na adrese <u>http://team20-17.studenti.fiit.stuba.sk/BeeWebpage/public/register</u> a následne v databáze zmeniť jeho parameter *role\_id*.

Pre admina je vytvorené špeciálne API pre získavanie informácií. Má prístup ku všetkým informáciám, no systém kontroluje, ci je vytvorená Session s používateľom s user\_id =2:

Získanie používateľov: Cesta: /admin/users HTTP metóda: POST Získanie zariadení používateľa: Cesta: /admin/devices HTTP metóda: POST Získanie posledného merania zariadenia: Cesta: /admin/measurements/actual HTTP metóda: POST Získanie všetkých meraní zariadenia: Cesta: /admin/measurements HTTP metóda: POST

## Zobrazenie prehľadu používateľov

Po prihlásení sa má admin možnosť prezerať si zoznam používateľov s informáciou o počte zariadení. Zoznam používateľov je zoradený od najdôležitejšieho. Za najdôležitejšieho používateľa považujeme takého, ktorý ma najviac zariadení.



#### Obrázok 14 Zobrazenie informácií o používateľovi

Po kliknutí na "Zobraziť zariadenia" si môžeme prezerať informácie o úľoch používateľa. Zobrazené úle sú vyrolované pod daným používateľom. Opis zobrazenia prehľadu a detailov úľu sa nachádza v ďalších kapitolách.

## Zobrazenie prehľadu úľov

## Cesta: https://team20-17.studenti.fiit.stuba.sk/BeeWebpage/public/portal

Prehľad úľov umožňuje prezerať si úle používateľa. Je zobrazená informácia o názve úľa, jeho lokácii a tiež informácie o poslednom meraní. V prípade prekročenia hraničnej hodnoty je používateľ graficky upozornení červenou farbou danej veličiny.

Kamenný mlyn TRNAVA		
Notifiidcie SMS E-mail	₿ VNÚŤORNÁ TEPLOTA: 41°C ▲VNÚŤORNÁ VLHKOST: 24% № POHYB ÚLA: NEPREVRÁTENÝ ED BATĚMA: 100%	₿ VONKAJŠIA TEPLOTA: 25°C ▲VONKAJŠIA VLHKOSŤ: 42% ∰ VÁHA: 24KG
Zobraziť namerané údaje		
Zobraziť grafy meraní		
♀ Zobraziť mapu		

Obrázok 15 Zobrazenie informácií o úli



Obrázok 16 Diagram vytvorenia úľov

Zoradenie záznamov o úľoch

Pre zlepšenie prehľadnosti úľov, je možné ich zoradenie podľa atribútov:

- Názov úľa
- Lokácia úľa
- Hodnota l'ubovol'nej veličiny posledného merania

Zobraziť vzostupne			~	Názov	Lokácia	Posledné meranie	
	Ob	rázok 17 Možnosť zoradeni	a záznamov				
Po vybratí možnosti posledného merania sa zobrazí možnosť výberu veličiny:							
Zobraziť vzostupne	~	Vnútorná vlhkosť	~	Názov	Lokácia	Posledné meranie	

Obrázok 18 Možnosť zoradenia podľa posledného merania

## Zobrazenie detailu úľa

Po kliknutí na "Zobraziť detail úľa" sa používateľ dostane na obrazovku zobrazujúcu detail úľa.

Úlik pri jazierku	Hraničné hodnoty:							
Trnava	Vonkajšia teplota MAX	70	la Vonkajšia vlhkosť MAX	90	Vnútorná teplota MAX	50	Vnútorná vlhkosť MAX	80
	Vonkajšia teplota MIN	-25	🌢 Vonkajšia vlhkosť MIN	0	Vnútorná teplota MIN	-20	Vnútorná vlhkosť MIN	0
	E Batéria MIN 20		4 Hmotnosť MAX 50			🖺 Uložiť	≓ Obnoviť predvolené hodnot	/

Obrázok 19 Zobrazenie informácií o hraničných hodnotách

Na vrchu zobrazenej stránky sa nachádza oblasť s informáciami o danom úli. Názov úľa aj lokácia úľa sú po kliknutí na ich názov editovateľné:

# Úlik 🛛

### Obrázok 20 Editovateľný názov úľa

Vytvorené API pre editovanie názvu a lokácie úľa má nasledujúci tvar:

## Aktualizácia názvu zariadenia:

Cesta: /user/device/name HTTP metóda: PUT

## Aktualizácia lokality zariadenia:

Cesta: /user/device/location HTTP metóda: PUT

Informácie o hraničných hodnotách úľa sú uložené v databáze. Pri vytvorení úľa sú im automaticky priradené predvolené hodnoty:

temperature_in_up_limit	smallint	50
weight_limit	smallint	50
temperature_out_up_limit	smallint	70
humidity_in_up_limit	smallint	80
humidity_out_up_limit	smallint	90
batery_limit	smallint	20
temperature_in_down_limit	smallint	'-20'::integer
temperature_out_down_limit	smallint	'-25'::integer
humidity_in_down_limit	smallint	0
humidity_out_down_limit	smallint	0

Obrázok 21 Hraničné hodnoty úľa v databáze

Tieto hodnoty si môže používateľ na podstránke detailu úľa upravovať. Po upravení hodnôt a stlačení tlačidla "Uložiť" sú nové hodnoty odoslané na server, kde sú skontrolované, či spadajú do určených limitov.

Limity pre hraničné hodnoty sú nasledovné:

	Minimálna hodnota	Maximálna hodnota
Vonkajšia teplota	-50	100
Vnútorná teplota	-50	100
Vonkajšia vlhkosť	0	100
Vnútorná vlhkosť	0	100
Váha	0	100
Batéria	0	50

Navyše musí platiť, že spodná hraničná hodnota danej veličiny je menšie ako vrchná hodnota veličiny.

Po úspešnej aktualizácii hodnôt sa zobrazí okno o úspešnosti aktualizácie hraničných hodnôt:

Hraničné hodnoty boli úspešne zmenené

Obrázok 22 Oznámenie o úspešnej aktualizácii hraničných hodnôt

×

Predchádzajúca 1 2 3 Nasledujúca

Pri nesprávnom zadaní hodnôt je vrátená chyba spolu s presnou informáciou o chybných poliach:

Zmena neúspešná! Vnútorná teplota musí byť hodnota z intervalu <-50, 100>. Vonkajšia teplota musí byť hodnota z intervalu <-50, 100>.

Obrázok 23 Hláška o neúspešnej aktualizácii hzraničných hodnôt

Používateľ má tiež možnosť hraničné hodnoty resetovať na ich pôvodné hodnoty kliknutím na tlačidlo "Obnoviť prevdolené hodnoty". V takomto pripadne sú priradené predvolené hraničné hodnoty, určené v databáze.

### Nastavenie limitov zariadenia:

Cesta: /user/device/limits HTTP metóda: PUT

#### **Resetovanie limitov zariadenia:**

Cesta: /user/device/limits /reset HTTP metóda: PUT

#### Zobrazenie hodnôt meraní:

Pod informáciami úľa je zobrazená tabuľka s nameranými údajmi. Tieto údaje sú zobrazené podľa dátumu od najnovšieho. Tabuľka umožňuje zoradiť si tieto údaje podľa ľubovoľného atribútu.

Na vytvorenie tabuľky bola použitá knižnica Datatables a jQuery.

Pre knižnicu Datatables sú dostupné pluginy s jazykmi implementácie. Preto používame plugin so slovenským jazykom, dostupnom *na cdn.datatables.net/plug-ins/1.10.16/i18n/Slovak.json*.

Čas 14	Vnútorná teplota	Vonkajšia teplota	Vnútorná vlhkost	Vonkajšia vlhkost	Prevrátenie 14	Hmotnosť 11	Batéria 🕕
2018-04-18 00:53:20	41°C	25°C	24%	42%	Neprevrátený	24 kg	100%
2018-04-11 15:11:40	36°C	15°C	80%	30%	Prevrátený	52 kg	79%
2018-04-11 15:01:40	31°C	24°C	30%	44%	Neprevrátený	127 kg	17%
2018-04-11 15:01:35	31°C	24°C	30%	44%	Neprevrátený	127 kg	17%
2018-04-11 15:01:30	19°C	19°C	51%	50%	Prevrátený	0 kg	17%
2018-04-11 15:01:21	19°C	19°C	51%	50%	Prevrátený	0 kg	17%
2018-03-30 15:42:06	20°C	15°C	85%	75%	Prevrátený	51 kg	80%
2018-03-30 15:32:30	19°C	19°C	51%	50%	Prevrátený	0 kg	17%
2018-03-29 23:17:45	20°C	19°C	48%	49%	Neprevrátený	0 kg	18%
2018-03-29 23:14:58	20°C	19°C	48%	49%	Neprevrátený	0 kg	18%
Záznamy 1 až 10 z celkom 25							

Obrázok 24 Tabuľka s nameranými hodnotami

## 3.2.8 Funkcionalita zobrazenia mapy

Táto funkcionalita je prístupná pre používateľa s bežným účtom, ale taktiež aj pre administrátora. Jej cieľom je zobraziť úľ používateľa v mape, kde si ju používateľ môže taktiež približovať, odďaľovať už podľa vlastného uváženia.



Obrázok 25 Zobrazenie mapy úľa

Ako je možné vidieť na obrázku, tak sa používateľovi po kliknutí na zobrazenie mapy ukáže v ľavom hornom rohu názov úľa spoločne s jeho označením na mape. Mapa sa vytvára na základe súradníc, ktoré sú uložené v databáze v tabuľke "devices". Berie sa do úvahy zemepisná dĺžka a šírka. Tieto súradnice sa do databázy dostanú zo Sigfox zariadenia, ktoré v odosielaných dátach posiela taktiež svoju polohu.

## 3.3 Testovanie

Táto kapitola opisuje jednotlivé testy funkčných modulov systému. V každom teste je podrobne popísaná akcia a porovnaná reakcia testera s očakávanou reakciou.

## 3.3.1 Testovanie vytvorenia objednávky

ID	1	Názov:	Vytvorenie objednávky			
Úro	veň	splnenia testu:	Musí	Tester	Matúš Sosňak	
Roz	hrar	iie:	Systém/používateľ			
Úče	l:		Overenie funkčnosti vytvorenia objednávky			
Vst	upné	podmienky:	Internet, PC, prihlasovacie údaje ako user			
Výs	tupn	é podmienky:	Vytvorenie objednávky prebehlo úspešne			
K	rok	Akcia	Očakávaná reakcia	Sku	točná reakcia	
1	17.s	Používateľ zadá do prehliadača adresu https://team20- studenti.fiit.stuba.sk/BeeW ebpage/public/.	Zobrazenie webovej stránky produktu	Zobrazenie webovej stránky produktu		
2		Kliknutie na tlačidlo prihlásenie	Zobrazenie prihlasovacieho formulára	Zobrazenie prihlasovacieho formulára		
3	ko	Zadanie a potvrdenie prektných prihlasovacích údajov	Presmerovanie na web stránku po prihlásení	Presmerovanie na web stránku po prihlásení		
4	Kli "	knutie na sekciu s názvom vytvorenie objednávky"	Zobrazenie podstránky vytvorenia objednávky	Zobrazenie podstránky vytvorenia objednávky		
5	Zad	anie korektných údajov do jednotlivých polí	Bez interakcie	Bez interakcie		
6	S	tlačenie tlačidla odoslať	Výpis hlášky "objednávka bola úspešne odoslaná na potvrdenie" a následne pridaný nový záznam v databáze a odoslaný e-mail v schránke administrátora	Výpis hlášky "objednávk bola úspešne odoslaná na potvrdenie" a následne pridaný nový záznam v databáze a odoslaný e-m v schránke administrátor		

ID	2	Názo	v:	Zobrazenie novej objednávky v administrátorskom portáli			
Úro	oveň	splnen	ia testu:	Musí	Tester	Matúš Sosňak	
Roz	hrar	nie:		Systém/používateľ			
Úče	el:			Overenie funkčnosti zobrazovania objednávok			
Vst	upné	podm	lienky:	Internet, PC, prihlasovacie údaje ako admin			
Výs	stupn	ié podr	mienky:	Zobrazenie novej objednávky prebehlo úspešne			
	Kro	k	Akcia	Očakávaná reakcia	Skutočná reakcia		
1	Používateľ zadá do prehliadača adresu https://team20- 17.studenti.fiit.stuba.sk/BeeWebpage/p ublic/.			Zobrazenie webovej stránky produktu	Zobrazenie webovej stránky produktu		
2	Kliknutie na tlačidlo prihlásenie			Zobrazenie prihlasovacieho formulára	Zobrazenie prihlasovacieho formulára		
3	Zadanie a potvrdenie korektných prihlasovacích údajov			Presmerovanie na web stránku po prihlásení	Presmerovanie na web stránku po prihlásení		
4	Kliknutie na sekciu s názvom "manažment objednávok"			Zobrazenie podstránky manažmentu objednávok	Zobrazenie podstránky manažmentu objednávok		
5		Nájde	nie pridanej objednávky	Objednávka je zobrazená v tabuľke a je pri nej tlačidlo "potvrdiť"	Objednávka je zobrazená v tabuľke a je pri nej tlačidlo "potvrdiť"		

## 3.3.2 Testovanie zobrazenia novej objednávky v administrátorskom portáli

## 3.3.3 Testovanie potvrdenia objednávky

ID	3 Názov:		ZOV:	Potvrdenie novej objednávky v administrátorskom portáli				
Úroveň splnenia testu:				Musí	Tester	Matúš Sosňak		
Rozhranie:				Systém/používateľ				
Úče	l:			Overenie funkčnosti potvrdzovania objednávok				
Vst	up	né p	odmienky:	Internet, PC, prihlasovacie údaje ako admin				
Výs	tuj	pné	podmienky:	Potvrdenie objednávky prebehlo úsj	Potvrdenie objednávky prebehlo úspešne			
K	rol	k	Akcia	Očakávaná reakcia	Skutočná reakcia			
1	Používateľ zadá do prehliadača adresu https://team20- 17.studenti.fiit.stuba.sk/ BeeWebpage/public/.		žívateľ zadá do hliadača adresu ttps://team20- denti.fiit.stuba.sk/ Webpage/public/.	Zobrazenie webovej stránky produktu	Zobrazenie webovej stránky produktu			
2	Kliknutie na tlačidlo prihlásenie		nutie na tlačidlo prihlásenie	Zobrazenie prihlasovacieho formulára	Zobrazenie prihlasovacieho formulára			
3	Zadanie a potvrdenie korektných prihlasovacích údajov		anie a potvrdenie korektných asovacích údajov	Presmerovanie na web stránku po prihlásení	Presmerovanie na web stránku j prihlásení			
4	Kliknutie na sekciu s názvom "manažment objednávok"		knutie na sekciu vom "manažment objednávok"	Zobrazenie podstránky manažmentu objednávok	Zobrazo manažm	enie podstránky entu objednávok		
5	Nájdenie pridanej objednávky		jdenie pridanej objednávky	Objednávka je zobrazená v tabuľke a je pri nej tlačidlo "potvrdiť"	Objednávka je zobrazená v tabuľke a je pri nej tlačidlo "potvrdiť"			
6	6 Kliknutie na tlačidlo "potvrdit"		nutie na tlačidlo "potvrdiť"	Zobrazenie hlášky "objednávka bola úspešne potvrdená" a zároveň zmena hodnoty verify_order v databáze na true a zároveň pridanie úľa v tabuľke "devices"	Zobrazenie hlášky "objednávka bola úspešne potvrdená" a zároveň zmena hodnoty verify_order v databáze na truc a zároveň pridanie úľa v tabuľk "devices"			

## 3.3.4 Testovanie kontaktnej podstránky

ID	4	Názov:		Testovanie funkčnosti kontaktnej podstránky			
Úro	oveň spl	nenia te	stu:	Musí	Tester	Matúš Sosňak	
Rozhranie:				Systém/používateľ			
Úče	el:			Overenie funkčnosti kontaktnej podstránky			
Vst	upné po	odmienk	y:	Internet, PC			
Výs	stupné p	odmien	ky:	Testovanie prebehlo úspešne			
	Kro	k	Akcia	Očakávaná reakcia	Skutočná reakcia		
1	Používateľ zadá do prehliadača adresu https://team20- 17.studenti.fiit.stuba.sk/BeeWe bpage/public/.			Zobrazenie webovej stránky produktu	Zobrazenie webovej stránky produktu		
2	Kliknutie na sekciu kontakt			Zobrazenie kontaktnej podstránky	Zobrazen poo	nie kontaktnej Istránky	
3	Zadanie korektných prihlasovacích údajov			Bez interakcie	Bez interakcie		
4	Kliknutie na tlačidlo "odoslať"			Vypísanie hlášky "Mail bol úspešne odoslaný" a zároveň odoslanie mailu administrátorovi	Vypísanie hlášky "Mail bol úspešne odoslaný" a zároveň odoslanie mailu administrátorovi		

## 4 Modul – Android

Táto kapitola obsahuje analýzu, návrh, implementáciu a testovanie spojené s Android aplikáciou.

## 4.1 Návrh mobilnej aplikácie

#### 4.1.1 Návrh zobrazenia grafov v android aplikácii

Android aplikácia bude obsahovať zobrazenie nameraných údajov v podobe grafov. Rovnako ako v textovej budú vykreslené merania vonkajšej a vnútornej teploty a vlhkosti, hmotnosti, detekcia prevrátenia úľa a stav batérie. Medzi týmito kategóriami sa bude prechádzať pomocou slide baru, ktorý sa bude nachádzať ešte nad grafom. Zobrazenie na landscape nebude obsahovať hodnoty v textovej podobe, ako je na obrázku ale iba samotný graf. Gestom priblíženia sa pohľad priblíži na danom mieste v grafe. Rovnako tak bude možné prechádzať hodnoty podľa dátumu merania.



Obrázok 26 Návrh zobrazenia grafov v android aplikácii

## 4.1.2 Profil používateľa

Obrazovka zobrazuje základné informácie, ktoré máme o používateľovi – včelárovi dostupné. Ide o jeho meno a priezvisko, zadaný email, telefónne číslo a počet registrovaných zariadení.

## 4.1.3 Návrh zobrazenia mapy v android aplikácii

Na stránku mapy sa používateľ dostane buď po rozkliknutí troch bodiek v pravom hornom rohu na obrazovke zobrazenia informácií o úli alebo z pravého bočného panelu. Používateľ by mal mať možnosť

vedieť si vyfiltrovať zobrazenie jednotlivých úľov. Na mape bude zobrazený znak úľu, ktorý bude indikovať úľ a po kliknutí na túto ikonu sa zobrazí meno úľa a jeho presná adresa – koordináty.



Obrázok 27 Návrh zobrazenia mapy v android aplikácii

## 4.2 Implementácia

Aplikácia bola implementovaná pre operačný systém Android a implementácia bola realizovaná pomocou vývojového prostredia Android Studio 3.0 od firmy JetBrains. Android aplikácia má za úlohu uľahčiť včelárovi sledovanie aktuálneho stavu úľov. Používateľ sa do aplikácie prihlási pomocou používateľského mena, čiže e-mailovej adresy a hesla. Backend aplikácie zabezpečí odoslanie týchto údajov na REST API serveru včelička. Server overí používateľ a v databáze, a ak používateľ zadal správne prihlasovacie údaje, tak umožní používateľovi vstúpiť do aplikácie. Používateľ si následne môže v aplikácií pozrieť hodnoty namerané senzormi na jeho úľoch. Dáta zo senzorov sa získavajú zo serveru včelička cez jeho REST API. Údaje sú najprv rozdelené podľa úľov. Pri každom úli v zozname používateľ vidí najaktuálnejšie hodnoty zo senzorov. Po kliknutí na konkrétny úľ sa používateľovi zobrazia historické dáta, ktoré sú rozdelené do samostatných kategórií pre teplotu, vlhkosť, hmotnosť, batériu a pozíciu úľa. Pomocou menu v pravom hornom rohu obrazovky si používateľ môže zvoliť možnosť, zobraziť súhrnné historické údaje zo senzorov.

#### 4.2.1 Profil používateľa

Obrazovka zobrazuje základné informácie, ktoré máme o používateľovi – včelárovi dostupné. Ide o jeho meno a priezvisko, zadaný email, telefónne číslo a počet registrovaných zariadení.

#### 4.2.2 Prihlasovacia stránka

Prihlasovacia stránka obsahuje textové polia na zadanie prihlasovacích údajov e-mailu a hesla. Po ich zadaní a potvrdení stlačením tlačidla Prihlásiť aplikácia najprv overí správnosť zadania e-mailu a hesla. Ak je zadaný platný e-mail, to znamená reťazec znakov, ktorý obsahuje znak "@" aj bodku, podobne či bolo vôbec zadané nejaké heslo, tak overí pripojenie k internetu. Ak je pripojenie v poriadku tak spustí metódu prihlásenia. Táto metóda zostaví HTTP požiadavku typu POST, vytvorí telo tejto požiadavky zo zadaných parametrov a odošle požiadavku na server. Server skontroluje existenciu používateľa v databáze a aj správnosť zadaného hesla a vráti buď kladnú alebo zápornú odpoveď. Ak vráti odpoveď, že všetko je v poriadku tak aplikácia zobrazí základný zoznam úľov, v opačnom prípade zobrazí chybové okienko s hláškou, že používateľ zadal chybné prihlasovacie údaje. Po prihlásení používateľ a sa tento používateľ uloží do lokálnej SQLite databázy. Do databázy sa uložia údaje, ktoré vráti server v odpovedi a to jeho osobné údaje a aj vygenerovaný token, pomocou ktorého bude používateľ vykonávať všetky ďalšie akcie. Rovnako sa nastaví aj premenná, ktorá sa použije pri nasledujúcom spustení aplikácie na priame prihlásenie používateľa. Týmto umožníme používateľovi opätovné spúšťanie aplikácie a nebude musieť stále zadávať prihlasovacie údaje. Odpoveď od servera obsahuje aj čas, kedy vyprší platnosť prihlásenia používateľa. Táto hodnota je UNIX-ový čas v sekundách a tiež sa uloží do DB k príslušnému používateľovi.

Pri opätovnom spustení aplikácie sa najprv overí, či je používateľ stále prihlásený, to znamená, že na konkrétnom zariadení sa nedávno prihlasoval do aplikácie, a ak áno, tak sa následne overí, či mu náhodou medzi časom nevypršala platnosť prihlásenia. Ak nie, tak sa mu zobrazí základný zoznam úľov. Ak sa jedna z vyššie spomenutých podmienok nesplní, tak sa používateľovi zobrazí prihlasovacia stránka na opätovné zadanie prihlasovacích údajov.

#### 4.2.3 Registračný formulár

Prihlasovacia stránka obsahuje aj tlačidlo na registráciu nového používateľa. Po kliknutí na toto tlačidlo sa zobrazí registračný formulár. Ten obsahuje meno a priezvisko, e-mail a dve polia pre heslo a jeho zopakovanie. Po vyplnení údajov o používateľovi a stačení tlačidla "Vytvoriť účet", registračná metóda skontroluje správnosť zadaných údajov. Ide hlavne o overenie zadania mena a priezviska do jedného poľa s medzerou, platnosti zadaného e-mailu, či obsahuje "@" a bodku a nakoniec dĺžku hesla, ktorá musí byť minimálne 8 znakov. Rovnako aj overuje, či sa zhodujú zadané heslá. Ak sú všetky údaje v poriadku, tak metóda ďalej vytvorí HTTP požiadavku typu POST s príslušným telom a odošle tieto dáta na server. Ten overí, či sa používateľ s daným e-mailom už nenachádza v databáze a vráti odpoveď. V

prípade negatívnej odpovede aplikácia zobrazí používateľovi chybovú hlášku, že používateľ s daným emailom už existuje a umožní mu zmeniť zadané údaje. Ak registrácia prebehne úspešne, tak aplikácia oznámi používateľovi, že registrácia bola úspešná a že sa môže prihlásiť. Aplikácia vráti používateľa automaticky na základnú prihlasovaciu stránku.

## 4.2.4 Základné zobrazenie úľov

Slúži pre zobrazenie základného prehľadu o úľoch. To zahŕňa názov úľa a posledné namerané hodnoty pre každý úľ. Taktiež obsahuje aj bočný panel s menu. V hornej časti panelu sa nachádza naše logo a odkaz na naše webové sídlo tímu naprogramované ako odkaz, ktorý po kliknutí otvorí predvolený prehliadač a načíta webovú stránku nášho tímu. Položky tohto menu sú:

- Objednávka zariadenia umožňuje používateľovi vytvoriť novú objednávku zariadenia. Po kliknutí sa zobrazí formulár tejto objednávky.
- **O projekte** zobrazí informácie o našom tímovom projekte. To zahŕňa vedúceho a členov tímu spolu s krátkym opisom projektu.
- **Profil** zobrazenie profilu používateľa, čiže údajov ktoré zadal pri registrácii.
- Notifikácie zobrazenie histórie notifikácií. Tu sa zobrazuje posledných 100 notifikácií o prekročení hraničných hodnôt.
- Odhlásenie tlačidlo na odhlásenie používateľa z aplikácie. Po kliknutí sa zobrazí okienko s otázkou, či chce používateľ naozaj pokračovať v odhlásení. Po odhlásení je používateľ vrátený na základnú prihlasovaciu stránku.

Metóda *onCreate()* slúži pre inicializáciu android aktivity. Inicializuje všetky grafické prvky aplikácie, spustí načítanie zoznamu úľov (metóda loadHiveNames) a načítanie posledných nameraných hodnôt pre každý úľ (metóda loadHiveBaseInfoServerReq).

Metóda *loadHiveBaseInfoServerReq()* načíta najnovšie hodnoty zo senzorov pre každý úľ (vnútorná/vonkajšia teplota, vlhkosť, stav batérie, hmotnosť úľa a stav akcelerometra). Dáta sa načítavajú cez REST API serveru Včelička. Na server sa v žiadosti odosiela názov úľa a token používateľa, ktorý získal počas prihlásenia. Prijaté dáta sú vo formáte JSON.

Metóda *loadHiveNames()* načíta zoznam úľov pre prihláseného používateľa. Dáta sa načítavajú cez REST API serveru Včelička. Na server sa v žiadosti odosiela identifikačné číslo používateľa a jeho token. Prijaté dáta sú vo formáte JSON.

Metóda *hiveClicked()* slúži na obsluhu udalosti po kliknutí používateľa na vybraný úľ. Otvorí android aktivitu "HiveDetailsActivity" obsahujúcu detaily pre vybraný úľ (história meraní, grafy, štatistiky atď.). Do vytvorenej aktivity posiela: názov úľa, identifikačné číslo úľa a token používateľa.

Všetky dáta zo senzorov, ktoré sa sťahujú zo servera Včelička sa lokálne ukladajú v SQLite databáze. Je to preto, aby v prípade výpadku pripojenia na internet bola aplikácia stále funkčná a schopná zobraziť aspoň poslednú úspešnú synchronizáciu so serverom. V databáze je vytvorená

tabuľka "measurements", ktorá obsahuje merania pre každý úľ. Pri prvom prihlásení používateľa sa do tejto tabuľky uložia všetky dáta zo servera. Po nasledujúcich prihláseniach sa stiahnu len dáta novšie ako už sú uložené. To sa dosiahne pomocou API na serveri, do ktorého pošle Android aplikácia JSON, ktorý obsahuje polia FROM a TO. Takto sa dá veľmi pohodlne nastaviť, nech nám server vráti len novšie údaje ako už máme uložené v SQLite databáze.

### 4.2.5 Zobrazenie detailu úľa

Slúži pre zobrazenie detailov pre vybraný úl, čiže históriu meraní v podobe grafov. Taktiež obsahuje aj bočný panel s menu, podobne ako pri základnom zobrazení úlov. Zobrazenie historických dát pre konkrétny úl je rozdelené do 5 kariet, pre teplotu, vlhkosť, hmotnosť, batériu a pozíciu. Každá z týchto kariet okrem karty Pozícia obsahuje v hornej časti graf zvolenej veličiny a pod týmto grafom sa nachádza zoznam historických meraní. Tento zoznam je zoradený tak, že v hornej časti sú najnovšie dáta a postupne sa vieme prescrollovať k starším. Graf zobrazuje namerané hodnoty graficky pomocou čiarového grafu. Pri teplote a vlhkosti sú 2 čiary, keďže meriame vnútornú aj vonkajšiu hodnotu týchto veličín. Ak graf obsahuje 2 čiary, tieto sú od seba farebne oddelené s príslušnou legendou v dolnej časti grafu. Os x obsahuje časovú značku a os y zase nameranú hodnotu v danom čase. Aplikácia je naprogramovaná tak, že v prípade otočenia zariadenia do horizontálnej polohy sa zmení aj zobrazenie tejto obrazovky, a to tak, že graf sa bude zobrazovať na celú obrazovku.

Zoznam, ktorý zobrazuje namerané dáta v minulosti je možné aktualizovať jednoduchým potiahnutím prsta nadol po obrazovke ako to poznáme z viacerých aplikácií. Vtedy sa pošle na server požiadavka s hodnotou FROM nastavenou na najnovšiu nameranú hodnotu. Prijaté dáta sú následne uložené do lokálnej databázy a samozrejme okamžite zobrazené používateľovi.

Karta Pozícia nepotrebuje grafické zobrazenie nameraných údajov, lebo táto karta obsahuje len informáciu o tom, či bol alebo nebol úľ v danom čase v základnej polohe.

## 4.2.6 Zobrazenie všetkých nameraných dát pre konkrétny úľ

Táto aktivita zobrazuje historické dáta pre konkrétny úľ. Tieto dáta vyzerajú podobne ako základný zoznam úľov po prihlásení. Rozdiel je v tom, že tuto je namiesto názvu úľa dátum a čas merania a v rámci tejto časovej značky sú zobrazené namerané údaje zo senzorov.

#### 4.2.7 Objednávka zariadenia

Bočný navigačný panel hlavnej obrazovky obsahuje aj možnosť objednávky zariadenia. Po kliknutí na túto položku sa používateľovi zobrazí formulár na vytvorenie objednávky. Tento formulár obsahuje informácie o zariadení, ktoré si chce používateľ objednať. Tuto musí vyplniť údaje ako názov zariadenia, jeho lokáciu, či chce dostávať notifikácie prostredníctvom SMS, e-mailu alebo oboch a poznámky. Následne metóda vytvorenia objednávky vytvorí HTTP požiadavku typu POST s telom,

ktoré obsahuje príslušné údaje. Telo požiadavky obsahuje aj ID používateľa a jeho token, ktorý mu server vygeneroval počas prihlásenia. Pomocou tohto tokenu a ID sa overí, či používateľ je naozaj ten za koho sa vydáva a dáta sa uložia do databázy na serveri. Objednávka je takto odoslaná na spracovanie. Poverený správca systému musí túto objednávku overiť a schváliť. Až po tomto schválení sa používateľovi zobrazí jeho novo objednané zariadenie v základnom prehľade.

## 4.2.8 O projekte

Z bočného navigačného panelu je možné zobraziť aj aktivitu O projekte. Táto aktivita zobrazuje mená všetkých členov tímu a aj hlavný popis nášho projektu ako aj poďakovanie za používanie nášho produktu.

## 4.2.9 Notifikácie

Táto aktivita používateľovi umožňuje pozrieť archív posledných 100 notifikácií. Notifikácie sa používateľovi posielajú, keď dôjde k prekročeniu nastavených hraničných hodnôt. Notifikácie prechádzajú na telefón aj keď nie je aplikácia spustená (vo forme natívnych android notifikácií). Pre spravovanie notifikácií sa využíva služba/knižnica Firebase Cloud Messaging. Používateľ nepríde o notifikácie, ktoré server včelička odoslal v čase bez internetového pripojenia alebo vypnutého telefónu/aplikácie. Používateľovi sa notifikácie zobrazia pri najbližšom pripojení k internetu, a ostanú uložené v telefóne.

## 4.2.10 Nastavenie hraničných hodnôt

Umožňuje používateľovi nastaviť, ktoré hodnoty považuje pre vybraný úľ za hraničné (po ich prekročení bude odoslaná notifikácia). Na začiatku sú pre každý úľ nastavené predvolené hraničné hodnoty. Keď používateľ stlačí tlačidlo uložiť, tak sa nové hraničné hodnoty odošlú na server včelička. Pre niektoré merané veličiny si môže používateľ zvoliť maximálnu aj minimálnu hodnotu (vnútorná/vonkajšia teplota a vlhkosť), a pre niektoré len jednu z nich (batéria – min, hmotnosť - max). Tlačidlom "Predvolené" vráti používateľ hraničné hodnoty na predvolené.

#### 4.2.11 Zobrazovanie úľov na mape

Umožňuje používateľovi zobraziť polohu svojich včelích úľov na mape (Google Maps). Používateľ sa môže v mape posúvať alebo meniť jej priblíženie (zoom). Po kliknutí na vybraný úľ sa používateľovi zobrazí informácia o jeho názve, lokalite a GPS súradniciach. V prípade, že je aktivita otvorená z detailu vybraného úľa, tak sa mape daný úľ zobrazí väčší a automaticky sa otvorí jeho popis. Pre využívanie tejto aktivity je potrebné funkčné internetové pripojenie.

## 4.3 Testovanie

Pri testovaní sme sa zamerali na otestovanie funkčnosti lokálnej databázy a implementovaných máp.

## 4.3.1 Testovanie lokálnej databázy

I D	1	Názov:		Testovanie lokálnej databázy			
Úr	oveň sp	lnenia te	stu:	Musí	Tester	Barbora Čelesová	
Ro	zhranie	:		Systém/používateľ			
Úč	el:			Overenie lokálnej databázy			
Vst	tupné p	odmienk	y:	Internet, PC			
Vý	stupné	podmien	ky:	Testovanie prebehlo úspešne			
	Kro	k	Akcia	Očakávaná reakcia	Skutočná reakcia		
1	Používateľ sa prihlási do Android aplikácie			Zobrazenie úľov používateľa	Zobrazenie úľov používateľa		
2	Používateľ si otvorí úľ			Zobrazenie meraní	Zobrazenie meraní		
3	Použív	vateľ si v	ypne internet	Bez interakcie	Bez interakcie		
4 Používateľ si vypne aplikáciu				Bez interakcie	Bez interakcie		
5	Použív Andro	vateľ sa p id apliká	rihlási do cie	Zobrazenie úľov používateľa	Zobrazenie úľov používateľa		
6	Použív	vateľ si o	tvorí úľ	Načítanie meraní z lokálnej databázy	Načítanie meraní z lokálnej databázy		

## 4.3.2 Testovanie funkčnosti máp

ID	2	Názov:		Testovanie funkčnosti máp			
Úro	oveň spl	nenia te	stu:	Musí	Tester	Barbora Čelesová	
Rozhranie:				Systém/používateľ			
Úče	: :			Overenie funkčnosti máp			
Vst	upné po	odmienk	y:	Internet, PC			
Výstupné podmienky:				Testovanie prebehlo úspešne			
	Kro	k	Akcia	Očakávaná reakcia	Skutočná reakcia		
1	Používateľ sa prihlási do Android aplikácie			Zobrazenie úľov používateľa	Zobrazenie úľov používateľa		
2 Používateľ si zobrazí pravý panel				Zobrazenie pravého panela	Zobrazenie pravého panela		
3	3 Používateľ si zobrazí mapu			Zobrazenie mapy a úľov na mape	Zobrazenie mapy a úľov na mape		
4	4 Používateľ klikne na úĽ			Zobrazenie detailov o úli	Zobrazenie detailov o úli		