

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

Internet vecí v našich životoch [IoT]

Inžinierske dielo

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

Obsah

1	Úvod.....	1
2	Globálne ciele projektu na zimný semester.....	2
3	Celkový pohľad na systém.....	2
3.1	Architektúra riešenia.....	2
3.2	Dátový model.....	3
4	Moduly systému.....	4
4.1	Analýza.....	4
4.1.1	Technológia Sigfox.....	4
4.1.2	Analýza frekvencie zberu a odosielania údajov.....	6
4.1.3	Analýza včelárstva.....	7
4.1.4	Carriots.....	9
4.1.5	Analýza existujúcich riešení.....	11
4.1.6	Arduino.....	13
4.1.7	Analýza senzorov.....	14
4.1.8	IoT zariadenie.....	16
4.1.9	Analýza merania stavu batérie.....	16
4.2	Návrh.....	17
4.2.1	Návrh zapojenia.....	17
4.2.2	Mockupy stránky.....	18
4.2.3	Mockupy aplikácie.....	21
4.2.4	Carriots service.....	22
4.2.5	Databáza PostgreSQL.....	23
4.2.6	Zabezpečenie spojenia.....	24
4.3	Implementácia.....	24
4.3.1	Arduino.....	25
4.3.2	Server Včelička.....	26
4.3.3	Mobilná aplikácia.....	27
4.4	Testovanie.....	28
5	Bibliografia.....	33

1 Úvod

V porovnaní s našim životom, je život včely len krátkym precitnutím, no jej dôležitosť je enormná. V prvom rade každému napadne opeľovanie rastlín, vďaka čomu máme ovocie či zeleninu. Predstava, že by túto prácu nevykonávali je katastrofická. Tým, že opeľujú rôzne rastliny na lúkach a pastvinách, zachovávajú prirodzený chod ekosystému planéty. V neposlednom rade sú nápomocné aj pri medicíne. Vyrábajú propolis, čo je prírodné liečivo vhodné pri liečbe popálenín či zápalov. Preto sme sa rozhodli využiť moderné technológie na pomoc včelárom pri pravidelných úkonoch potrebných na udržanie včelstva.

Profesionálny včelár, či ten, ktorý to považuje len za hobby potrebuje mať prehľad o stave svojich včelstiev, pokiaľ možno bez nutných osobných kontrol. Tí, ktorí nemajú pravidelný prístup k úľom ocenia možnosť vzdialenej kontroly a upozornenia na výraznú, ohrozujúcu situáciu, ktorá nastala. Najdôležitejším faktorom v živote včelstva je prítomnosť zdravej plodnej matky. Jednou zo základných súčastí nášho riešenia je poskytovanie informácií o teplote úľa a okolia. V situácii, kedy včelstvu chýba plod si vyžaduje zásah včelára, ktorý túto skutočnosť zistí práve monitorovaním teploty. Ďalším zo sledovaných faktorov je hmotnosť úľa. Pre laika informácia značiaca prítomnosť medu, pre včelára celá alchymia. Od nutnosti pridávania medníkov na ukladanie nového medu až po potrebu pridania zásob pre prežitie včiel počas zimy. V neposlednom rade je potrebné zabezpečiť informovanie o prevrátení úľa respektíve v dnešnej dobe aj o ukradnutí celého úľa.

Včelstvá väčšinou neumiestňujeme v blízkosti domova, ich miesto je v prírode, mimo každodennej civilizácie. Preto pravidelne aktualizované informácie a upozornenia o hraničných stavoch úľa zjednodušia prácu včelárom. Keďže predpokladáme, že v dnešnej dobe sa včelárstvu venujú aj starší ľudia, ktorí nemajú vytvorený vzťah s modernými technológiami, vytvárame jednoduché riešenia s intuitívnym používaním.

Obsahom tohto dokumentu je priebežná dokumentácia k inžinierskemu dielu nášho tímového projektu. Skladá z ďalších troch kapitol, ktoré odrážajú globálne ciele stanovené na zimný semester, celkový pohľad na systém a jednotlivé moduly systému. V tretej kapitole sme sa zamerali hlavne na našu navrhnutú architektúru systému, dátový model a diagram tried. Súčasťou štvrtej kapitoly je analýza potrebných súčastí projektu, ktorú sme vykonávali v prvom šprinte. Následne od jej obsahu sme sa odrazili pri návrhu riešenia, jeho implementácii a samotnom testovaní.

2 Globálne ciele projektu na zimný semester

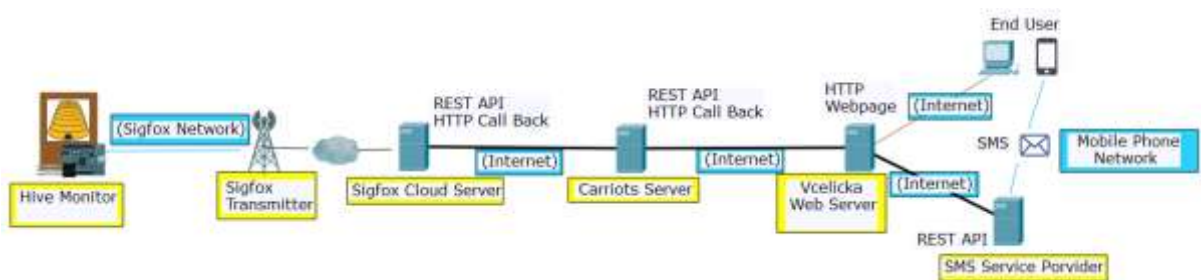
Základným cieľom nášho projektu pre zimný semester je zobrazenie veličín vo webovej aplikácii a android aplikácii. V závislosti od dodania hardvéru sa bude aktívne pracovať aj na tejto časti projektu. Na konci semestra bude mať tím funkčný prototyp úľa spolu so získaním údajov zo senzorov.

3 Celkový pohľad na systém

Táto kapitola sa venuje hlavne architektúre riešenia, opisu jej jednotlivých prvkov a dátovému modelu, podľa ktorého budú spracovávané dáta o používateľoch.

3.1 Architektúra riešenia

Táto kapitola stručne popisuje jednotlivé časti spoločne tvoriace architektúru nášho systému. Samotné prepojenie je zobrazené na nasledujúcom obrázku.



Obrázok 1 Architektúra riešenia

Hive Monitor

Pozostáva z mikrokontroléra so sigfox anténou (Arduino Mega 2560) a meracích senzorov. Medzi senzory patria:

- Senzor na meranie vonkajšej teploty.
- Senzor na meranie vnútornej teploty.
- Senzor na meranie vlhkosti v úli.
- Senzor na meranie naklonenia úľa (akcelerometer).
- Senzor na meranie hmotnosti úľa

Mikrokontrolér aj senzory sú pripevnené na včelom úli. Mikrokontrolér odosiela namerané dáta cez Sigfox sieť pomocou Sigfox antény.

Sigfox Transmitter

Komunikuje so zariadeniami pripojenými na Sigfox sieť. Dáta zo zariadení posiela na Sigfox Cloud Server.

Sigfox Cloud Server.

Zhromažďuje dáta zo zariadení na Sigfox sieti. Dáta sa dajú zo serveru stiahnuť cez internet pomocou REST API rozhrania v JSON formáte. Keď prídu na Sigfox Cloud Server nové namerané dáta, tak cez službu HTTP Callback zavolá funkciu na prijatie stiahnutie dát na Carriots Server.

Carriots Server.

Carriots je platforma pre IoT zariadenia, a bude slúžiť ako úložisko pre namerané dáta. Dáta sa dajú zo serveru stiahnuť cez internet pomocou REST API rozhrania v JSON formáte.

Včelička Web Server

Hostuje našu webovú stránku, pomocou ktorej si budú môcť koncoví používatelia pozerat' namerané dáta zo včelích úľov.

Koncový používatelia

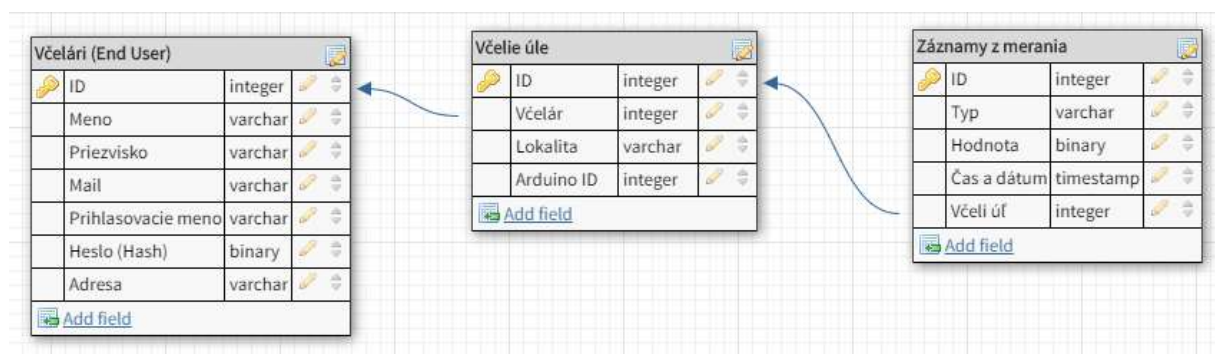
Koncoví používatelia sú registrovaný včelári a pozvaný pozorovatelia. Namerané dáta zo včelích úľov si môžu používatelia pozrieť pomocou našej webovej stránky včelička, alebo pomocou mobilnej aplikácie (Android).

SMS Service Provider

Slúži na notifikovanie používateľov o prekročení hraničných hodnôt (napr. prevrátenie úľa). SMS sa dajú posilať pomocou REST rozhrania.

3.2 Dátový model

Táto kapitola popisuje dátový model, podľa ktorého sú v našom systéme spravované dáta používateľov.



Obrázok 2 Dátový model

Včelári

Včelári sú koncoví používatelia nášho riešenia. K nameraným dátam z úľov pristupujú pomocou mobilnej aplikácie, alebo pomocou webovej stránky. V našom systéme budú o používateľoch

ukladané základné informácie: meno, priezvisko, mailovú adresu, prihlasovacie meno, heslo (hash + salt), adresa a identifikačné číslo používateľa.

Včelie úle

V našom systéme sú uložené informácie o každom monitorovanom včelom úle. Medzi uložené informácie patria: lokalita úľu, identifikačné číslo používateľa, ktorému včelí úľ patrí, identifikačné číslo priradeného meracieho zariadenia (Arduino) a identifikačné číslo úľu.

Záznamy z merania

Obsahuje namerané hodnoty zo senzorov. Medzi uložené informácie patria: typ merania (vlhkosť, hmotnosť...), nameraná hodnota, čas a dátum merania, včelí úľ, na ktorom bola hodnota nameraná a identifikačné číslo merania.

4 Moduly systému

Táto kapitola opisuje jednotlivé moduly nášho projektu. Pre niektoré časti bolo potrebné urobiť analýzu oblasti a následne urobiť návrh funkcionality týchto modulov. Navrhnuté moduly boli následne implementované v daných programových prostrediach a pomocou rôznych programovacích jazykov. Všetky implementované časti bolo nutné otestovať a preukázať ich správnu funkcionality.

4.1 Analýza

Súčasťou prvého šprintu bolo okrem iného analyzovanie jednotlivých oblastí, ktoré budú súčasťou nášho projektu. Obsahom tejto kapitoly je teda to najpodstatnejšie, čo sme sa z analýzy dozvedeli. Ide hlavne o priblíženie technológie Sigfox, zistenie frekvencie zberu odosielania údajov, analýza existujúcich riešení, ktoré sú nasadené do reálnej prevádzky, Carriots, Arduino či IoT zariadenia.

4.1.1 Technológia Sigfox

Jedná sa o sieť, pomocou ktorej môžu IoT zariadenia medzi sebou komunikovať. Sigfox využíva na komunikáciu úzke frekvenčné kanály (UNB) a bunky tejto siete umožňujú pokryť oveľa väčšie územie než základné stanice pri GSM. Táto technológia je vybudovaná na bunkovom systéme podobne ako GSM a využíva topológiu hviezda. Dosah každej základňovej stanici je cca 200 km pri priamej viditeľnosti, v teréne kde je viacero prekážok je to 50 km a v lesových oblastiach je to cca 2-10 km. Sieť pracuje vo frekvenčnom pásme 868 MHz v Európe a 902 MHz v USA. Najvyšší vysielač výkon je povolený 25 mW.



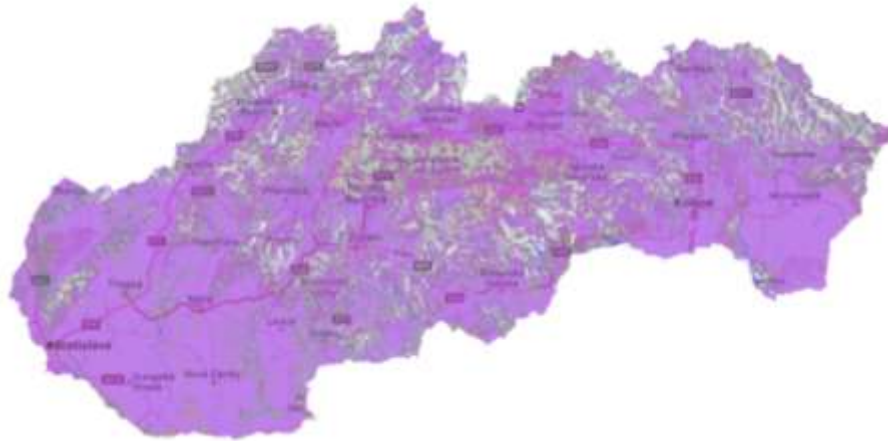
Obrázok 3 Architektúra siete Sigfox

Medzi základné prvky Sigfox architektúry patria:

- Sensory – slúžia na meranie určitej veličiny. Sensory môžu byť nastavené tak, že buď posielajú data do jedného hlavného senzora a až potom ďalej alebo posielajú údaje priamo každý zvlášť.
- Sigfox stanice – slúžia na zaistenie dostupnosti siete Sigfox. Existuje ich viacero a každá stanica pokrýva určitú oblasť.
- Sigfox úložisko – sa používa na ukladanie používateľských dát.
- Používateľské zariadenia – prijímajú dáta zo senzorov a obvykle z nich robia buď nejaké štatistiky. Všeobecne slúžia ako zobrazovacie zariadenia pre senzory.

V tejto technológii je implementovaný protokol, ktorý umožňuje prenášať malé objemy dát a má nízku spotrebu energie, čo je pre IoT ideálne. Každý paket je veľký 12 bajtov a každé koncové zariadenie využívajúce technológiu Sigfox by malo poslať denne maximálne 140 takýchto paketov, čo predstavuje jednu správu za približne 10 minút a celkovú dennú kapacitu 1680 bajtov. Paket nemá žiadnu predpísanú štruktúru. Na adresovanie zariadení sa využívajú interné identifikátory, ktoré emituje certifikačná autorita, pričom zariadenie sa zaregistruje u prevádzkovateľa pri aktivácii služby. Problémom pri tomto protokole je negarantovanie prijatia správy. Aby bola ale pravdepodobnosť doručenia čo najvyššia, tak sa správa posielala vždy trikrát a to zakaždým na inú náhodne vybranú frekvenciu. Vysielač potom prijatú správu pošle na server prevádzkovateľa a ten skontroluje jej validitu. Následne sa správa sprístupní klientovi, ktorému je určená buď cez REST API, prípadne sa správa zašle na server klienta cez http callback.

Na Slovensku poskytuje technológiu Sigfox len jediný mobilný operátor, ktorým je SimpleCell Networks Slovakia. Táto sieť je určená výhradne pre komunikáciu medzi IoT zariadeniami. Na nasledujúcom obrázku je možné vidieť aktuálne pokrytie siete Sigfox v SR.



Obrázok 4 Aktuálne pokrytie siete Sigfox v SR

4.1.2 Analýza frekvencie zberu a odosielania údajov

Čo máme k dispozícii:

- Uplink (zo zariadenia do Sigfox siete)
 - max 140 správ za deň (1 správa/10 min)
 - v správe môžeme poslať 10 bajtov dát
 - 3x poslaná každá správa poslať
 - žiadne overenie prijatia
- Downlink (zo Sigfox siete na zariadenie)
 - každý paket 8 bajtov
 - každá správa má k využitiu 8 bajtov dát

Posielanie údajov je značne obmedzené maximálnym počtom odoslaných správ za deň (140) a veľkosťou jednej správy (10 B). Frekvencia zberu je obmedzená len veľkosťou pamäte Arduina. Odosielať môžeme konkrétnu hodnotu merania alebo priemer viacerých meraní (opakujúcich sa hneď po sebe, alebo po určitom časovom intervale napr. 10 min) pre zvýšenie presnosti merania. Hodnotu z akcelerometra nie je potrebné posielat' pravidelne, ale stačí keď ju odošleme len keď dôjde k zmene (prevrhnutie úľa). Na hodnotu z akcelerometra stačí 1 bit (prevrhnutý/neprevrhnutý). Či je úľ prevrhnutý by vyhodnocovalo Arduino na základe meraní z akcelerometra. Niektoré namerané hodnoty môže byť potrebné posielat' častejšie ako iné. Včelár (koncový používateľ), by si mohol v systéme vybrať, ktoré hodnoty potrebuje merať častejšie (prispôsobenie podľa osobných preferencií včelára). Treba nechať „rezervu“ (napr. 20 správ) pre odosielanie správ z meraní, ktoré prekročili hraničné hodnoty, keďže tieto dáta sú pre včelára dôležitejšie ako tie čo sa posielajú pravidelne. Keďže doručenie správy v Sigfox nie je zaručené, môžeme posielat' informácie o meraniach, ktoré prekročili hraničné hodnoty viac krát po vybranom časovom intervale (napr. 2x po 10 min). Hraničné hodnoty sú pre včelára dôležitejšie ako tie čo sa posielajú pravidelne. K dispozícii máme uplink 4 správy/deň, ktoré sa dajú použiť podľa potreby. Napríklad na odoslanie príkazu na zmenu frekvencie

merania vybranej veličiny, alebo na žiadosť o odoslanie vybranej veličiny (ak chce včelár vedieť nejakú z hodnôt okamžite).

4.1.3 Analýza včelárstva

Včelárstvo je amatérsky alebo profesionálny chov včiel, za účelom výroby medu, opelenia plodín alebo na odchov včiel pre ďalších včelárov. Starosť o včelie úle si vyžaduje starostlivosť včelára – výber medu, prikrmovanie včiel, ošetrovanie včelstva, rozširovanie úľov. Tieto úlohy sú rozličné počas celého roka.

Január

V januári včely potrebujú pokoj. Včely sedia pokojne v úli v tesnom chumáči, len niektoré včely počas slnečných dní opustia úľ. V druhej polovici mesiaca začne matka klásť vajíčka a včely sa starajú o plodisko veľkosti dlane. Zbytočné vyrušovanie v tomto čase znamená pre včely záťaž. V tomto období nie sú potrebné žiadne práce s úľom.

Február

Včely v tomto období vytvárajú nové plody, nezáleží na vonkajšej teplote. Plodisko sa však skutočne rozrastie až keď včely začnú znášať peľ.

Marec

V teplejších oblastiach sa už začiatkom marca vyliahne prvá generácia včelstva. Včelie matky už kladú niekoľko stoviek vajíčok denne a svoj výkon postupne zvyšujú. V polovici marca sa už včely starajú o 10000 zaplodovaných buniek, z ktorých sa po 21 dňoch vyliahnú mladé včely.

Apríl

Včelstvo sa v tomto období rozrastie na 30 až 40 tisíc jedincov. Zimné včely sa vytrácajú z úľa. Hlavným poslaním včelára v tomto období je kontrola rojivej nálady a to prezeraním plastov raz za týždeň. Všetky plasty sa vyťahujú z plodiska a kontroluje sa rojová materská bunka.

Máj

V máji začína hromadná znáška. Približne polovica včiel je zamestnaná v úli (upratovanie, starostlivosť o plod, stavba diela). Máj je mesiac, kedy má včelár najviac práce: rojové opatrenia, pridávanie medníkov, tvorba mladých rodín.

Jún

Obdobie vytáčania medu. Včelár vyberie z úľov medové plásty, avšak nie všetky. Je potrebné nechať pre včely zásoby.

Júl a August

V júli treba kontrolovať populáciu klieštika včelieho. Je potrebné sledovať spád klieštika na podložke a zavčasu a správne zasiahnuť, aby nebol ohrozený plod, z ktorého sa vyľiahne zimná generácia včiel - tá je vlastne budúcnosť rodiny.

September

Letné včely zbierajú peľ, aby sa mohli zimné včely vytvoriť zásoby. Rodiny sa zmenšujú. Včelár kontroluje zásoby a prítomnosť matky v úli.

Október

Včely sa začnú združovať do chumáča. Počas teplých dní ešte občas vyletia von. Pri klesajúcich teplotách je jadro chumáča tak vyhrievané, že teplota na okraju chumáča neklesne pod 10 stupňov. V strede chumáča je uložená včelia matka.

November

Včelár už nemá priamo so včelami čo na práci, až na občasnú vonkajšiu kontrolu, či včely nie sú ničím vyrušované.

December

V decembri nás včely nepotrebujú. Sedia pokojne v úli a navzájom sa ohrievajú natlačené v zimnom chumáči. Vonkajšia teplota chumáča je okolo 10°C. Čím menej sú vyrušované, tým pokojnejšie a lepšie prečkajú zimu. Až keď sa koncom decembra/začiatkom januára predlžuje deň, začnú byť aktívnejšie.

Monitorovanie teploty

Pomocou monitorovania teploty dokážeme zistiť:

- Vhodný čas ošetrovania včelstva proti škodcom (Klieštik včelí)
- Prítomnosť včelej matky
- Oplodnenie včelej matky

Ako hraničný stav sme zvolili teplotu vo vnútri úľa 35°C.



Obrázok 5 Graf teploty v úli

Monitorovanie váhy:

Bežný úl váži 20-25kg na jar, až 70-80kg v jesenných obdobiach. Pomocou monitorovania hmotnosti dokážeme zistiť:

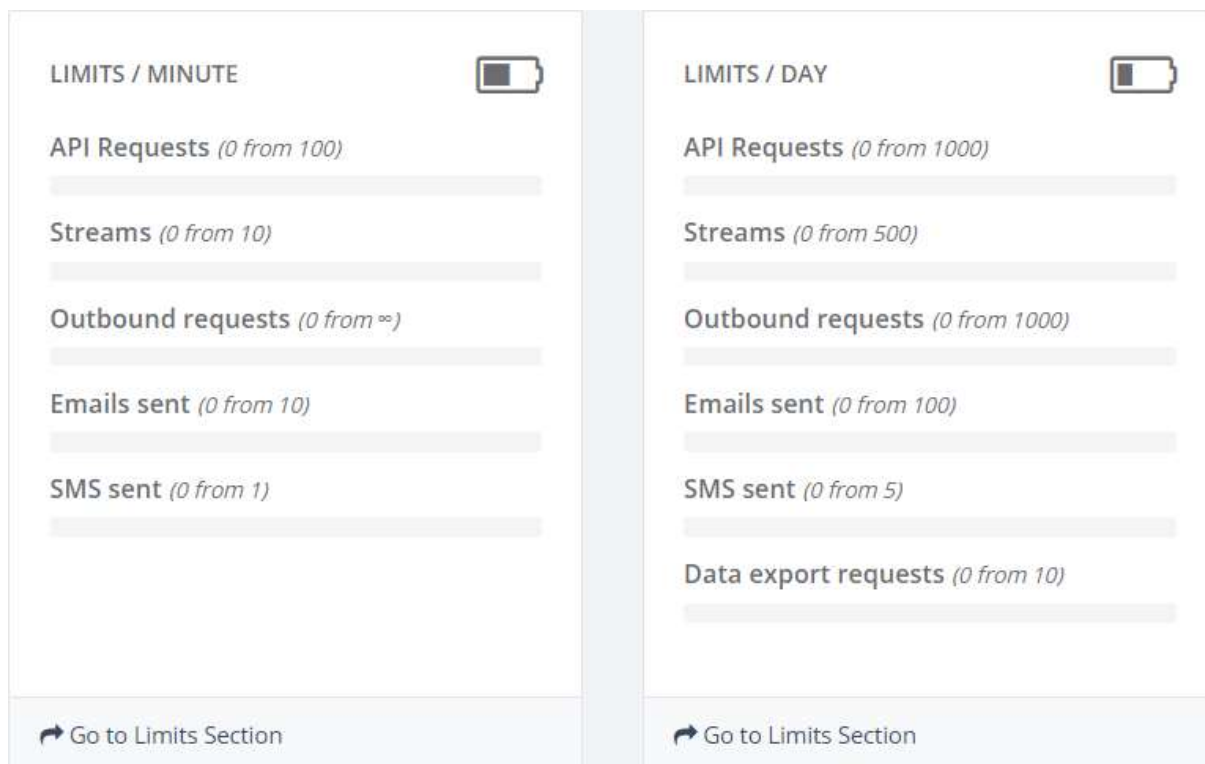
- V znáškovom období množstvo medu
- V mesiacoch máj a jún je nutná váha pre zistenie pridávanie medníkov, aby včely mali priestor ukladať med. Ak včely zistia, že majú málo miesta, dochádza k rojeniu a môže sa stať, že si včely vychovávajú novú včeliu matku a polka včelstva si odíde hľadať nové obydlie a zoberie so sebou aj časť medu
- V júli prichádza príprava včelstva na zimu. Po vytočení medu musíme včelám vrátiť zásoby na zimu, aby prežili do jari
- Pomocou váhy zistíme, že už znáškové obdobie skončilo a že môžeme pridať zásoby vo forme cukrového roztoku

Monitorovanie vlhkosti:

Keď včely naplnia úl medom, začnú sa zhlukovať a zvýši sa vlhkosť. Pri určitom stupni vlhkosti nastane vhodný okamžik pre vytočenie medu. Počas obdobia chovu mladých včiel sú stredné úrovne vlhkosti v úli medzi 50 a 60 percentami. Ak je vlhkosť príliš nízka, vajcia nebudú vyliahnuté a larvy sa vysušia. Ak je vlhkosť príliš vysoká, včely sú náchylnejšie na hubové choroby.

4.1.4 Carriots

Carriots je Platform as a Service (PaaS) navrhnutá pre Internet vecí (IOT). Carriots umožňuje zbieranie a ukladanie rôznych druhom dát zo zariadení používateľov. Umožňuje vytvárať aplikácie s vlastným SDK (Software development kit). Používateľ môže jednoducho integrovať vlastné aplikácie s externými IT systémami cez API, webové služby a prostredie hostingu. Carriots podporuje vývoj a rozšírenie od malých prototypov po tisíce zariadení. Limity, platné pre voľné členstvo zobrazuje



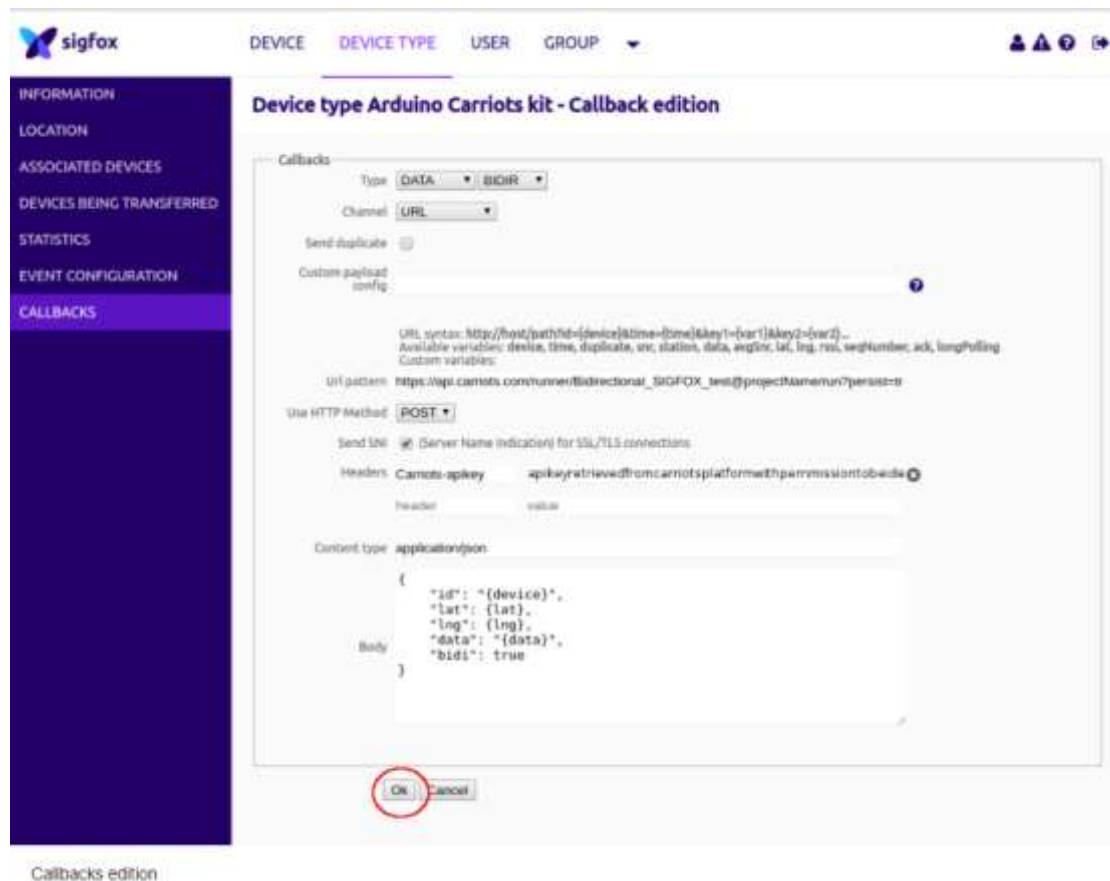
Obrázok 6 Limity pre voľné členstvo služby Carriots

Dôvody použitia

- redukuje čas vývoja projektu
- podporuje triggers: odosielanie emailov, sms správ

Hlavné výhody

- voľný prístup umožňuje manažovať až 2 zariadenia
- podpora REST API a Groovy SDK
- dokumentácia API a SDK
- podpora komunikácie so Sigfox cloudom



Obrázok 7 Nastavenie Callbacks pre Carriots v Sigfox cloude

Funkcie platformy

- **Manažment zariadení** - kontrola a interakcia so zariadeniami na diaľku. Kontrola stavu, zmena konfigurácie.
- **Listeners** - pri získaní dát alebo uložení je možné vytvárať udalosti s použitím IF-THEN-ELSE štruktúr.
- **Rules** - vytváranie Groovy skriptov, ktoré môžu byť použité v Listeneroch
- **Triggers** - umožňuje poslať dáta do externých systémov prostredníctvom emailu alebo sms
- **Export dát** - umožňuje spoluprácu s ďalšími IT systémami, vytvorenie exportovacích súborov, alebo použitie REST API pre manažovanie výstupných dát
- **Vytváranie logov** - zaznamenávanie interakcií s platformou

4.1.5 Analýza existujúcich riešení

V tejto kapitole sme sa zaoberali existujúcimi riešeniami, ktoré sú podobné ako náš projekt. Z každého riešenia sme sa sčasti inšpirovali a pridali podobnú funkcionality do nášho návrhu.

Arduino Beehive monitor

Autor Marc Curtis vytvoril monitorovací kit včelieho úľa na základe Arduina, senzorov a posielanie dát cez GSM sieť. Na zber dát použil platformu Xively. Jeho cieľmi boli:

- upozornenia cez SMS alebo e-mail ak hodnoty prekročia hraničné hodnoty
- merať hmotnosť úľa na zisťovanie množstva medu a množstva včiel v úli
- monitorovanie vstupu do úľa
- monitorovať frekvenciu bzučania v úli
- detegovať pohyb úľa, vietor, prevalenie úľa alebo vandalizmus

Základným kameňom jeho projektu bolo Arduino Duo a teplotný a vlhkosťový senzor DHT22. Keďže úle sú mimo dosahu akéhokoľvek Wi-Fi smerovača, bolo potrebné zabezpečiť posielanie dát. To vyriešil GPRS modulom pre Arduino. Dáta boli posielané cez GSM/GPRS sieť do Xively. Ďalším cieľom bolo zabezpečiť postačujúce napájanie. Všetky moduly fungujú perfektne pokiaľ sú pripojené k pevnému napájaniu. Na druhej strane, 9V batéria vydržala v jeho prípade 2 hodiny, kým stratila schopnosť poskytovať postačujúce množstvo elektrickej energie. Tento problém vyriešil solárnym panelom a lítiovou batériou, ktoré dokopy vytvárajú teoreticky sebestačnú energetickú jednotku. Na stránke projektu [1] má aj výsledky meraní a z nich vyplýva, že včely si v úli udržujú priemernú teplotu okolo 34°C.

Bee Smart TM

Beebot [2] je vzdialená diagnostická a monitorovacia stanica pre každý včelí úľ. Zhromažďuje informácie o zdraví úľa a produktivite pomocou nameranej teploty, vlhkosti a akustickej analýze zvuku každých 15 minút. Pri riešení využili bezdrôtovú váhu HiHive, ktorá sleduje zmenu hmotnosti a poskytuje maximálny prehľad úľa. Bola špeciálne navrhnutá pre úle a náročné podmienky. Na jedno nabitie vydrží až 3 mesiace. Bee Smart tiež sleduje pohyb a informuje včelárov o prípadnom premiestnení alebo prevrátení úľa. Riešenie poskytuje záznamy v reálnom čase, šetrí čas včelárov, zjednodušuje prevádzku, znižuje náklady a napomáha zvýšiť výnos.

Všetky informácie sa zostavia do správy a posielajú sa na cloud tri krát denne. Samotné dáta prechádzajú sériou algoritmov a upozorňujú používateľov v prípade potreby. Týmto je zabezpečené minimalizovanie počtu manuálneho sledovania situácie zo strany včelárov. Beebot využíva na komunikáciu Wi-Fi. Komunikačný modul je aktívny iba 8 minút z 24 hodín, čo nespôsobuje význačné rušenie včiel. Zariadenie je napájané z batérie, ktorá vydrží po dobu 6 mesiacov.

ProBee

Ide o súbor zariadení pre elektronické monitorovanie včelstiev, prezentáciu výsledkov a ich vyhodnocovanie, spoločne s evidenciou všetkých aktivít včelára. Systém sa skladá z niekoľkých

samostatných častí a to napríklad zvukový a teplotný senzor, úľová váha, vibračný senzor s GPS vysielacom na sledovanie polohy a úľová kamera. Všetky tieto časti zasielajú informácie na server, kde sú ukladané a vyhodnocované. Takisto obsahuje možnosť zasielania upozornení v prípadoch, že sa napríklad hranične zmení teplota alebo váha. Systém ProBee je napojený na výkonné počítačové centrum IBM, kde sú analyzované akustické merania. Výsledky sa následne vracajú naspäť do ProBee, kde sú názorným spôsobom zobrazené alebo zaslané používateľovi [3].

4.1.6 Arduino

Je to voľne šíriteľná elektronická platforma založená na ľahko použiteľnom hardvéri a softvéri. Arduino dosky dokážu čítať vstupy - svetlo pomocou snímača, stlačené tlačidlo alebo Twitter správu - a zmeniť ho na výstup - aktiváciu motora, zapnutie LED, publikovanie niečoho online. Môžete povedať svojej doske, čo má robiť tým, že pošlete sadu inštrukcií na mikrokontrolér. Použite programovací jazyk Arduino (založený na Wiring) a softvér Arduino (IDE) založený na Processing. V priebehu rokov bolo Arduino mozgom tisícov projektov, od každodenne používaných objektov po zložité vedecké nástroje. Na tejto platforme s otvoreným zdrojom sa zhromaždila celosvetová komunita tvorcov - študenti, umelci, programátori a odborníci, ktorých príspevky prispeli k neuveriteľnému množstvu prístupných poznatkov, ktoré môžu byť veľmi nápomocné pre začiatočníkov i odborníkov. Arduino sa narodilo v Ivrea Interaction Design Institute ako jednoduchý nástroj pre rýchle prototypovanie, zameraný na študentov bez znalostí z oblasti elektroniky a programovania. Akonáhle sa dostal do širšej komunity, Arduino doska sa začala meniť, aby sa prispôbila novým potrebám a výzvam, rozšírila svoju ponuku od jednoduchých 8-bitových dosiek k produktom pre aplikácie IoT, prenosné, vnorené prostredia a 3D tlač. Všetky dosky Arduino sú voľne šíriteľné, umožňujú používateľom budovať ich nezávisle na sebe a nakoniec ich prispôbovať konkrétnym potrebám. Softvér je tiež voľne šíriteľný a rastie prostredníctvom príspevkov používateľov na celom svete [4].

Arduino sa používa v tisícoch rôznych projektov a aplikácií vďaka jednoduchému dizajnu. Softvér Arduino je ľahko použiteľný pre začiatočníkov, ale je dostatočne flexibilný aj pre pokročilých používateľov. Beží na systémoch Mac, Windows aj Linux. Učitelia aj študenti ho používajú na vytváranie nízko nákladových vedeckých nástrojov, na preukázanie zásad chémie a fyziky alebo na programovania a robotiku. Návrhári a architekti vytvárajú interaktívne prototypy, hudobníci a umelci ju používajú na inštaláciu a experimentovanie s novými hudobnými nástrojmi. Výrobcovia ju, samozrejme, používajú na výstavbu mnohých projektov vystavených napríklad v spoločnosti Maker Faire. Arduino je kľúčovým nástrojom naučiť sa nové veci.

Existuje mnoho ďalších mikrokontrolérov a platforiem mikrokontrolérov, ktoré sú k dispozícii pre fyzickú výpočtovú techniku. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard a mnoho ďalších ponúkajú podobnú funkčnosť. Všetky tieto nástroje zabalia komplikované detaily programovania mikrokontrolérov do ľahko použiteľného balíka. Arduino tiež

zjednodušuje proces práce s mikrokontrolérami. Výhody, ktoré prináša pre učiteľov, študentov a zainteresovaných amatérov v porovnaní s inými systémami:

- **Lacný** – dosky Arduino sú relatívne lacné v porovnaní s ostatnými platňami s mikrokontrolérom. Najlacnejšia verzia modulu Arduino môže byť zostavená ručne a dokonca aj predmontované moduly Arduino stoja menej ako 50 dolárov.
- **Univerzálny** – Softvér Arduino (IDE) pracuje na operačných systémoch Windows, Macintosh OSX a Linux. Väčšina mikrokontrolérových systémov je obmedzená na Windows.
- **Jednoduché a prehľadné programovacie prostredie** – Softvér Arduino (IDE) je pre začiatočníkov ľahko použiteľný, ale je dostatočne flexibilný, aby mohli pokročilí používatelia využiť. Pre učiteľov je to pohodlne založené na spracovateľskom programovacom prostredí, takže študenti, ktorí sa učia programovať v tomto prostredí, budú oboznámení s tým, ako funguje IDE Arduino.
- **Voľne šíriteľný softvér** – Softvér Arduino sa uverejňuje ako nástroje s otvoreným zdrojovým kódom, ktoré môžu skúsení programátori rozšíriť. Jazyk sa dá rozšíriť prostredníctvom knižníc C++ a ľudia, ktorí chcú porozumieť technickým detailom, môžu urobiť skok z Arduina do programovacieho jazyka AVR-C, na ktorom je založený. Podobne môžete pridať kód AVR-C priamo do programov Arduino.
- **Voľne šíriteľný hardvér** – Plány Arduino sa uverejňujú pod licenciou Creative Commons, takže skúsení dizajnéri obvodov môžu vytvoriť vlastnú verziu modulu, rozšíriť ju a zdokonaľiť. Dokonca aj relatívne neskúsení užívatelia môžu vytvoriť verziu modulu, aby pochopili, ako to funguje a ušetrili peniaze.



Obrázok 8 Arduino MEGA 2560

4.1.7 Analýza senzorov

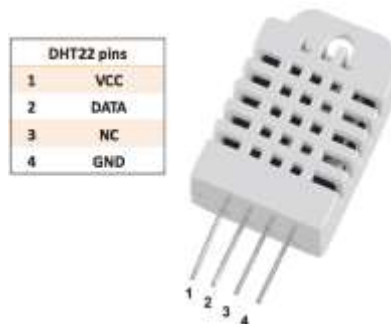
Táto kapitola opisuje základné vlastnosti senzorov teploty, vlhkosti a akcelerometra. Tieto údaje sme použili pri návrhu riešenia.

Senzor teploty a vlhkosti DHT22

Je to presný digitálny senzor teploty a vlhkosti prostredia. Ponúka širší rozsah teplôt a presnejšie merania ako jeho predchodca, DHT11. Na prácu so senzorom je potrebná knižnica. Na odfiltrovanie šumu z prostredia, výrobca odporúča pripojiť pull-up rezistor na DATA pin.

Základné špecifikácie:

- Operačné napätie: 3,3V – 6V
- Meranie vlhkosti: 0-100%RH \pm 2%RH
- Meranie teploty: -40~80°C \pm 0.5°C
- Nízka spotreba energie pri pomerne stabilných meraniach



Obrázok 9 Senzor teploty a vlhkosti DHT22

Akcelerometer

Tento senzor obsahuje 3-osí gyroskop spolu s 3-osím akcelerometrom. Senzor je založený na čipe MPU-6050.

Základné špecifikácie:

- Napájanie: 3 - 5V
- Komunikácia: komunikačný štandard IIC
- Vstavaný 16 bitový AD prevodník, 16 bitový výstup dát
- Rozsah gyroskopu: + 250 500 1000 2000 ° / s
- Rozsah zrýchlenia: \pm 2 \pm 4 \pm 8 \pm 16 g
- Veľkosť: 21 * 15 * 1.2 mm
- Hmotnosť: 3 g



Obrázok 10 Akcelerometer a gyroskop

4.1.8 IoT zariadenie

Táto kapitola sa zaoberá základnou špecifikáciou IoT zariadenia, ktoré rozšíri naše Arduino o prístup do Sigfox siete.

LPWAN SIGFOX anténa

Tento modul obsahuje lacný a malý UART modem spolu s 5dBi anténou. Modem ľahko spolupracuje s Arduino, Raspberry a aj s inými mikrokontrolérmi, ktoré majú Tx a Rx piny. Je určený pre LPWAN (Low Power Wide Area Network) siete. Tento typ siete je určený pre nízkoenergetické prenosy malých objemov dát a je vhodný pre zariadenia napájané z batérií.

Základné špecifikácie:

- Typ: rozširujúci kit
- Vývojová platforma: vlastná, Arduino, Raspberry Pi, Mbed
- Kompatibilita s AT príkazmi: áno
- Vývojové prostredie (IDE): vlastná, Arduino, Atmel Studio, mbed
- Senzory: teplota

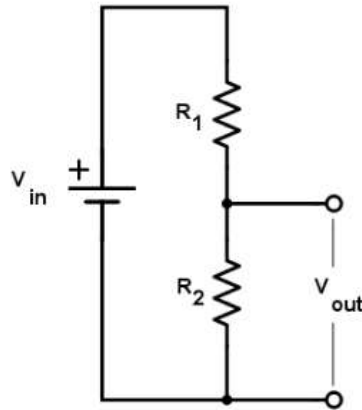


Obrázok 11 UART modem a anténa

4.1.9 Analýza merania stavu batérie

Stav batérie je možné vypočítať na základe hodnoty napätia batérie. Analógové vstupy môžu byť použité na meranie DC napätia medzi 0 a 5V. Pre meranie väčšieho rozsahu napätia je možné použiť dva odpory, pomocou ktorých vytvoríme napäťový delič. Delič napätia zníži merané napätie v rozsahu analógových vstupov Arduina. Následne použitím analógového vstupu je možné merať napätie väčšie ako 5V.

Obvod napäťového deliča pozostáva z dvoch odporov v sérii, ktoré rozdelí vstupné napätie na napätie merateľne analógovým vstupom Arduina. Schéma zapojenia takéhoto napäťového deliča je zobrazená na nasledujúcom obrázku, kde V_{OUT} je výstupne napätie (zmenšene napätie, ktoré je výstupom deliča), V_{IN} - vstupne napätie, R_1, R_2 - hodnoty odporov a $\frac{R_2}{R_1+R_2}$ pomer zmenšenia napätia.



Obrázok 12 Schéma zapojenia napät'ového deliča

$$V_{OUT} = V_{IN} * \frac{R_2}{R_1 + R_2}$$

Arduino obsahuje port respektíve pin, ktorý je možné použiť na napájanie Arduino z externého zdroja napájania, alebo ak je Arduino napájané pomocou napájacieho konektora je možné pomocou tohto pinu napätie zdroja získať. Takéto zapojenie je možné vidieť na obrázku nižšie. Arduino prečíta hodnoty z analógového vstupu a následným vhodným prekonvertovaním získa hodnotu napätia zdroja.

4.2 Návrh

Táto kapitola poskytuje pohľad na návrh jednotlivých modulov systému a ich vzájomné prepojenie.

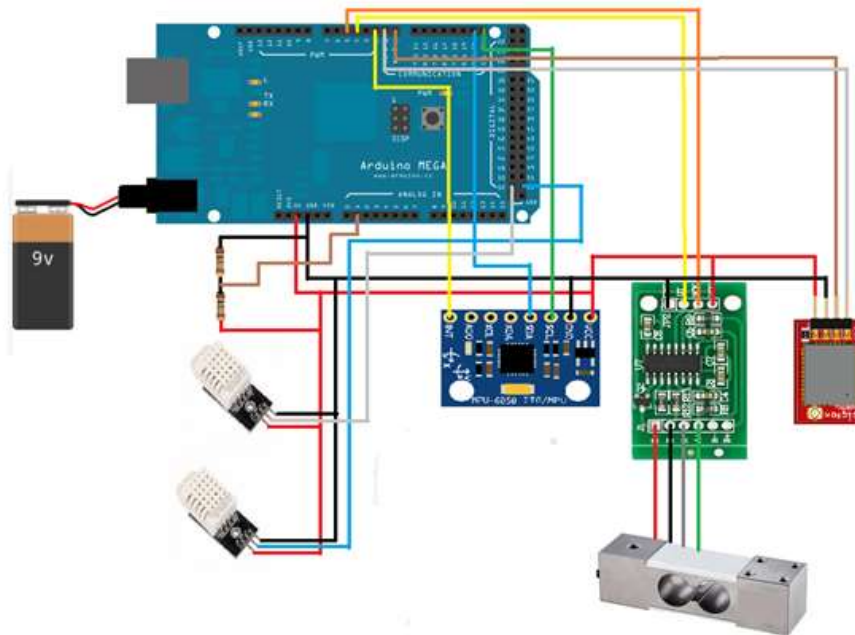
4.2.1 Návrh zapojenia

Pre meranie hmotnosti, teploty a odosielanie týchto údajov využijeme Arduino MEGA 2560. Arduino bude napájané 9V zdrojom napätia. Na Arduino budú pripojené dva senzory DHT22 na meranie vonkajšej aj vnútornej teploty a vlhkosti, akcelerometer MPU6050 pre meranie polohy úľa na základe ktorej bude detegované prevrátenie/odcudzenie úľa, analógovo – digitálny 24bit prevodník HX711 a hmotnostný senzor na meranie váhy úľa za účelom detekcie úľa plného medu a LpWan SigFox node modemu pre odosielanie nameraných údajov na SigFox Cloud. Zapojenie senzorov na Arduino je znázornené na nasledujúcom obrázku.

Priradené piny jednotlivým senzorom:

- DHT22 (1)
 - Digitálny pin 53
 - Napájacie piny +5V, GND
- DHT22 (2)
 - Digitálny pin 52
 - Napájacie piny +5V, GND

- HX711
 - PWM pin 4, 5
 - Napájacie piny +5V, GND
- MPU6050
 - Digitálny pin 2
 - Digitálne (Comuniacion pins SCL, SDA) piny 20, 21
 - Napájacie piny +5V, GND
- LpWan SigFox node
 - Digitálne (Comuniacion pins TX0, RX0) piny 20, 21



Obrázok 13 Zapojenie senzorov na Arduino

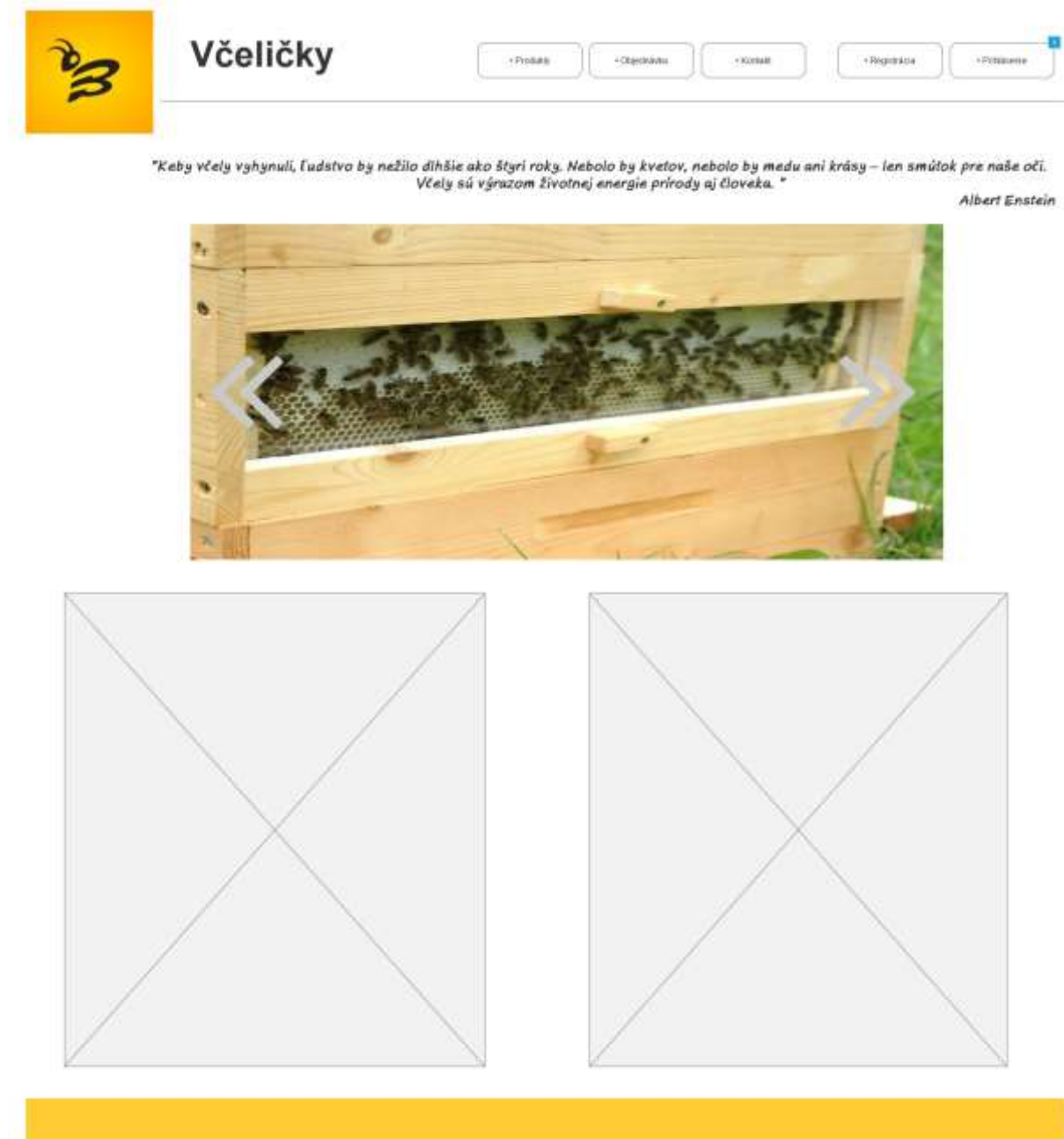
4.2.2 Mockupy stránky

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

Titulná stránka

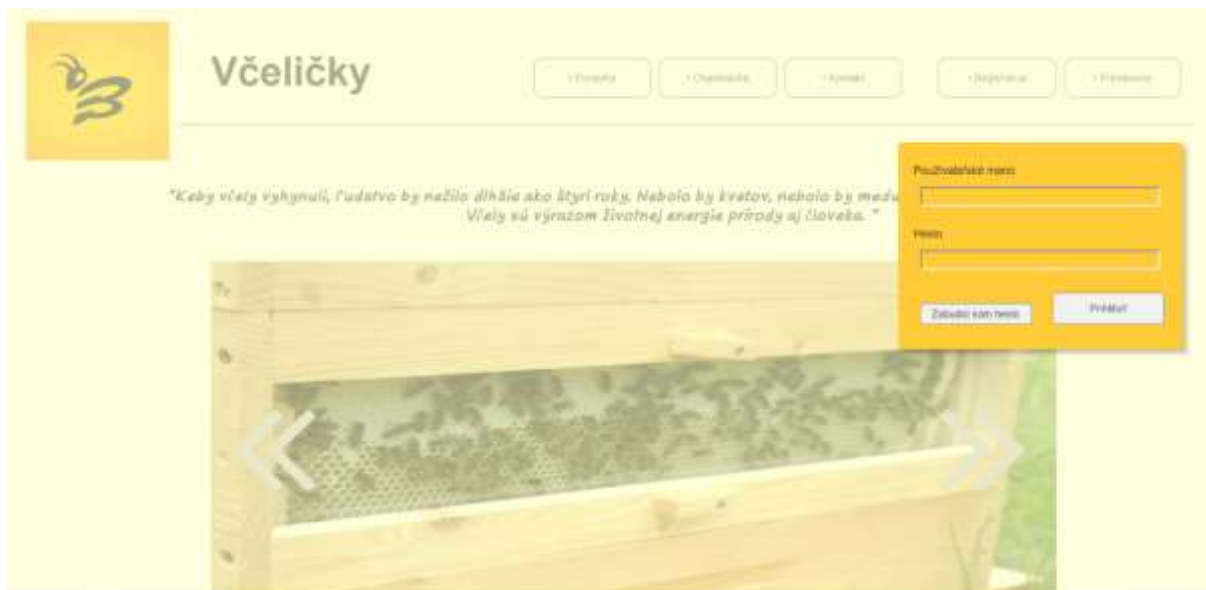
Základným prvkom stránky bude prezentácia obrázkov, ktorá má za úlohu zaujať nových ale aj stálych používateľov stránky. Jej obsahom budú zo začiatku tematické obrázky včiel, úľov alebo nášho riešenia. Po spustení stránky do reálnej prevádzky sa tam budú môcť nachádzať napríklad odkazy na nadchádzajúce spoločné udalosti alebo prípadné zmeny/novinky v projekte. Pod prezentáciou obrázkov bude umiestnený stručný zaujímavý popis, ktorý bude mať za úlohu nového používateľa nadchnúť a presvedčiť ho o investícii do nášho produktu.

Horný panel titulnej stránky bude obsahovať naše logo, nadpis stránky a päť ďalších položiek. V časti produkty sa bude nachádzať popis hardvérových zariadení použitých v našom riešení a ich spôsob využitia a implementovania. Ďalšou časťou bude objednávka pre zákazníka a kontakty na nás. Poslednými prvkami bude registrácia nového používateľa a prihlásenie.



Obrázok 14 Titulná stránka

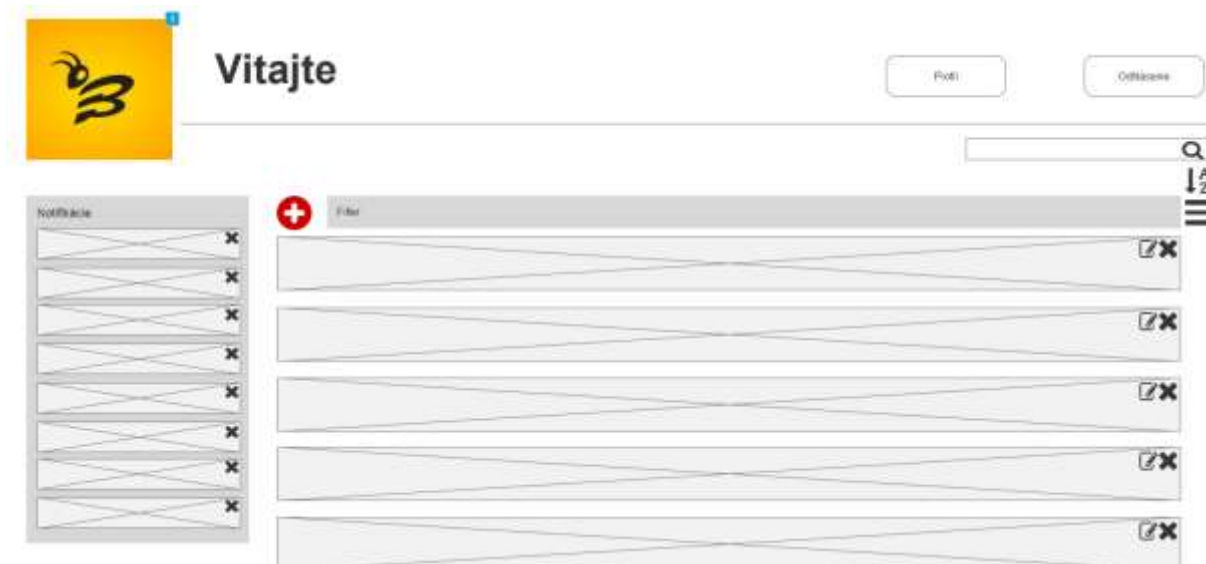
Nasledujúci obrázok zobrazuje jednoduchý formulár, ktorý bol zvolený na prihlásenie používateľov. Používatelia sa budú prihlasovať pomocou svojho emailu a zadaného hesla. V prípade, že zabudnú heslo, formulár im poskytne možnosť pomocou kontrolnej otázky overiť svoju totožnosť.



Obrázok 15 Titulná stránka – prihlásenie

Stránka po prihlásení včelára

Podľa zadaných prihlasovacích údajov systém rozozná, či ide o včelára alebo administrátora. Včelár si bude môcť v hornom paneli zobrazit' svoj profil. Hlavným obsahom bude prehľad úľov, ktoré má včelár pridelené. V každom momente si bude môcť jednotlivé záznamy pridávať, upravovať, mazať či filtrovať alebo vyhľadávať medzi nimi. Podstatnou súčasťou sú aj notifikácie, ktoré sú umiestené v pravej časti.



Obrázok 16 Stránka po prihlásení včelára

Stránka po prihlásení administrátora

Z grafického hľadiska bude kompozícia stránky po prihlásení administrátora veľmi podobne implementovaná ako stránka po prihlásení včelára, s tým rozdielom, že bude obsahovať záznamy o včelároch. Tie bude môcť spravovať a po rozkliknutí každého záznamu bude mať k dispozícii informácie o samotných úľoch včelára.



Obrázok 17 Stránka po prihlásení administrátora

4.2.3 Mockupy aplikácie

Pomocou webovej aplikácie proto.io boli vytvorené mockupy android aplikácie. V druhom šprinte vznikol návrh úvodnej obrazovky, prihlasovacej obrazovky a hlavná obrazovka s prehľadom úľov. V treťom šprinte bol pridaný bočný panel a jeho obsah.

Úvodné logo a prihlasovacia obrazovka aplikácie

Počas prvých sekúnd zapínania aplikácie je zobrazené logo ako hlavná dominanta obrazovky. Rovnako ako pri webovej aplikácii, používateľ sa prihlasuje so svojim emailom a zadaným heslom. Posledná, tretia obrazovka zobrazuje chybové okno, ktoré sa zobrazí po nesprávnom zadaní prihlasovacích údajov.



Obrázok 18 Úvodné logo a prihlasovacia obrazovka aplikácie

Prehľad úľov a bočný panel aplikácie

Hlavná obrazovka bude obsahovať prehľad úľov včelára. Nadpisom budú oddelené jednotlivé úle a v jednoduchom zobrazení budú obsahovať informácie o teplotách, váhe a vlhkosti. Farebné guľičky budú notifikovať aktuálny stav a uľahčia tak včelárovi prehľad.

Bočný panel bude obsahovať základné informácie o nás ako tvorcoch projektu a v hlavnej časti profil včelára, historické notifikácie, informácie o projekte a možnosť odhlásenia.



Obrázok 19 Prehľad úľov a bočný panel aplikácie

4.2.4 Carriots service

Na ukladanie dát sme sa rozhodli používať službu Carriots. Dáta v Carriots sú uložené vo formáte JSON v nerelačnej databáze.

Merania budú v Carriots databáze uložené v nasledovnom formáte:

```
"Merania": [  
  { "cas": "18.10.2017.22.37", "hodnota": 35, "typ": "IT" },  
  { "cas": "18.10.2017.22.37", "hodnota": 35, "typ": "OT" },  
  { "cas": "18.10.2017.22.37", "hodnota": 80, "typ": "H" },  
  { "cas": "18.10.2017.22.37", "hodnota": true, "typ": "P" }  
]
```

Obrázok 20 Formát merania v databáze

Získanie dát:

Pre účely získavanie dát používa Carriots REST API volania. Pre získanie meraní je potrebné uskutočniť metódu GET na URL <https://api.carriots.com/streams/>. Takýto request vráti merania zo všetkých zariadení.

Content-Type	application/json; charset=utf-8
Carriots.apiKey	582155c4a8a15467d4fbede176862673f4c5a5137b911e8a7cbf5034ff7c38ce

Obrázok 21 Hlavičky požiadaviek pre API volanie

V prípade, že chceme získať dáta z konkrétneho zariadenia, pridáme URL parameter „device“. V takom prípade bude vytvorený dopyt na adresu, v ktorej názov zariadenia označujeme ako „device_name“: https://api.carriots.com/streams/?device=„device_name“

Carriots API tiež poskytuje filtrovanie podľa času merania, umožňuje sort získaných výsledkov, nastavenie limitu alebo offset. Všetky dostupné parametre, ktoré poskytuje Carriots API sú dostupné v dokumentácii na adrese https://www.carriots.com/documentation/api/data_management.

4.2.5 Databáza PostgreSQL

Pre uchovávanie údajov o používateľoch a im patriacim zariadeniam sme sa rozhodli použiť databázu PostgreSQL, ktorá je nainštalovaná na webovom školskom webovom serveri, dostupnom na adrese <http://147.175.149.151/>.

Údaje v databáze nám umožnia uskutočniť REST API volania zo servera na Carriots pre získanie nameraných údajov pre konkrétneho používateľa. Každý používateľ má pridané zariadenia (tabuľka devices) tak ako sú vytvorené v službe Carriots. V službe Carriots sú uložené všetky namerané údaje. V databáze na serveri udržujeme len názov zariadenia zhodný s Carriots a GPS koordináty, ktoré v službe Carriots nie je možné pridať a držať ich na serveri je výhodné.

Tabuľka users:

- Id: Primárny kľúč
- Password_hash: Hodnota hashu hesla
- Password_salt: Hodnota soli, použitej pri vytvorení hodnoty hashu
- Role_id: Cudzí kľúč do tabuľky "roles"
- Email: Email používateľa - slúži na prihlasovanie
- Name: Meno a priezvisko používateľa

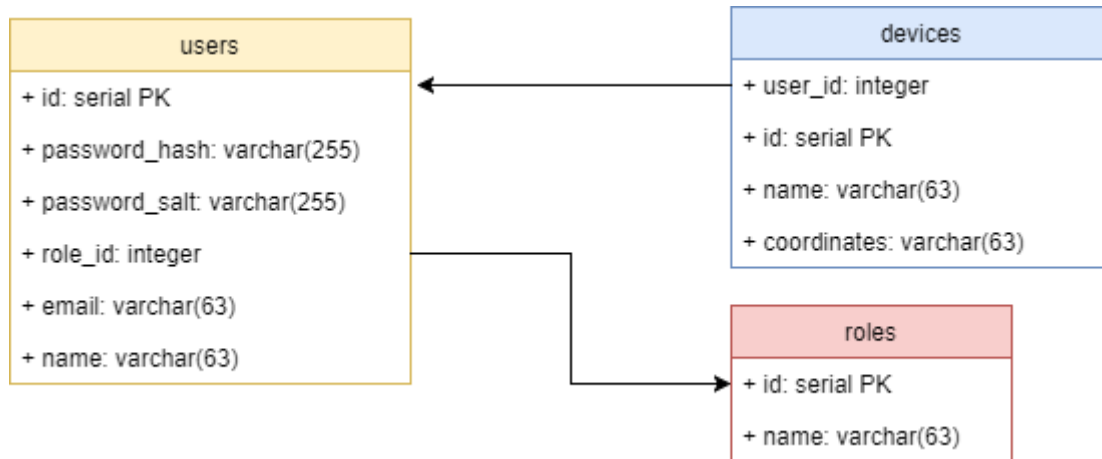
Tabuľka devices:

- Id: Primárny kľúč
- User_id: Cudzí kľúč do tabuľky "users"
- Name: Názov zariadenia, tak ako je uložený v službe Carriots

- Coordinates: Gps súradnice zariadenia

Tabuľka roles:

- Id: Primárny kľúč
 - Enumerácia, ktorá značí typ role
 - 1: Administrátor
 - 2: Včelár
 - 3: Pozorovateľ
- Name: Označuje názov danej role



Obrázok 22 Návrh databázy včelárov

4.2.6 Zabezpečenie spojenia

Po zadaní používateľského mena a hesla sa bude ešte pred odosielaním údajov heslo šifrovať nasledovne. Vezme sa čistý text hesla a zahashuje sa spoločne so saltom, ktorý bude uložený v android aplikácii. Ako hashovacia funkcia bola vybraná sha3-512, ktorá je momentálne považovaná za jednu z najbezpečnejších. Takto zahashovaný text bude následne odoslaný do siete. API na serveri potom prijme dané údaje a porovná z databázy najprv používateľské meno a následne jednotlivé hash. Týmto krokom sa zabráni odchyteniu hesla resp. sniffingu. Útočník získa len hash, spolu so saltom a nie čistý text hesla. Nemôže ani hádať, ktorý hash bol použitý keďže tam je taktiež pridaný salt. Pre prácu so sha3-512 by bolo vhodné mať najnovšiu verziu t.j. php 7.0. Samozrejme je možné použiť kľudne aj iný programovací jazyk, napr. Javascript. Tento algoritmus bude ale úplne účinný až vtedy, keď sa ako komunikačný protokol bude používať šifrovací protokol TLS medzi Java aplikáciou a API na serveri.

4.3 Implementácia

V tejto kapitole je popísaná implementácia jednotlivých modulov projektu. Každý modul bol implementovaný v špecifickom prostredí a s použitím špeciálnych nástrojov. V rámci projektu bola nutná implementácia komunikačných rozhraní na tímovom serveri ako aj mobilná a webová aplikácia.

4.3.1 Arduino

Táto kapitola popisuje implementáciu modulu Arduino. Zameriava sa hlavne na správnu funkcionálnu jednotlivých senzorov a spracovanie nameraných dát.

Modul merania teploty a vlhkosti

Pre meranie teploty a vlhkosti využívame digitálny senzor DHT22, ktorý dokáže merať obe veličiny. Na meranie hodnôt zo senzorov používame knižnicu DHT Sensor Library verzie 1.3.0. Pre získavanie hodnôt inicializujeme každý senzor, kde zadáme digitálny port na, ktorý je pripojený senzor a typ senzoru, ktorý používame. `<DHT dht(52, DHT22)>` a `<DHT dht2(53, DHT22)>`. Knižnica poskytuje funkcie `readHumidity()` a `readTemperature()`, pomocou ktorých získame potrebné hodnoty, ktoré môžeme ďalej spracovať. Na obrázku je znázornený výstup zo senzorov.

```
-----DHT22-----  
Humidity(IN): 50.40 %, Temperature(IN): 23.10 C  
Humidity(OUT): 46.70 %, Temperature(OUT): 25.40 C  
-----DHT22-----
```

Obrázok 23 Výstup zo senzorov

Modul detegovania prevrátenia/odcudzenia

Pre detegovanie prevrátenia/odcudzenia používame akcelerometer MPU-6050, ktorý meria osy x, y a z. Pre meranie využívame knižnicu MPU-6050 a I2Cdev. Pre komunikovanie MPU-6050 používame I2C protokol. Pre detegovanie používame vstavaný 1024 bytový FIFO buffer. Hodnoty senzora MPU-6050 sa ukladajú do buffra, ktoré sú následne čítané Arduino. Buffer je používaný spolu so signálom prerušenia. Ak MPU-6050 vloží dáta do buffra, signál prerušenia, signalizuje Arduino, že bola vložená hodnota do buffra a čaká na prečítanie Arduino. Hodnoty akcelerometra a gyroskopu sú nazývané „raw“ hodnoty. Predtým než získame údaje z akcelerometra, zariadenie je potrebné inicializovať pomocou funkcie `<mpu.initialize(>` a nastaviť potrebné offsety pre zvýšenie presnosti `mpu.setXGyroOffset(220)`. Zariadenie je pripravené pre čítanie hodnôt z akcelerometra. Arduino v slučke kontroluje či vzniklo prerušenie, ak áno prečíta vstup z FIFO buffra. Údaje získané z buffra zobrazí v stupňoch pomocou funkcie `mpu.dmpGetYawPitchRoll(ypr, &q, &gravity)`. Na základe získaných uhlov detegujeme prevrátenie/odcudzenie úľa. Na obrázku je znázornený výstup z akcelerometra MPU-6050.

```
-----Accelerometer-----  
X: -0.30°, Y: -0.07°  
State: hive is on right place  
-----Accelerometer-----  
-----Accelerometer-----  
X: -0.24°, Y: 23.11°  
State: hive was moved  
-----Accelerometer-----
```

Obrázok 24 Výstup z akcelerometra

Modul merania stavu batérie

Pre meranie stavu batérie používame napäťový delič, ktorý bol spomenutý vyššie. Hodnoty jednotlivých rezistoroch sú $R_1 = 10k\Omega$, $R_2 = 150k\Omega$, čím získam pomer $\frac{1}{16}$. Ako zdroj napätia používame batériový box s napätím 9V. Napäťovým deličom získame možnosť merať napätia zdroja, ktoré je väčšie ako 5V. Pre získanie napätia zdroja napájania získame 16 hodnôt meraných z analógového vstupu, ktoré sčítame čím získame hodnotu reprezentujúcu napätia zdroja. Túto hodnotu následne konvertujeme a kalibrujeme pre získanie hodnoty napätia zdroja vo voltoch. Z hodnoty napätia následne získame percentuálne ohodnotenie stavu batérie na základe vzorca:

percentage = (voltage - 6) * 33.3; Na obrázku je znázornený výstup.

```
-----Battery-----  
Voltage: 8.99 V  
Percentage: 99 %  
-----Battery-----
```

Obrázok 25 Výstup z batérie

4.3.2 Server Včelička

Tímový server poskytuje okrem webového sídla tímu aj ďalšiu funkcionality pre náš projekt. Jedným z modulov je overenie používateľa v databáze. Na serveri je implementovaná funkcionality na prihlásenie a registráciu používateľa v jazyku PHP. Jednotlivé triedy spolu vytvárajú API na prihlásenie a registráciu používateľa. Zabezpečujú overenie používateľa v databáze, ktorý sa chce prihlásiť do mobilnej alebo webovej aplikácie a aj pridanie nového používateľa pri registrácii do nášho systému.

4.3.2.1 Webová stránka

Hlavná časť webovej stránky je implementovaná vo frameworku Slim3. Na verziovanie PHP balíkov sa používa composer. Na zabezpečenie responzibility sa používa Bootstrap4. Na verziovanie javascript knižníc sa používa npm. Na správu javascript a css súborov sme použili nástroj gulp, pomocou ktorého sa spustia skripty, ktoré majú za úlohu uľahčiť správu a kompresiu týchto súborov. Používateľ sa na webovú stránku prihlási pomocou používateľského e-mailovej adresy a hesla. Server overí používateľa v databáze, a ak používateľ zadal správne prihlasovacie údaje prihlási ho na stránku. Používateľ si následne môže v aplikácii pozrieť hodnoty namerané senzormi na jeho úľoch. Dáta zo senzorov sa získavajú zo serveru včelička cez jeho REST API.

API pre získanie dát

Na serveri sú implementované funkcie pre získanie zariadení meraní z Carriots. Tieto funkcie plánujeme implementovať na Session, ktorá sa vytvorí po prihlásení používateľa.

Poskytnutie zariadení pre používateľa:

Cesta k získaniu zariadení: /db/devices/{user_id}

- HTTP metóda: GET
- Parameter {user_id} = id používateľa
- Vracia JSON zariadení [{"name":"nazov_zariadenia"}], tak ako sú uložené v Carriots

Poskytnutie posledného merania:

Cesta k získaniu zariadení: /api/measurements/{device_name}

- HTTP metóda: GET
- Parameter {device_name} = názov zariadenia
- Vracia JSON posledného merania[{"cas":"20.11.2017, "hodnota":30, typ:"IT"}]

4.3.3 Mobilná aplikácia

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 zahashovanie hesla a následné odoslanie na REST API serveru včelička. Server overí používateľa v databáze, a ak používateľ zadal správne prihlasovacie údaje prihlási ho do aplikácie. Používateľ si následne môže v aplikácii pozrieť hodnoty namerané senzormi na jeho úľoch. Dáta zo senzorov sa získavajú zo serveru včelička cez jeho REST API.

4.4 Testovanie

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

Tabuľka 1 Testovanie merania fyzikálnych veličín pomocou Arduina

ID:	1	Názov:	Meranie fyzikálnych veličín pomocou Arduina	
Úroveň splnenia testu:	Musí	Tester:	Jozef Vaľko	
Rozhranie:	Systém/používateľ			
Účel:	Overenie funkčnosti prototypu pre meranie veličín			
Vstupné podmienky:	Dostupný HW			
Výstupné podmienky:	Všetky merania prebehli v poriadku			
Kr ok	Akcia	Očakávaná reakcia	Skutočná reakcia	
1	Používateľ zapojí senzory podľa schémy zapojenia	Správne zapojenie senzorov na Arduino	Správne zapojenie senzorov na Arduino	
2	Používateľ spustí „Monitor Sériového portu“	Spustenie „Monitor Sériového portu“	Spustenie „Monitor Sériového portu“	
3	Používateľ zadá ľubovoľnú klávesu	Zadanie ľubovoľnej klávesy	Zadanie ľubovoľnej klávesy	
4	Používateľ overí meranie teploty a vlhkosti	Prečítanie a skontrolovanie hodnoty	Prečítanie a skontrolovanie hodnoty	
5	Používateľ overí meranie stavu batérie	Prečítanie a skontrolovanie hodnoty	Prečítanie a skontrolovanie hodnoty	
6	Používateľ overí funkčnosť akcelerometra	Zmena polohy prototypu a zistenie stavu úľa	Zmena polohy prototypu a zistenie stavu úľa	

Tabuľka 2 Testovanie "MainActivity"

ID:	2	Názov:	Testovanie „MainActivity“ (Android aktivita)	
Úroveň splnenia testu:	Musí	Tester:	Jozef Vaľko	
Rozhranie:	Systém/používateľ			
Účel:	Overenie správnej funkčnosti „MainActivity“ (Android aktivita)			
Vstupné podmienky:	Funkčné pripojenie na internet. Používateľ je prihlásený do aplikácie			
Výstupné podmienky:				
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia	
1	Spustenie aktivity po prihlásení používateľa.	Zobrazí sa prehľad úľov, a najnovších hodnôt, ktoré boli na nich namerané.	Zobrazí sa prehľad úľov, a najnovších hodnôt, ktoré boli na nich namerané.	
2	Kliknutie na vybraný úľ	Otvorí sa aktivita zobrazujúca detailne informácie o vybranom úli	Otvorí sa aktivita zobrazujúca detailne informácie o vybranom úli	
3	Potiahnutie prstom z ľavého kraja doprava	Zobrazenie bočného panelu (Menu). Bočný panel obsahuje položky: „Profil“, „Notifikácie“, „Odhlásenie“, Názov tímu, Mail tímu a odkaz na webovú stránku.	Zobrazenie bočného panelu (Menu). Bočný panel obsahuje položky: „Profil“, „Notifikácie“, „Odhlásenie“, Názov tímu, Mail tímu a odkaz na webovú stránku.	
4	Kliknutie na ikonu menu (ľavý horný roh)	Zobrazenie bočného panelu (Menu). Bočný panel obsahuje položky: „Profil“, „Notifikácie“, „Odhlásenie“, Názov tímu, Mail tímu a odkaz na webovú stránku.	Zobrazenie bočného panelu (Menu). Bočný panel obsahuje položky: „Profil“, „Notifikácie“, „Odhlásenie“, Názov tímu, Mail tímu a odkaz na webovú stránku.	
5	Kliknutie na ikonu možnosti (pravý horný roh)	Otvorenie panelu možností. Panel možností obsahuje možnosť „Edituj“	Otvorenie panelu možností. Panel možností obsahuje možnosť „Edituj“	

Tabuľka 3 Testovanie "LoginActivity"

ID:	3	Názov:	Testovanie „LoginActivity“ (Android aktivita)	
Úroveň splnenia testu:	Musí	Tester:	Jakub Pullmann	
Rozhranie:	Systém/používateľ			
Účel:	Overenie správnej funkčnosti „LoginActivity“ (Android aktivita)			
Vstupné podmienky:	Funkčné pripojenie na internet.			
Výstupné podmienky:				
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia	
1	Spustenie aplikácie	Zobrazenie aktivity „LoginActivity“	Zobrazenie aktivity „LoginActivity“	
2	Zadanie správnych prihlasovacích údajov (email, heslo). Stlačenie tlačidla prihlásiť	Zobrazenie aktivity „MainActivity“	Zobrazenie aktivity „MainActivity“	
3	Zadanie nesprávnych prihlasovacích údajov (email, heslo). Stlačenie tlačidla prihlásiť	Zobrazenie okna s chybovou hláškou: „Chyba! Nesprávne prihlasovacie údaje“	Zobrazenie okna s chybovou hláškou: „Chyba! Nesprávne prihlasovacie údaje“	
4	Stlačenie tlačidla prihlásiť bez zadania prihlasovacích údajov	Zobrazenie okna s chybovou hláškou: „Chyba! Nesprávne prihlasovacie údaje“	Zobrazenie okna s chybovou hláškou: „Chyba! Nesprávne prihlasovacie údaje“	

Tabuľka 4 Testovanie prihlásenia (Web)

ID:	4	Názov:	Testovanie funkcie prihlásenia na Web stránke včelička	
Úroveň splnenia testu:	Musí	Tester:	Matúš Sosňak	
Rozhranie:	Systém/používateľ			
Účel:	Overenie korektnosti funkcie prihlásenia používateľa			
Vstupné podmienky:	Internetové pripojenie, Url webu, Prihlasovacie údaje,			
Výstupné podmienky:				
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia	
1	Spustenie web stránky včelička	Zobrazenie hlavnej web stránky	Zobrazenie hlavnej web stránky	
2	Klíknutie na tlačidlo prihlásenie	Zobrazenie formulára pre zadanie prihlasovacích údajov	Zobrazenie formulára pre zadanie prihlasovacích údajov	
3	Zadanie správnych prihlasovacích údajov (login, heslo). Stlačenie tlačidla prihlásiť sa	Úspešné zobrazenie web stránky prihláseného používateľa	Úspešné zobrazenie web stránky prihláseného používateľa	
4	Zadanie nesprávnych prihlasovacích údajov (login, heslo). Stlačenie tlačidla prihlásiť	Zobrazenie okna s chybovou hláškou: „Chyba! Nesprávne prihlasovacie údaje“	Žiadna reakcia	

Tabuľka 5 Testovanie API na získanie zariadení používateľa

ID:	5	Názov:	Testovanie API na získanie zariadení používateľa	
Úroveň splnenia testu:	Musí	Tester:	Tomáš Koreň	
Rozhranie:	Systém/používateľ			
Účel:	Overenie správnej funkčnosti získania zariadení používateľa			
Vstupné podmienky:	Internetové pripojenie, Url, Rest Api Client			
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia	
1	Zapnutie REST API klienta (napr. Advanced REST Client)	Zapnutie klienta	Zapnutie klienta	
2	GET na URL: http://147.175.149.151/bee/public/db/devices/3	Vrátenie zariadení používateľa s ID=3	Vrátenie zariadení používateľa s ID=3	
3	GET na URL: http://147.175.149.151/bee/public/db/devices/1	Vrátenie null	Vrátenie zariadení používateľa s ID=3	

Tabuľka 6 Testovanie API na získanie posledného merania

ID:	6	Názov:	Testovanie API na získanie posledného merania	
Úroveň splnenia testu:		Musí	Tester:	Tomáš Koreň
Rozhranie:		Systém/používateľ		
Účel:		Overenie správnej funkčnosti získania posledného merania		
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia	
1	GET na URL: http://147.175.149.151/bee/public/api/measurements/DeviceTomas@fiittp20.fiittp20	Zobrazenie posledného merania pre zariadenie DeviceTomas@fiittp20.fiittp20	Zobrazenie posledného merania pre zariadenie DeviceTomas@fiittp20.fiittp20	
2	GET na URL: http://147.175.149.151/bee/public/api/measurements/DeviceBratislava@fiittp20.fiittp20	Zobrazenie posledného merania pre zariadenie DeviceBratislava@fiittp20.fiittp20	Zobrazenie posledného merania pre zariadenie DeviceBratislava@fiittp20.fiittp20	

Tabuľka 7 Testovanie registrácie používateľa

ID:	7	Názov:	Testovanie registrácia používateľa	
Úroveň splnenia testu:		Musí	Tester:	Michal Puškáš
Rozhranie:		Systém/používateľ		
Účel:		Overenie správnosti registrácie používateľa		
Krok	Akcia	Očakávaná reakcia	Skutočná reakcia	
1	Vynechanie alebo nesprávne vyplnenie polí v registračnom formulári	Upozornenie používateľa chybovou hláškou a zabránenie odoslaniu údajov	Upozornenie používateľa chybovou hláškou a zabránenie odoslaniu údajov	
2	Zaregistrovanie používateľa kliknutím na tlačidlo "Vytvoriť účet"	Pridanie riadku v databáze s novým používateľom	Pridanie riadku v databáze s novým používateľom	

5 Bibliografia

- [1] M. Curtis, „Arduino Beehive monitor,“ 15 08 2014. [Online]. Available: <https://hackaday.io/project/2453-arduino-beehive-monitor>. [Cit. 24 10 2017].
- [2] Bee Smart Technologies, „Bee smart Technologies,“ [Online]. Available: <https://beesmarttechnologies.com/beebot/>. [Cit. 24 10 2017].
- [3] ProBee, „ProBee,“ [Online]. Available: <http://www.probee.cz/Default>. [Cit. 24 10 2017].
- [4] Arduino, [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Cit. 19 10 2017].