**Stredná priemyselná škola elektrotechnická  
Komenského 44, 040 01 Košice**

**Multifunkčné monitorovacie zariadenie - meranie údajov včelích úľov**

Stredoškolská odborná činnosť

Č. odboru: 7

|  |  |
| --- | --- |
|  | Riešitelia |
| Košice | Martin Belej |
| 2024 | Matúš Bárány |
|  | Ročník štúdia: štvrtý |

**Stredná priemyselná škola elektrotechnická  
Komenského 44, 040 01 Košice**

**Multifunkčné monitorovacie zariadenie - meranie údajov včelích úľov**

Stredoškolská odborná činnosť

Č. odboru: 7

|  |  |
| --- | --- |
|  | Riešitelia |
| Košice | Martin Belej |
| 2024 | Matúš Bárány |
|  | Ročník štúdia: štvrtý |
|  | Školiteľ: Peter Keusch |

Vyhlasujem, že prácu stredoškolskej odbornej činnosti na tému (autor napíše názov svojej práce), napr. „Riadenie križovatky“ som vypracoval samostatne, s použitím uvedených literárnych zdrojov. Prácu som neprihlásil a ani neprezentoval v žiadnej inej súťaži, ktorá je pod gestorstvom MŠMVVaŠ SR. Som si vedomý dôsledkov, ak uvedené údaje nie sú pravdivé.

Obsah

[Obsah 3](#_Toc158803974)

[1 Úvod 1](#_Toc158803975)

[2 Problematika a prehľad literatúry 2](#_Toc158803976)

[2.1 Prehľad existujúcich riešení 2](#_Toc158803977)

[2.1.1 Beeconn.net 2](#_Toc158803978)

[2.1.2 Optibee.fr 2](#_Toc158803979)

[2.2 Použité technológie 3](#_Toc158803980)

[2.2.1 Vývojová doska ESP32 3](#_Toc158803981)

[2.2.2 Senzor hmotnosti 4](#_Toc158803982)

[2.2.3 HX - 711 4](#_Toc158803983)

[2.2.4 DHT 11 4](#_Toc158803984)

[2.2.5 Servo motor 4](#_Toc158803985)

[2.2.6 Java Spring 4](#_Toc158803986)

[2.2.7 Svelte 5](#_Toc158803987)

[2.2.8 Echarts 6](#_Toc158803988)

[2.2.9 @vincjo/datatables 6](#_Toc158803989)

[2.2.10 Tailwind CSS 6](#_Toc158803990)

[2.2.11 leaflet.js 6](#_Toc158803991)

[2.2.12 rollup.js 6](#_Toc158803992)

[2.2.13 Server - VPS 7](#_Toc158803993)

[2.2.14 MariaDB SQL + PhpMyAdmin 7](#_Toc158803994)

[2.2.15 Apache server + Tomcat 7](#_Toc158803995)

[2.2.16 Github automatické akcie 7](#_Toc158803996)

[2.3 Finančná analýza 8](#_Toc158803997)

[3 Výsledky práce 9](#_Toc158803998)

[3.1 Program pre mikrokontrolér 9](#_Toc158803999)

[3.2 Pripojenie zariadenia k internetu 9](#_Toc158804000)

[3.3 Napájanie 9](#_Toc158804001)

[3.4 Aktualizácia dát 10](#_Toc158804002)

[3.5 BeeConnect - senzory a smart konektory 10](#_Toc158804003)

[3.6 Tlačidlo a LED 10](#_Toc158804004)

[3.7 Plošné spoje 11](#_Toc158804005)

[3.8 Konštrukcia váhy 11](#_Toc158804006)

[3.9 Krabička zariadenia 11](#_Toc158804007)

[3.10 Akcie 11](#_Toc158804008)

[3.11 Rest API 12](#_Toc158804009)

[3.12 Štruktúra SQL (diagram) 12](#_Toc158804010)

[3.13 Párovanie nového zariadenia 12](#_Toc158804011)

[3.14 Konfigurácia frontendu 12](#_Toc158804012)

[3.15 Prihlasovanie / Registrácia 13](#_Toc158804013)

[3.16 Domovská stránka 13](#_Toc158804014)

[3.17 Upraviteľná domovská stránka dashboard 13](#_Toc158804015)

[3.18 Čiarový graf 13](#_Toc158804016)

[3.19 Mapa 13](#_Toc158804017)

[3.20 Aktuálna hodnota na senzore 14](#_Toc158804018)

[3.21 Prehľad úľov a ich nastavenia 14](#_Toc158804019)

[3.22 Upozornenia 14](#_Toc158804020)

[3.23 Kalendár 14](#_Toc158804021)

[3.24 Preklad do rôznych jazykov 14](#_Toc158804022)

[3.25 Server - VPS 14](#_Toc158804023)

[3.26 Štatistiky 15](#_Toc158804024)

[4 Závery a zhrnutie 16](#_Toc158804025)

[5 Resumé 18](#_Toc158804026)

[6 Zoznam použitej literatúry 19](#_Toc158804027)

# Úvod

Prvý zárodok myšlienky nášho projektu vznikol už dávnejšie, no reálnu podobu nadobudol, až po rozhovore s pánom učiteľom, ktorý nás bližšie uviedol do problému. Aj vďaka jeho perspektíve k danej problematike sme dokázali identifikovať strasti, ktoré má viacero včelárov a zároveň ťažkosti pri kúpe podobného zariadenia snažiace sa vyriešiť rovnaký problém ako naše. Hlavným prínosom týchto, ako aj nášho zariadenia je automatické meranie dát a následne zasielanie na server. Po zistení, že na trhu nie je dostatočne funkčný a robustný monitorovací systém a po krátkom (nedostatočnom) uvážení sme sa rozhodli pomôcť komunite včelárov a uľahčiť im prácu s včelami.

Cieľom nášho projektu je teda ponúknuť včelárom efektívny a spoľahlivý monitorovací systém, ktorý prinesie nádych moderných technológií do tradičného remesla včelárstva.

Pre náš projekt sme si zvolili mikropočítač ESP 32 a software bol postavený na frameworkoch java spring a svelte. Vývoju hardvéru sa venoval Martin, ktorý ESP programoval v jazyku Embeded C, zatiaľ čo frontendu sa venoval Matúš, ktorý si vybral javascript framework svelte. Na backende bolo z veľkej časti robené spoločne, kde sme použili technológie java spring.

Pri kúpe komponentov sme sa rozhodli hlavne pre slovenský e-shop techfun.sk ale kupovali sme aj zo zahraničných internetových obchodov ako Aliexpress.com alebo TME.eu

Chceli by sme poďakovať p. Ing. Petrovi Keuschovi, ktorý nám slúžil ako vynikajúci tester ako aj konzultant pre náš projekt. Taktiež by sme chceli poďakovať p. Ing. Zamborimu, za dôsledný výklad potrieb včelára, ako aj za prvú objednávku monitorovacieho zariadenia, ktoré v tej dobe ešte nebolo na svete.

# Problematika a prehľad literatúry

Ako sme už načrtli v úvode, našim cieľom bolo vytvoriť multifunkčné monitorovacie zariadenie. Ako hlavné ciele sme si stanovili jednoduchosť používania zariadenia, modulárnosť a použiteľnosť zariadenia v praxi.

## Prehľad existujúcich riešení

Na trhu sa už nachádzajú rôzne riešenia, ktoré sú avšak buď nespoľahlivé alebo príliš drahé. Väčšina z dostupných riešení funguje na gsm sim karte alebo wifi sieti. Našli sme aj také, ktoré používali.

### Beeconn.net

<https://beeconn.net/store.html>

Cena: 549.99€

Výhody:

Možnosť výmeny batérií

Možnosť napájať zariadenie cez solárny panel

Obsahuje displej na hlavnej komunikačnej jednotke

SIM - karta sa nachádza len na jednom, mieste - jednoduchšia výmena

Možnosť zapojenia viacero zariadení na jeden vysielač - resp. do série

Nevýhody:

Zariadenie nie je možné nabiť - nakoľko pracuje s 3xAA batériami

Plastová konštrukcia

Nekvalitné váhové moduly - predpokladaná životnosť zariadenia je 1 - 2 roky

Limitované zasielanie údajov - zariadenie odošle len jednu sms za deň

### Optibee.fr

<https://www.optibee.fr/en/>

Cena 399 - 599€

Výhody:

Kovová konštrukcia

Odnímateľná

Nevýhody:

Zastaralá webová stránka - pravdepodobne sa už toto zariadenie nepredáva

Naše zariadenie ponúka viacero vylepšení oproti aktuálnym ponukám dostupných na internete. Hlavnou výhodou nášho zariadenia sú tzv BeeConnect konektory. Vďaka týmto konektorom si môže včelár vybrať vlastnú kombináciu zariadení ako napríklad automatizované otváranie letáčov, či regulácia priechodu vzduchu a senzorov ako senzor teploty alebo vlhkosti, ktoré potrebuje na svojom úli. Naše riešenie obsahuje aj možnosť upravenia si ovládacieho panelu podľa vlastnej potreby, čo znamená že včelári si môžu vybrať a sledovať údaje na hlavnej stránke, ktoré reálne potrebujú a zaujímajú ich.

## Použité technológie

Technológie, ktoré sme použili sme si vyberali na základe toho ako dobre s nimi vieme pracovať, ale zároveň sme sa zopár z nich museli naučiť od základov. Dôležitým aspektom bol samozrejme aj čas, za ktorý sme museli toto zariadenie vytvoriť. Keďže natívne mobilné aplikácie sú náročnejšie na výrobu, najmä ak chceme niečo platformovo nezávislé, rozhodli sme sa urobiť kompromis. Kompromisom bolo vytvorené webovej stránky oproti natívnej android či ios aplikácií.

### Vývojová doska ESP32

ESP 32 je mikropočítač, ktorý slúži ako hlavná procesná jednotka nášho monitorovacieho zariadenia. Vývojová doska disponuje wifi a bluetooth modulmi, vďaka ktorým je možné komunikovať s internetom. Tento mikropočítač je programovateľný v embedded c. Obsahuje 26 programovateľných GPIO pinov, väčšinu ktorých sme využili pri našom projekte.

**Špecifikácie:**

Typ čipu: ESP32-WROOM-32U

SRAM: 520 KB

Flash: 4MB

Dvojjadrový procesor dual-core Xtensa® 32-bit LX6 MCU

Frekvencia: 40 MHz

### Senzor hmotnosti

Senzor určený pre mikrokontroléry, ktorý funguje na princípe ohýbania miniatúrnych kovových častí a zmenou elektrického odporu. Rozdiely hodnôt by však boli veľmi malé, a preto je pripojený ako wheatsonov mostík, čo umožňuje presné meranie nízkeho odporu.

### HX - 711

24-bitový analogovo-digitálny prevodník (ADC) navrhnutý pre senzory hmotnosti, pretože rozdiely sú pre mikrokontrolér stále príliš malé. Komunikácia prebieha cez jeho vlastné sériové rozhranie.

### DHT 11

Modul Určený pre meranie teploty a vlhkosti. Obsahuje mikrokontrolér, ktorý posiela hodnoty cez sériovú zbernicu.

### Servo motor

Motor, ktorý obsahuje potenciometer a čip, pomocou ktorého vieme motor pohnúť na určitú polohu. Nevýhodou je napríklad presnosť, ale pre naše potreby to postačuje.

### Java Spring

Java Spring Framework je jednou z najpopulárnejších a rozšírených technológii pre vývoj fullstack webových aplikácií v jazyku Java. Funguje na logike Model-View-Controller (MVC), ktorá umožňuje vytváranie robustných webových aplikácií. Model slúži na logiku webovej aplikácie, View je zodpovedný za zobrazovanie informácií užívateľovi zatiaľ čo Controller spracúva jeho požiadavky a riadi tok dát medzi Modelom a View. S touto paradigmou sme však neboli úplne spokojní, nakoľko spring používa súbory JSP, čo je tzv. jazyk HTML spojený s jazykom java. Tento spôsob vytvárania podstránok avšak nevyhovoval naším požiadavkám a preto sme sa rozhodli integrovať frontendový framework svelte. Do springu sme importovali framework Hibernate, pomocou ktorého dokáže server komunikovať s SQL databázou, ponúka množstvo anotácií, s ktorými vieme prevádzať riadky v databáze na objekty v Jave a naopak. Zabezpečuje to Java Persistence API (JPA), ktorý poskytuje množstvo nástrojov pre objektovo-relačné mapovanie (ORM).

**Výhody:**

**Nám známy Java ekosystém** - hlavnou výhodou Java springu je samotný jazyk, na ktorom je postavený. S týmto ekosystémom sme zohratí a preto nám táto voľba robila najmenší problém pri vývoji.

**Jednoduchosť pracovania s databázou** - framework hibernate, nám umožňuje jednoduchú prácu s databázou.

**Škálovateľnosť/Modulárnosť** - Spring je navrhnutý tak, aby bol modulárny, čo nám umožňuje oddeliť si časti nášho systému. Týmto spôsobom môžeme efektívne spravovať našu aplikáciu a udržiavať čistý kód.

**Jednoduchý deployment na server** - Java Spring Boot nám umožňuje vytvoriť súbor .war, ktorý je následne jednoducho nasaditeľný na hociktorý server s aplikáciou Tomcat.

**Nevýhody:**

**Výkon:** Pri porovnaní s niektorými inými ľahšími frameworkmi môže byť výkon Springu trochu nižší. Pre niektoré aplikácie môže byť toto obmedzenie nevýhodné.

### Svelte

Svelte je javascript framework slúžiaci na tvorbu užívateľských webových rozhraní. Jeho hlavnou prednosťou je, že narozdiel od iných populárnejších frameworkov nepoužíva tzv. Virtual DOM. To má za následok rýchlejšie načítanie stránky u klienta vďaka nižšiemu objemu odoslanému JavaScriptu, ako aj samotnému zrýchlenému chodu webovej stránky, pretože Svelte nemusí spravovať komponenty v klientskom JavaScripte, ale všetko je ponechané na webový prehliadač. Jeho ďalšou ohromnou výhodou je neprehľadnosť a intuitívnosť pri vytváraní rozhraní. Všetky premenné sú automaticky reaktívne a ich zmena sa hneď prejaví na stránke. Bolo by ešte na záver vhodné upozorniť, že svelte a svelte kit nie sú jedno a to isté. Zatiaľ čo svelte kit je framework pre vytváranie webových aplikácií, zahŕňajúci frontend a backend, svelte je len predpis ako písať jednotlivé komponenty ako navigačný panel alebo tlačidlo. Nakoľko už disponujeme java springom ako technológiou backendu, rozhodli sme sa pre svelte.

**Výhody:**

Prístupný

Ľahko pochopiteľný

Možnosť skompilovať ho na jeden javascriptový súbor

**Nevýhody:**

Malá komunita (v porovnaní s komunitou napr. frameworku React JS)

### Echarts

Apache ECharts je open-source JavaScriptový nástroj na vizualizáciu dát, ktorý beží plynulo na počítačoch aj mobilných zariadeniach. ECharts je založený na formáte svg, v ktorom poskytuje spôsoby na vytváranie intuitívnych, interaktívnych a vysoko prispôsobiteľných grafov.

### @vincjo/datatables

Táto knižnica ponúka rozhranie pre vytváranie dynamických tabuliek vo frameworku Svelte

### Tailwind CSS

Je framework na prácu s CSS (Cascading Style Sheets) založený na princípe utility-first (funkcionalita predovšetkým). Vybrali sme si ho nakoľko umožňuje používateľom vytvárať webové aplikácie rýchlejšie a jednoduchšie za cenu škaredšieho HTML kódu. Jeho silu sme pocítili hlavne pri tom, ako sme vďaka nemu vedeli jednoducho dodržiavať rovnaké veľkosti komponentov a ich odstupy. Nemuseli sme pritom opustiť zdrojový kód HTML, ktorý sme upravovali, čo nám značne urýchlilo vývoj užívateľského rozhrania.

### leaflet.js

Leaflet je open-source JavaScript knižnica určená pre tvorbu interaktívnych máp. Jej kompaktnosť a jednoduchosť implementácie, boli hlavné dôvody jej zvolenia v našom projekte.

### rollup.js

Nástroj slúžiaci na kompiláciu modulov JavaScriptu, do knižníc alebo celých aplikácií. Rozhodli sme sa ho spomenúť práve preto, nakoľko nám pomohol so spojením backendu v java springu a frontendu svelte, ktorý rollup.js skompiloval na single page application. Túto SPA sme potom dokázali vrátiť v Jave ako obyčajnú stránku s HTML a JavaScriptom.

### Server - VPS

Pri vývoji sme sa rozhodli mať najnovšiu verziu našej webovej stránky dostupnú na internete. Uľahčilo to vývoj, keďže sme obaja pracovali s rovnakými, aktuálnymi dátami z databázy a nekompatibilné kódy hardwaru a frontendu boli hneď odhalené. Ako VPS providera sme sa rozhodli pre hicoria.com, hlavne kvôli jej nízkej cene základného servera. Pre naše účely úplne postačovalo 1x CPU, 1024 MB RAM a 20 GB pamäte HDD za 22.25 € na 6 mesiacov. Na VPS serveri bola spustená SQL MariaDB databáza, Apache + Tomcat na hostovanie stránky ako aj Github Actions self-hosted softvér.

### MariaDB SQL + PhpMyAdmin

MariaDB je jedným z najpopulárnejších relačných databázových serverov na svete. Preto sme si ho bez váhania zvolili aj my. Všeobecne sa používa hlavne pre svoju rýchlosť, škálovateľnosť a robustnosť. Pri práci a úprave dát sme si zvoli grafické webové rozhranie phpMyAdmin. Je to bezplatný nástroj napísaný v jazyku PHP, určený na správu MySQL cez internet. phpMyAdmin podporuje širokú škálu operácií na MySQL a MariaDB.

### Apache server + Tomcat

Pri budovaní projektu sme si zvolili Apache Tomcat ako naše riešenie na deployment našeho systému. Tomcat nám poskytuje stabilný, výkonný a spoľahlivý kontajner pre Java aplikácie, ktorý sa dal jednoducho automatizovať viď. github akcie. Apache server slúžil na konfiguráciu ssl certifikátov ako aj proxy server, pretože tomcat bežal na porte 8080

### Github automatické akcie

Ako už bolo spomenuté, počas vývoja našej aplikácie bolo dôležité aby sme mali najnovšiu verziu našeho systému dostupnú na internete, vďaka čomu sme vedeli otestovať funkčnosť monitorovacieho zariadenia. Aby sme sa vyhli manuálnemu nasadzovaniu, rozhodli sme sa použiť službu github actions, ktorý je dostupný v rámci platformy github. Keďže náš repozitár nie je verejný, bolo nutné si vytvoriť self-hosted inštanciu na našom VPS serveri. Pri každom pridaní do našej hlavnej git vetvy bol kód automaticky skompilovaný nasadený na Tomcat.

## Finančná analýza

Súčiastky sme kupovali z rôznych zdrojov na základe najnižšej ceny, počas robenia projektu. Poštovné nie je započítané vo výslednej cene.

|  |  |  |
| --- | --- | --- |
|  | Cena v € | Obchod |
| Esp 32 | 9.85€ | techfun.sk |
| HX 711 | 1.50€ | techfun.sk |
| SIM modul | 9.30€ | techfun.sk |
| Senzor hmotnosti x 4 | 2.50€ \* 4 | AliExpress |
| Použitý plast pri 3D tlači | 1.00€ | Alza.sk |
| Konektory | 0,027€ \* 6; 0,02€ \* 24 | TMe.sk |
| Kabeláž | 0.50€ | AliExpress |
| Konektor typu C | 0.20€ | AliExpress |
| Plošný spoj | 4.50€ | techfun.sk |
| Papiere na plošný spoj | 1.50€ | techfun.sk |
| Materiály konštrukcie | 30€ | Hornbach |
| Rezistory | 0.45€ | techfun.sk |
| Nabíjací modul | 1.60€ | AliExpress |
| Batéria | 6.60€ | Vlastné zdroje |
| Pamäťové čipy | 2€ | techfun.sk |
| VPS server | 12 \* 3€/mesiac | hicoria.com |
| diódy | 0.1003 € | Vlastné zdroje |
| Výrobná cena našej váhy | 91.679€ |  |

Potrebný čas na vytvorenie našeho projektu je veľmi zložité vypočítať. Nakoľko každý na ňom pracoval vo svojom voľnom čase a počas týždňov sme mali rôzne veľa voľného času náš odhad nemusí byť vôbec presný Náš odhad by bol okolo 250 hodín.

# Výsledky práce

Môžeme bez pochyby zhrnúť, že s výsledkami našej práce sme nad mieru spokojní a to čo sa nám za ten čas podarilo sme veľmi hrdí. Dokázali sme vytvoriť funkčné multifunkčné monitorovacie zariadenie, ktoré spĺňa väčšinu požiadaviek aké sme si zvolili.

## Program pre mikrokontrolér

Program je písaný v jazyku embedded C a pre lepšiu prehľadnosť je čiastočne napísaný v triedach a objektoch. Zdrojové kódy sú organizované v header súboroch.

Zariadenie pracuje v režimoch ONLINE a IDLE. V ONLINE režime je zariadenie schopné vykonávať akcie hneď po ich vytvorení, avšak zariadenie by v ňom nemalo pracovať príliš dlho, pretože míňa oveľa viac energie ako režim IDLE.

## Pripojenie zariadenia k internetu

Zariadenie sa môže pripojiť k internetu 2 spôsobmi:

* Cez mobilnú sieť
* Pomocou WiFi

Na pripojenie cez WiFi sme používali knižnicu pre ESP 32 WiFi.h pomocou ktorej sa dá jednoducho komunikovať so vstavaným WiFi modulom.

Pre pripojenie na mobilnú sieť využívame Modul SIM800L. Komunikácia medzi ESP 32 a modulom zabezpečuje sériová linka. Modulu posiela ESP 32 príkazy a ak je akcia úspešná, vráti OK.

Zariadenie komunikuje so serverom žiadosťami vo formáte JSON.

## Napájanie

Napájanie pre mikrokontrolér zabezpečuje Li-Po 3.7V batéria, čo následne step-up prevodník zvýši na 5V. Súčasťou je aj ochranný obvod, aby sa zabránilo úplnému vybitiu batérie, čo ju poškodzuje, ako aj prebíjaniu. Zariadenie je možné nabíjať cez USB konektor typu C.

Aby sa predĺžila výdrž batérie, ESP 32 je v režime deep sleep, z ktorého sa zobúdza v intervaloch, kde prebehne aktualizácia dát. Dĺžku intervalu si užívateľ vie nastaviť.

## Aktualizácia dát

Aktualizácia prebieha v určených intervaloch alebo po žiadosti od používateľa sa vykonáva aktualizácia dát, kde zariadenie najprv nájde všetky pripojené senzory a iné zariadenia, postupne odmeria všetky hodnoty, pripojí sa na internet a pošle žiadosť na aktualizáciu stavu, prípadne správu cez websocket (pri online režime).

## BeeConnect - senzory a smart konektory

Okrem senzorov určených na meranie hmotnosti sa dajú k váhe pripojiť tieto dodatkové senzory a ďalšie zariadenia:

* Senzor na meranie teploty
* Senzor na meranie intenzity svetla
* Senzor na meranie hlasitosti zvuku
* Senzor na meranie vlhkosti
* Servo motor na pohyb častí včelieho úľa

Keďže je ich niekoľko druhov a každý včelár má iné požiadavky a preferencie, chceli sme mať modulárny systém, kde je ich možné ľubovoľne kombinovať ako taktiež aj neobmedzene pripojiť a odpojiť aj za prevádzky zariadenia. Preto sme vymysleli unikátny systém “BeeConnect” na inteligentné spravovanie senzorov. Tento systém funguje na princípe zbernice i2c a pamäťových čipov EEPROM (konkrétne 24C01CT-I/OT). Zámer bol, aby konektory neboli veľké, takže sme použili 4-pinové konektory typu JST. Prvý pin slúži ako konštantný 5V výstup zo zariadenia, stredové dva piny na komunikáciu s pamäťou a v istých prípadoch aj ako vstup/výstup napätie. Tieto tri piny sú medzi konektormi navzájom prepojené, takže s nimi nie je možné osobitne pracovať. Kvôli tomu je dôležitý štvrtý pin, ktorý slúži ako zem pre pamäťový čip a zároveň aj ako vstup na čítanie hodnôt z analógových senzorov. Všetky pamäťové čipy majú rovnakú adresu na zbernici i2c, aktivita pamäte závisí od napätia na štvrtom pine. Pre servo motor bolo potrebné použiť aj dva tranzistory, keďže cez digitálne plný ESP32 nemôže prejsť potrebný prúd pre pohnutie motora.

## Tlačidlo a LED

Tlačidlo na váhe sa používa na párovanie zariadenia (viď. nižšie) a na uspatie váhy. Keďže zariadenia nemá žiadny displej, rozhodli sme sa pripojiť kontrolnú LED, ktorá ukazuje aktuálny stav zariadenia.

## Plošné spoje

V zariadení sa nachádzajú dva vyrobené plošné spoje - hlavná doska a predný panel. V každom konektory senzora je tiež miniatúrny plošný spoj.

Všetky plošné spoje sme navrhli v softvéri Eagle a vyrábali pomocou papiera určeného pre prenos tonera z laserovej tlačiarne na plošný spoj. Po vytlačení sme tento papier položili na dosku, zažehlili a následne vložili do FeCl₃ (chloridu železitého). Po vyleptaní sme tento spoj museli očistiť, aby sa odstránil toner z tlačiarne. Keďže sa toner odstraňoval veľmi ťažko a nepomohol ani lieh, museli sme použiť čistiaci prostriedok a zoškrabať ho.

## Konštrukcia váhy

Našu váhu sme skonštruovali ako dva rámy, obe v tvare písmena H, ktoré sú umiestnené na sebe, pričom na každom ramene medzi dvoma rámami sa nachádza váhový modul. Ich vývody sú následne zoskupené v strede konštrukcie a cez zosilňovač potom zapojené jedným konektorom do krabičky.

## Krabička zariadenia

Krabička zariadenia bola vymodelovaná CAD software-om Inventor. Snažili sme sa ju vytvoriť čo najmenšiu a najpevnejšiu pre lepšiu manipuláciu so zariadením. Design krabičky bol inšpirovaný hernou konzolou nintendo, nakoľko obyčajný kváder nám prišiel príliš nudný. Krabička má odnímateľný vršok, usb c konektor na nabíjanie, 7 JSP konektorov pre senzory, ako aj názov produktu “BuzzyBees” napísaný na jej boku.

## Akcie

Pre lepšiu interakciu medzi zariadením a serverom sme sa rozhodli štandardizovať formát správ a implementovať systém akcií. Akcie odchádzajúce zo servera obsahujú pokyny pre zariadenie a delia sa na systémové a užívateľom vytvoriteľné. Každá akcia obsahuje typ, identifikačné číslo, čas kedy sa má vykonať a parametre vo formáte JSON.

Každá akcia pre zariadenie má svoj záznam v databáze s aktuálnym stavom, ktorý môže byť:

PENDING - akcia čaká na dotaz zo zariadenia

SENT - akcia bola odoslaná zariadeniu

SCHEDULED - akcia bola spracovaná zariadením a úspešne naplánovaná

DONE - akcia bola úspešne vykonaná

ERROR + typ chyby - pri vykonávaní akcie sa vyskytla chyba

Ak je čas spustenia nastavený na hodnotu 0, vykoná sa najskôr, ako je to možné zo strany zariadenia (závisí od aktuálneho režimu). Po vykonaní akcie príde zo zariadenia odpoveď v akcii pre server s typom “výsledok akcie” a identifikačných číslom a výsledkom akcie, pomocou ktorého server aktualizuje stav akcie.

## Rest API

Zdrojové kódy v JAVE pre Rest API sú rozdelené do kontrolérov podľa účelu. Niektoré automaticky požadujú session id od používateľa, aby nedošlo k neoprávnenému prístupu a nemuseli sme to v každej ceste implementovať. Zložitejšie algoritmy sme sa snažili písať do iných tried a v kontroléroch zdrojový kód minimalizovať, nakoľko sú tak lepšie prehľadné ľahšie upravovateľné. Pre každú odpoveď od servera slúži objekt nami vytvorenej triedy “Api Response” a Spring Boot ho automaticky prevedie na formát JSON.

## Štruktúra SQL (diagram)

Pri vytváraní štruktúry databázy bol použitý klient MySQL Workbench, nakoľko rozhranie v PhpMyAdmin nespĺňalo naše požiadavky.

## Párovanie nového zariadenia

Párovanie nového zariadenia môže byť realizované zadaním identifikačného čísla na webe (pri zariadení s GSM modulom)*.* Pri párovaní zariadenia pomocou identifikačného čísla je z bezpečnostných dôvodov potrebné stlačiť tlačidlo.

## Konfigurácia frontendu

Projekt používa osobitú kombináciu technológií a to Svelte a Java Spring Boot. V dôsledku toho bolo nutné vymyslieť ako sa tieto technológie spoja. Náš systém funguje na paradigme SPA - single page application. To znamená, že celá stránka sa pošle v jednom súbore a následne sa stránka mení dynamickým prepisovaním aktuálnej webovej stránky novými údajmi zo servera, namiesto štandardnej metódy načítavania celých nových stránok webovým prehliadačom. Cieľom je rýchlejšie prechody, ktoré robia webovú stránku pôsobiacu viac ako natívna aplikácia. Náš projekt zahŕňa dve takéto SPA stránky, úvodnú stránku a ovládací panel. Pre toto rozdelenie sme sa rozhodli kvôli nutnosti prihlásenia sa v ovládacom panely. Server tým pádom môže zamietnuť časť našej stránky ešte pred tým než sa pošle ku klientovi ak nie je prihlásený.

## Prihlasovanie / Registrácia

Systém prihlasovania funguje pomocou premennej session id v cookies, ktorú server dostáva spolu s requestom na stránku a ak užívateľ prihlásený, sám server ho vráti na login page. Pri ukladaní hesla používame hashovací algoritmus SHA-256 a frontend disponuje minimálnymi požiadavkami dĺžky a rôznorodosti znakov hesla .

## Domovská stránka

Na Domovskej stránke si potencionálny zákazník prečítať o našom multifunkčnom monitorovacom zariadení, čo dokáže a ako mu môže uľahčiť jeho prácu. V podstránkach je možné dozvedieť sa o tvorcoch tohto projektu ako aj prečítať si FaQ, často kladené otázky.

## Upraviteľná domovská stránka dashboard

Hlavná stránka dashboard bola vytvorená tak, aby bola čo najviac upraviteľná podľa požiadaviek užívateľa. Dashboard ponúka viacero možností úpravy. Užívateľ pri stlačení tlačidla upraviť uvidí zoznam možných kariet. Kliknutím na ne ich vieme postupne pridať do našeho ovládacieho panelu. Karty, ktoré nepotrebujeme vieme odstrániť pomocou červeného tlačidla vymazať. Karty vieme následne presúvať ako aj upravovať ich veľkosť. Každá karta obsahuje vlastné nastavenia, v ktorých je možné upravovať, z ktorých zariadení a aký typ dát bude karta zobrazovať. Pri väčšine kariet je možné zobrazovať dáta i z viacerých zariadení naraz.

## Čiarový graf

Jednou z hlavných kartičiek, je čiarový graf. Je zobrazený pomocou knižnice apache echarts a je nastaviteľný tak aby dokázal ukazovať hociktorý typ dát, ktoré sa doň vložia. Kartička dokáže byť nastavená tak aby dokázala vykresliť dáta z viacerých zariadení naraz ako i viac senzorov rovnakého typu. Taktiež je možné nastaviť si obdobie zobrazených dát.

## Mapa

Mapa je jednoduchá kartička na ktorej sú zobrazené umiestnenia všetkých zariadení podľa toho aké polohy boli zadané pri pridaní zariadenia do aplikácie. V budúcnosti by sme radi pridali aktívne sledovanie polohy. Zariadenie by tak dokázalo upozorniť majiteľa v prípade jeho odcudzenia.

## Aktuálna hodnota na senzore

Táto kartička ukazuje práve poslednú hodnotu senzora daného typu. Aj keď je na prvý pohľad jednoduchá. Bolo zložité upravovať veľkosť textu na základe veľkosti kartičky.

## Prehľad úľov a ich nastavenia

Stránka obsahuje tabuľku všetkých monitorovacích zariadení. V nej sú vyobrazené základné hodnoty každej z nich ako napríklad batéria, váha ako aj iné. Tabuľka obsahuje vyhľadávacie pole, ktoré dokáže nájsť konkrétne zariadenie podľa mena. Osobitou funkciu je exportovať, ktorá stiahne tabuľku vo formáte xlsx alebo csv.

## Upozornenia

Užívateľ si dokáže prezrieť všetky upozornenia na stránke, kde si ich po prečítaní vie označiť za pozreté alebo ich vymazať. Aby si ich však nemusel stále pozerať manuálne vyriešili sme ich pomocou push notifikácií. Na serveri sme použili knižnicu web-push na ich posielanie a kryptografickú knižnicu BouncyCastle pre bezpečné doručenie správnemu používateľovi.

## Kalendár

Na stránke sa nachádza kalendár, kde si včelár vie pridať pripomienky. Následne ak potvrdí “push notifications”, server v zadaní čas zašle upozornenie.

## Preklad do rôznych jazykov

Naša stránka ponúka preklad do viacerých jazykov. Tento systém bol vytvorený tzv. “from scratch”, inými slovami nepoužili sme žiadne knižnice. Na serveri je uložený .json súbor pre každý text v stránke, ktorý je následne odoslaný na frontend. Ten ho spracuje a následne je možné odkazovať na json ako cestu.

## Server - VPS

Náš celý projekt je spustený na VPS serveri od spoločnosti hicoria. Server s týmito parametrami stojí mesačne 3,06€, je to najlacnejšia konfigurácia servera, no na naše účely bohato postačuje.

Parametre virtuálneho servera sú nasledovné:

CPU: 1x XEON E7

RAM: 1024 MB DDR3

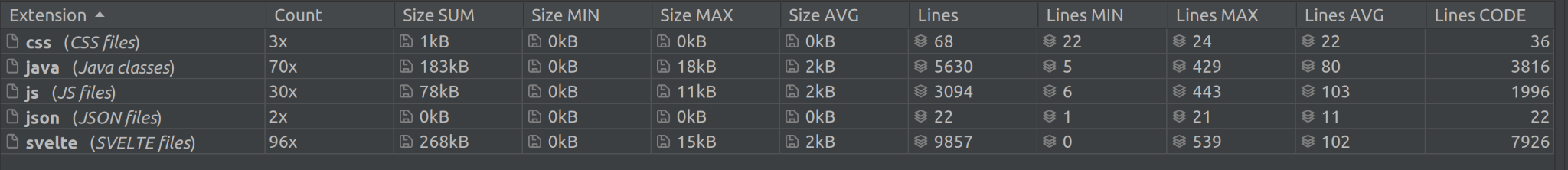
SSD: 20 GB SAS SATA3

OS: Debian Server 11 64bit

Stránka je spustená na systéme Tomcat, ktorý je potom proxy-hostovaný cez apache 2 na port 443 a pomocou certbota bol pridaný SSL certifikát aby bolo možné stránku posielať cez bezpečnejší protokol HTTPS oproti základnému menej bezpečnému HTTP.

## Štatistiky

Multifunkčný monitorovací systém má v čase písania dokumentácie dokopy 13796 riadkov kódu a 18671 riadkov celkovo (spolu s komentármi).



# Závery a zhrnutie

Remeslo včelárstva je dôležitá oblasť v poľnohospodárstve, ktorá prispieva k opeľovaniu plodín, biodiverzite ako aj podpore zdravých ekosystémov a vyvíjala počas stáročí spolu s človekom. Aj napriek tomu, že v dnešnej dobe má včelár možnosť kúpiť si rôzne doplnky pre svoje včelstvá, ktoré mu uľahčujú prácu s nimi, sme toho názoru, že remeslo včelárstva má veľký potenciál v inovácií, ktorá môže ponúknuť moderná výpočtová technológia súčasnosti.

Naše riešenie ponúka monitorovanie vonkajších vlastností včelích úľov ako napríklad hmotnosť, teplota, či vlhkosť úľu, ktoré sú vitálne pre zdravie včelstiev. Následné prehľadné zobrazenie na stránke, následne umožňuje včelárovi transparentne sledovať stav včiel, ako aj ich jednoduché spravovanie. Úlohou tohto riešenia je pravidelne zabezpečiť informácie pre včelára, aby nemusel fyzicky byť pri úli a vedel ho kontrolovať aj na diaľku.

Tento projekt je založený na technológií mikropočítačov. Vytvorené monitorovacie zariadenie predstavuje reprezentáciu tejto technológie, ktorá využíva mikroprocesory a periférne senzory. Kľúčovým prvkom je základná doska, ktorá riadi celý systém.

Hlavnými prioritami nášho projektu bola jednoduchosť používania a modularizácia, vďaka ktorej si každý včelár môže prispôsobiť zariadenia k vlastným potrebám. Je to možné vidieť na hardvéri a ako aj na softvéri. Na hardvéri to vidíme v podobe šiestich univerzálnych konektorov na bočnom paneli zariadenia. Toto riešenie zabezpečuje kompatibilitu periférnych zariadení, ktorá bola pre nás kľúčová. Modularizáciu / upraviteľnosť na strane softvéru môžeme nájsť na stránke, ktorá ponúka upraviteľný ovládací panel. Ten zabezpečuje relevanciu ukázaných dát včelárovi, nakoľko má možnosť upraviť, pridať či vymazať ľubovoľné údaje z ovládacieho panelu a nahradiť ich relevantnejšími. Základným prvkom tohto systému sú tzv. kartičky. Tie je možné v budúcnosti rozširovať podľa potrieb včelárov.

Vývoj nášho projektu nebol vždy úplne hladký a počas cesty sme sa stretli s viacerými výzvami. Jednou z nich bol napríklad spôsob prepojenia frontendu a backendu, nakoľko naša kombinácia technológií nie je až tak obvyklá a bolo za potrebu vymyslieť vlastný systém. Ďalším menším problémom no rovnako ako aj výzvou bolo aj výroba plošných spojov. Niektoré sme museli vyrobiť viackrát, pretože prvotné prototypy mali zopár nedostatkov v podobe zlých rozmerov alebo poškodených ciest. Nakoniec sa nám tieto výzvy podarilo prekonať a s výsledným produktom sme nad mieru spokojní.

Náš systém má pred sebou ešte dlhú cestu k dokonalosti. Veľmi radi by sme implementovali komplexnejšiu prácu s dátami, ako sú napríklad priemerné dáta počas istých úsekov, predpovede dát do budúcna, či rôzne vizualizácie údajov. Ďalším nedostatok nášho systému je v súčasnosti fakt, že každé zariadenie musí obsahovať SIM kartu ak chce byť pripojené na internet cez mobilné dáta. Elegantným riešením by podľa nás mohlo byť možno sériového zapojenia zariadení medzi sebou. Dokázali by tak komunikovať medzi sebou a celá komunikácia so serverom by bola uskutočnená cez jedno hlavné zariadenie so SIM kartou. Veríme, že naše inovácie prinesú reálnu hodnotu včelárom a pomôžu im v lepšom riadení a monitorovaní ich včelstiev, prispievajúc tak k udržateľnému a prosperujúcemu včelárstvu.

# Resumé

Our project aims to empower beekeepers by integrating modern technology into their beekeeping businesses and day to day operations, by helping them manage their beehives more wisely. What we did to solve this problem was creating a multi-functional monitoring device that keeps track of various measurements and presents them neatly on a personalized dashboard. Throughout the project development we had to create multiple specifically tailored technologies like peripheral device management, actions between server and device, beehive pairing or special editable modular dashboard modifiable for a user's preference. The result is a small compact multi-functional device with a custom designed case and real printed circuit boards inside. The device can be connected to the internet via WiFi or cellular network and offers the user remote access. We hope that our project will help beekeepers to take better care of their bee colonies and it will contribute to sustainable and successful beekeeping.

# Zoznam použitej literatúry

[1] TechFun. ESP32-WROOM-32U vývojová doska so socketom pre anténu [online]. Dostupné na internete:<https://techfun.sk/produkt/esp32-wroom-32u-vyvojova-doska-so-socketom-pre-antenu/>

[2] Rollup js. Introduction [online]. Dostupné na internete:<https://rollupjs.org/introduction/>

[3] Spring Framework. Spring Framework - Overview [online]. Verzia 3.2.x. Dostupné na internete:<https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/overview.html>

[4] Apache ECharts. Features [online]. Dostupné na internete:<https://echarts.apache.org/en/feature.html>

[5] GitHub. DataTables [online]. Dostupné na internete:<https://github.com/vincjo/datatables>

[6] Tailwind CSS. Tailwind CSS [online]. Dostupné na internete:<https://tailwindcss.com>

[7] Github. [online] <https://github.com/tncrazvan/svelte-3-jssr-starter>