

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Tvorba API pro portál Včelstva online

Hana Kratochvilová

© 2021 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Hana Kratochvilová

Systémové inženýrství a informatika
Informatika

Název práce

Tvorba API pro portál Včelstva online

Název anglicky

API Development for Web Portal Včelstva online

Cíle práce

Cílem bakalářské práce je návrh řešení API pro mobilní aplikaci portálu Včelstva online za účelem podpory a marketingu prodeje medu ze dvora.

Metodika

Po studiu vybrané literatury bude navrženo a zpracováno API, které poslouží pro mobilní aplikaci portálu Včelstva online. API slouží k rozšíření funkcionality webu a automatizaci určitých procedur.

Vzniklá aplikace bude sloužit pro včelaře zapojené do projektu Včelstva online k podpoře a propagaci prodeje medu ze dvora. Souběžně vznikají další absolventské práce, které mají za cíl vznik mobilní aplikace a které k tomu potřebují získávání dat.

Práce bude vznikat v rámci plánovaných výsledků projektu IGA PEF.

Doporučený rozsah práce

35-40 stran

Klíčová slova

api, php, chov včel, mobilní aplikace

Doporučené zdroje informací

BIEHL, Matthias, 2015. API Architecture: The Big Picture for Building APIs. 2. vydání. API-University. ISBN 978-1508676645.

JACOBSON, Daniel, Greg BRAIL a Dan WOODS, 2012. APIs: A Strategy Guide. USA: O'Reilly Media. ISBN 978-1-449-30892-6.

JIN, Brenda, Saurabh SAHNI a Amir SHEVAT, 2018. Designing Web APIs. USA: O'Reilly Media. ISBN 978-1-492-02692-1.

MITCHELL, Lorna Jane, 2016. PHP Web Services. Druhé vydání. USA: O'Reilly Media. ISBN 978-1-491-93309-1.

Předběžný termín obhajoby

2020/21 LS – PEF

Vedoucí práce

Ing. Jiří Brožek, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 26. 02. 2021

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Tvorba API pro portál Včelstva online" jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušila autorská práva třetích osob.

V Praze dne 15. 3. 2021

Poděkování

Ráda bych touto cestou poděkovala vedoucímu své bakalářské práce, panu Ing. Jiřímu Brožkovi, Ph.D. Dále bych ráda poděkovala panu doc. Ing. Janu Bartoškovi, Ph.D. za rady a čas, který mi věnoval a za možnost být součástí tak perspektivního projektu, kterým Včelstva online jsou. Dále bych ráda poděkovala Ing. Pavlu Junkovi za veškerou pomoc a rady při psaní kódu API.

Tvorba API pro portál Včelstva online

Abstrakt

Práce se zabývá vytvořením komunikačního rozhraní API pro portál Včelstva online, které bude v budoucnu sloužit pro propojení databáze portálu a mobilní aplikace. Budoucí mobilní aplikace i API samotné jsou součástí širšího projektu zaměřeného na podporu a propagaci prodeje medu ze dvora. Součástí práce je i podrobný popis funkcí portálu Včelstva online pro demonstraci, které informace a v jaké podobě jsou ukládány včelaři do databáze portálu. Následně jsou prostřednictvím API z databáze portálu Včelstva online tyto informace o prodejních místech medu, otevírací době, adrese prodejny a nabízeném sortimentu přenášeny. Dále je součástí API převod pravoúhlého trigonometrického formátu JTSK do formátu WGS84 využívaného v Google Maps, které budou pro mobilní aplikaci použity. Součástí práce je i detailní popis funkcí API, aby vývojáři budoucí mobilní aplikace měli co nejpřesnější informace o jeho fungování a mohli jej tak co nejefektivněji implementovat do systému.

Klíčová slova: API; PHP; MySQL; Nette; JSON; chov včel; mobilní aplikace; Včelstva online; JTSK; WGS84

API Development for Web Portal Včelstva online

Abstract

The Thesis focuses on the creation of a communication interface API for the portal Včelstva online, which will be used to connect the database of the portal and a mobile application. Future mobile application and API itself are part of a wider project aimed at supporting and promoting sale of honey from the yards. Part of the thesis is detailed description of the functions of the portal Včelstva online to demonstrate what information and in what form are saved by beekeepers in the portal database. Subsequently, this information about honey sales points, opening hours, store address and the offered assortment are transmitted via the API from the database of the Včelstva online portal. The API also includes conversion of the rectangular trigonometric format JTSK to the format WGS84 used in Google Maps, which will be used in the mobile application. Part of the thesis is also detailed description of API functions, so that the developers of the future mobile application have the most accurate information about its operation and can thus implement it as effectively as possible into the system.

Keywords: API; PHP; MySQL; Nette; JSON; beekeeping; mobile application; Včelstva online; JTSK; WGS84

Obsah

1 Úvod	11
2 Cíl práce a metodika	12
2.1 Cíl práce	12
2.2 Metodika	12
3 Teoretická východiska	13
3.1 HTTP.....	13
3.1.1 Webové transakce	13
3.1.2 Metody dotazování	13
3.1.3 Kódy odpovědí HTTP serveru	13
3.2 API	14
3.2.1 Co je to API	14
3.2.2 Proč používat API.....	15
3.2.3 Vytvoření API pro zakrytí implementace	16
3.2.4 Jak používat API.....	16
3.2.5 API na mobilních zařízeních.....	16
3.2.6 Veřejné a soukromé API.....	17
3.2.7 Podoba zprávy.....	17
3.2.8 JSON.....	18
3.2.9 Poskytovatelé API.....	18
3.3 Datové formáty pro uložení souřadnic	19
3.3.1 Formát JTSK.....	19
3.3.2 Formát WGS84	19
3.4 Využité nástroje	19
3.4.1 PHP	19
3.4.2 MySQL	21
3.4.3 Propojení PHP a MySQL.....	21
3.4.4 Výhody použití PHP společně s MySQL	22
3.4.5 MariaDB	22
3.5 Framework	22
3.5.1 Nette Framework	23
4 Praktická část	24
4.1 Požadavky na aplikaci.....	24

4.2	Požadavky na API – dle funkčnosti aplikace	24
4.3	Včelstva online	24
4.3.1	Včelař	25
4.3.2	Prodej medu	26
4.3.3	Pozorování včelstva – úlový deník	26
4.3.4	Zemědělec	28
4.3.5	Včelařský spolek	29
4.3.6	Technologie	29
4.3.7	Prodej medu na portálu Včelstva online	31
4.4	Mobilní aplikace	32
4.5	Prodej medu ze dvora	32
4.5.1	Propagace	32
4.6	Fungování mobilní aplikace	33
4.6.1	Stahování dat	33
4.7	Vytvoření API pro mobilní aplikaci	33
4.7.1	Potřeby mobilní aplikace	34
4.7.2	Zpracování požadavku na API	35
4.7.3	Odpověď aplikaci	38
4.7.4	Vrácení požadavku	39
5	Závěr	41
6	Seznam použitých zdrojů	42

Seznam obrázků

Obrázek 1	Moje stanoviště se včelami	25
Obrázek 2	Prodej medu	26
Obrázek 3	Kontroly včelstva a úlu	27
Obrázek 4	Úlový deník.....	27
Obrázek 5	Přidávání postřiků	28
Obrázek 6	Včelařský spolek.....	29
Obrázek 7	Portál Včelstva online	30
Obrázek 8	Prodej medu	31

Seznam tabulek

Tabulka 1	Nejpoužívanější HTTP metody	13
Tabulka 2	Prodejna medu v databázi	34

1 Úvod

Včelařství má v tuzemsku mnohaletou tradici a můžeme zde najít mnoho zkušených a kvalitních včelařů. Ti jsou schopni vyprodukovat velké množství medu, který mají nejen pro vlastní spotřebu, ale také jej často prodávají svým stálým zákazníkům. Velkou výzvou pro ně ovšem představuje prodej přebytků. Řada včelařů je totiž velmi zdatná v chovu včel, ale už méně v marketingu. Zároveň existuje celá řada lidí má zájem kupovat med přímo od včelaře, ale už nemá přehled o prodejnách medu a jeho aktuální dostupnosti. Za účelem vyřešení tohoto problému bude vytvořena mobilní aplikace, která přehledně zobrazí prodejny medu včelařů, kteří mají svou prodejnu zapsanou na portálu Včelstva online. Pro takovou mobilní aplikaci je však nejdříve nutné vytvořit komunikační rozhraní pro čtení dat z databáze portálu.

Tato práce si klade za cíl vytvořit takové komunikační rozhraní v podobě API (Application Programming Interface). Rozhraní by mělo zprostředkovat komunikaci v dostatečně rychlosti a objemu potřebném pro plynulý chod mobilní aplikace. Komunikace by měla probíhat přímo s databází portálu Včelstva online, kde jsou uloženy informace o adresách prodejen medu, nabízeném sortimentu a otevírací době. Přímé propojení s databází by navíc umožnilo aktualizaci informací samotnými prodejci medu, kteří jsou zaregistrovaní a mají zapsanou prodejnu na portálu Včelstva online.

Rešeršní část práce se zabývá samotnou technologií API, jejím vývojem a využitím. Dále jsou rozebrány nástroje a programovací jazyky, které jsou neopomenutelné při vytváření tohoto aplikačního rozhraní pro komunikaci. Praktická část pak rozebírá fungování samotného portálu Včelstva online, pro demonstraci možností registrovaných včelařů a údajů, které budou vkládány do databáze a následně odesílány přes API do mobilní aplikace. V druhé polovině praktické části je popsáno vytvořené API pro portál Včelstva online a jakým způsobem pracuje. Tento popis by měl následně sloužit vývojáři mobilní aplikace jako návod, jakým způsobem má s API pro komunikaci s databází portálu Včelstva online zacházet pro efektivní využití ve své aplikaci. Součástí API by měl být i převod pravoúhlého trigonometrického formátu JTSK do formátu WGS84 využívaného v Google Maps, které budou pro mobilní aplikaci použity.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem bakalářské práce je návrh řešení API pro mobilní aplikaci portálu Včelstva online za účelem podpory a marketingu prodeje medu ze dvora.

2.2 Metodika

Po studiu vybrané literatury bude navrženo a zpracováno API, které poslouží pro mobilní aplikaci portálu Včelstva online. API slouží k rozšíření funkcionality webu a automatizaci určitých procedur.

Vzniklá aplikace bude sloužit pro včelaře zapojené do projektu Včelstva online k podpoře a propagaci prodeje medu ze dvora. Vytvořené API by mělo přenášet potřebná data v adekvátním formátu pro potřeby mobilní aplikace. Souběžně vznikají další absolventské práce, které mají za cíl vznik mobilní aplikace a které k tomu potřebují získávání dat. Práce bude vznikat v rámci plánovaných výsledků projektu IGA PEF.

Při vytváření bakalářské práce bylo postupováno dle následujícího harmonogramu:

- leden až březen 2020: studium vybrané literatury související s tématem práce
- březen až červen 2020: konzultace k potřebným parametrům a obsahu API
- duben až září 2020: tvorba a implementace API do portálu Včelstva online
- září až listopad 2020: diskuze výsledků a výstupů v rámci projektu IGA
- srpen až prosinec 2020: sepsání a finalizace praktické a řešeršní části práce

3 Teoretická východiska

3.1 HTTP

Webové prohlížeče, servery a webové aplikace spolu komunikují skrze Hyper Text Transfer Protocol (HTTP), který přenáší data v textovém tvaru. Jelikož HTTP používá osvědčené přenosové protokoly, garantuje, že data nebudou nijak poškozena při přenosu. Jedná se tedy o spolehlivý přenos (Gourley, 2002, s. 3).

3.1.1 Webové transakce

HTTP transakce se skládá z požadavku, který je poslán klientem na server, a odpovědi, která je poslána serverem zpátky klientovi. Tato komunikace se děje skrze formátované bloky, které se nazývají HTTP zprávy (Gourley, 2002, s. 8).

3.1.2 Metody dotazování

HTTP podporuje několik požadavkových příkazů, které se nazývají metody. Každý HTTP požadavek má metodu. Metoda říká serveru, kterou akci má provést (tabulka 1).

Tabulka 1 Nejpoužívanější HTTP metody

metoda HTTP	popis
GET	poslání požadavku na data ze serveru
PUT	požadavek zaslání dat na server
DELETE	smazat objekt serveru
TRACE	zaslání kopie požadavku zpět klientovi
HEAD	pošle jen HTTP hlavičku z odpovědi konkrétnímu zdroji

Zdroj: Gourley, 2002

3.1.3 Kódy odpovědí HTTP serveru

Každá HTTP odpověď vrací zprávu s návratovým kódem. Návratový kód je složen ze tří číslic a poskytuje klientovi informaci, jestli žádost byla úspěšná. První číslice udává charakter kódu: 1 značí informační, 2 kladné, 3 značí přesměrování, 4 informují o problému na straně klienta, 5 problém na straně serveru. Častými jsou např. 200 – OK (vše proběhlo úspěšně), 404 – Not Found (nelze najít, chyba na straně serveru) (Gourley, 2002, s. 9).

3.2 API

Programové aplikační rozhraní (API z *angl.* Application Programming Interface), je aplikační rozhraní, kterým jeden program prezentuje určitá data jiným programům. Smyslem API je zprostředkování komunikace mezi dvěma platformami, které si vyměňují data. Jeho hlavní výhoda spočívá v použití naprogramovaného způsobu komunikace a začlenění jej do nového programu. Není tudíž nutné vymýšlet pokaždé něco nového a šetří se tak čas (Jin a kol., 2018, s. 15).

3.2.1 Co je to API

Aplikační rozhraní API nabízí jednoduchý způsob pro propojení, integraci a rozšiřování softwarového systému. API je způsob, kterým spolu komunikují aplikace skrze síť (většinou internet) a používají jazyk, kterému oba rozumí. Je to obecný koncept, který může být použit více způsoby, a proto pro něj existují různé definice (Jacobson, 2012, s. 5):

- API je webová služba, která poskytuje zdroje skrze webové technologie, jakou je např. HTTP. API jsou používána na stavění distribuovatelných softwarových systému a umožnění volného propojování.

- API je jednoduché, jasné a přístupné rozhraní, ke kterému se mohou různé aplikace jednoduše připojit.

- API poskytuje most mezi společnostmi, interní data společností jsou poskytována externím zákazníkům. (Biehl, 2015, s. 4)

Aplikační rozhraní (API) jsou základní součástí našeho propojeného světa. Software používá tato rozhraní ke komunikaci od aplikací na mobilních telefonech po vzdálené servery, API jsou všude. Ať už je na ně pohlíženo jako na jednoduchá technická rozhraní, nebo produkty samy o sobě, celé systémy na ně spoléhají. Stejně tak na ně spoléhají celé společnosti a organizace od technických startupů po internetové giganty, ale i netechnické malé a střední podniky, velké korporace a vládní instituce (Lauret, 2019, s. 3). API kteréhokoliv typu zjednodušují tvorbu softwaru. API kteréhokoliv typu je předně a zejména rozhraní. Bod, kde se setkávají dva systémy, objekty, organizace a interagují spolu (Lauret, 2019, s. 4).

Většinou API není přímo viditelné, nenabízí uživatelské rozhraní a typicky s ním nemůže komunikovat koncový uživatel, ale naopak operuje pod povrchem a přímo komunikuje s jiným softwarem (Preibisch, 2018, s. 3). Je používáno ve strojích, ve strojové komunikaci a k integraci dvou a více softwarových systémů. Jediný, kdo přímo pracuje s API jsou vývojáři vytvářející aplikace a programy s tímto API. To je důvodem, proč API potřebují být vytvářena s ohledem na to, kdo je bude používat pro své aplikace. Zároveň musejí být jednoduchá a efektivní. Bývalo běžné se při vytváření služby orientovat na potřeby poskytovatele služby, ale API by měla být navrhována dle potřeb spotřebitelů (Biehl, 2015, s. 4).

API je rozhraní, kterým softwarový program prezentuje data jiným programům a lidem. Protože jsou API navrhována pro práci s ostatními programy, záměrem je, aby byla srozumitelná a používána lidmi, kteří ty ostatní programy vytváří. Vytvořená API jsou základem pro budování webových platforem. Je to způsob, jakým je vytvořena identita a uchována skrze účty, od emailové adresy společnosti po webové aplikace umožňující objednat pizzu. API je způsob, jakým je například předpověď počasí z National Weather Service sdílena se stovkami aplikací specializujícími se na její prezentaci. API jsou důležitou součástí velkých společností jako je Amazon, Google a Facebook (Brenda, 2018, s. 1).

3.2.2 Proč používat API

API nabízí jednoduchý způsob pro připojení a integraci se softwarovými systémy. Softwary můžeme najít všude, od bank a trhů až po stroje a auta. Přesto jsou tyto softwarové systémy typicky izolovány a funkce jednoho systému nejsou přístupné jinému systému. Cílem API je propojení těchto separátních softwarových jednotek. Zároveň API poskytuje kompatibilitu mezi těmito softwary, která je zásadní pro jejich připojení, rozšiřování a integraci. Při použití propojení dvou nebo vícero softwarů přes API můžeme propojit podnik s podnikem, podnik se svými produkty, služby se svými produkty a produkty přímo s dalšími produkty (Biehl, 2015, s. 4).

Společnosti nemohou nutit firemní partnery k užití svých služeb. Cílem je tedy vytvořit natolik kvalitní služby, že je tito partneři budou chtít používat sami od sebe. Služba by měla být: užitečná a pomáhat partnerům ve výkonu jejich podnikání, přesně

vyhovující potřebám partnerů, dále být jednoduchá a srozumitelná, snadno integrovatelná a monitorovatelná partnery, bezpečná, spolehlivá a výkonná (Biehl, 2015, s. 5).

API jsou používána pro interní a externí integraci, jelikož umožňují lepší propojení mezi odděleními podniku. Například společnost Amazon používá API pro integraci informačních systémů svých oddělení. Když jsou rozhraní a technologie již k dispozici pro interní integraci, je snazší je poskytnout externě. Externí API jsou také nezbytná pro realizaci mobilních aplikací. Mobilní aplikace používají jinak těžko přístupná data podniku, která jsou k nim doručena právě skrze API. Poskytnutí externího přístupu ke zdrojům umožňuje developerům třetích stran, kterým společnost nemusí platit, aby vytvářeli inovace pro danou společnost (Biehl, 2015, s. 5).

3.2.3 Vytvoření API pro zakrytí implementace

Kvalitní návrh API je důležitý, jelikož lidé používající API jej chtějí užívat bez toho, aby byli obtěžováni a vyrušováni zbytečnými detaily, které jsou pro ně nadbytečné. A proto návrh API musí zakrýt detaily implementace, tj. co se ve skutečnosti děje. Běžného uživatele nezajímá celý proces, chce se pouze co nejrychleji dostat k výsledku. Jako každé rozhraní je API vytvářeno pro své uživatele, aby jim pomohlo dosáhnout toho, co potřebují. Cíle, které mohou být použitím daného API dosaženy musí být srozumitelné pro uživatele (Lauret, 2019, s. 10).

3.2.4 Jak používat API

API jsou stavěna, aby byla snadno použitelná. Nejsou přímo určena pro koncové uživatele, ale jsou integrována do aplikací, které jsou následně nabízeny koncovým uživatelům. Kompletní řešení, které API použije, se obvykle skládá z:

1. Klienta, nebo aplikace, která volá API a zpracovává data poskytnutá API, tento klient je zodpovědný za koncový výsledek.
2. Několika API, která poskytují data pro aplikaci.
3. API platformy, která spravuje API (Biehl, 2015, s. 6).

3.2.5 API na mobilních zařízeních

Počet mobilních zařízení a tabletů již překročil počet počítačů. Mobilní aplikace se liší od standardních aplikací pro stolní počítače tím, že většina mobilních aplikací není

ani autonomní, ani soběstačná. Aplikace potřebují připojení k serverům na internetu, aby byly použitelné, nebo alespoň byly plně využity jejich funkce (Biehl, 2015, s. 14).

Mobilní aplikace sama o sobě je často jen uživatelské rozhraní. Některé aplikace provádí logické operace přímo na mobilních telefonech, ale ve většině případů jsou tyto výpočty prováděny na serverovém úložišti, nebo v datovém centru společnosti. Funkcionalita poskytovaná těmito servery je přístupná skrze API. Také data jsou uložena na straně serveru a stahována skrze API požadavky. Data, která jsou sbírána od uživatelů jsou většinou poslána API na serveru, které předá data službě, nebo je uloží přímo do databáze. Data posílaná přes API musí mít malou velikost a být dělitelná. Tím se zajistí, že API může být použito i zařízeními s limitní výpočetní silou a limitovaným připojením k internetu (Biehl, 2015, s. 15).

3.2.6 Veřejné a soukromé API

API se vyskytují ve dvou typech, veřejné (*angl. public*) a soukromé (*angl. private*). Na veřejné API je nahlíženo jako na službu, nebo na produkt. Veřejná API jsou poskytnuta komukoliv, kdo souhlasí s podmínkami užití. Za tato API se může platit, nebo může být zdarma všem přístupné (Lauret, 2019, s. 9).

Veřejná API však tvoří jen velmi malou část. Daleko větší je tvořena soukromými API. Soukromé API se vytvoří pro interní účely, tj. jen pro aplikace vytvořené programátory pro potřeby společnosti samotné. V tomto případě je shodný poskytovatel API a uživatel API. V otázce soukromé/veřejné nezáleží na tom, jak je API vystaveno, ale hlavně komu (Lauret, 2019, s. 9).

3.2.7 Podoba zprávy

Pro sdělení stavu identifikovaného zdroje požadavku vyžádané zprávy používá API tzv. tělo entity zprávy. API využívá textový formát pro reprezentaci stavu zdroje jako sady polí. Mezi nejpoužívanějšími formáty patří Extensible Markup Language (XML) a JavaScriptový objektový zápis (JSON). XML stejně jako HTML organizuje informace v dokumentu vnořenými párovými závorkami. Dobře naformátovaný soubor XML musí mít závorky perfektně spárované, tzn. všechny závorky správně ukončeny. Tímto způsobem párování drží XML strukturu dokumentu. JSON používá složené závorky pro hierarchické oddělení struktury informací dokumentu. Většina programátorů je zvyklá

na tento styl výrazů, čímž je snadný pro všechny, kteří jsou zvyklí na objektové programování (Massé, 2011, s. 47).

3.2.8 JSON

Jako formát pro datovou výměnu JSON podporuje lehkou a snadnou spolupráci. Dnes je to velmi populární formát, který je velmi často používán při vytváření API. JSON si zapůjčil části syntaxe z JavaScriptu a zároveň těží z bezproblémové integrace s prostředím prohlížeče. Když není dopředu definován standardní datový formát pro zdroj, je doporučeno používat pro strukturu informací formát JSON. Ten je složen z objektů, které mají sety dvojic: název a hodnota. Syntaxe JSON objektu definuje názvy proměnných ve formátu řetězců, které jsou vždy ohraničeny dvojitými uvozovkami nahoře. Toto pravidlo je přísnější než u JavaScriptu, a proto kvůli své podobnosti může vést ke špatnému formátování JSON (Massé, 2011, s. 47).

Příklad – správně naformátovaný JSON:

```
{
  "jmeno": "Jan",
  "prijmeni": "Kudrna",
  "vek": 48,
  "vzdelani": {
    "stredni" : "Gymnázium Slaný",
    "univerzitetni" : "Česká zemědělská univerzita",
  }
}
```

3.2.9 Poskytovatelé API

API jsou používána různými způsoby, pro interní použití u interních systému ve firmě, nebo externí použití pro partnery společnosti. Poskytovatelé je nabízejí pro externí použití komukoliv, kdo odsouhlasí příslušné obchodní podmínky. Veřejné API od společností Twitter a Facebook určená pro vývojáře patří mezi nejznámější. Přestože by se mohlo zdát, že takováto veřejná API určená pro všechny jsou nejrozšířenější, není tomu tak. Častější jsou ta se zaměřením na použití partnerskými společnostmi, nebo jen na interní použití (Jacobson, 2012, s.19).

Například Twitter, který je poskytovatelem sociální sítě, má tři úrovně API. Standardní je poskytnuta všem vývojářům aplikací, má ale omezení na počet požadavků,

kteřé aplikace pošlá. Premium úroveň je zpoplatněna a nabízí více služeb. Nejvyšší úroveň nabízí přístup k datům Twitteru, a to včetně 30denního archivu (Using the Twitter API).

Na území Evropské unie je od roku 2018 účinná Směrnice o platebních službách č. 2015/2366, kterou vydal Evropský parlament. Tato směrnice nařizuje všem bankám působících v zemích EU zdarma zpřístupnit svá API. Banka tedy po získání souhlasu klienta musí bezpečně zpřístupnit data dalšímu subjektu, např. bance nebo obchodu, k bankovním platebním službám. Banky mají možnost buď poskytnout API, nebo screen scraping – systém se připojuje za jinou osobu, strojově přečte, co je na obrazovce a vyplňuje textová pole (Směrnice Evropského parlamentu a Rady (EU) 2015/2366, 2015).

3.3 Datové formáty pro uložení souřadnic

3.3.1 Formát JTSK

Formát Jednotné trigonometrické sítě katastrální (JTSK) je založen na jednotlivých katastrálních územích vymezených trigonometrickou sítí. Tato síť byla původně vytvořena pro území České republiky a Slovenska. Pokud se pracuje s mapou a je žádoucí zobrazení dat pouze z ČR, je výhodné použít tuto síť, jelikož se jedná o pravoúhlý systém. Na tak malém území jako je ČR, není potřeba započítávat zakřivení Země (Šimbera, 2018, s. 6).

3.3.2 Formát WGS84

Systém World Geodetic System 1984 (WGS84) není definován geometricky, ale moderněji tj. fyzikálně. Pro přesnější určování polohy na zemském povrchu bylo potřeba zpracovat nepravidelný tvar Země. „Jako normální zemské těleso byl podle Pizzettiho zvolen hladinový rotační elipsoid, jehož základní parametry, velikost, tvar, hmotnost a rychlost rotace, byly odvozeny z pozorování, zejména pak z pozorování družic.“ (Jurkina, 2006, s. 3)

3.4 Využití nástroje

3.4.1 PHP

PHP je všudypřítomný jazyk kompatibilní s většinou operačních systémů. Je snadné se jej naučit, čímž je ideálním nástrojem pro začátečníky programování webových aplikací

(Davis, 2007, s.4). Tento programovací jazyk umožňuje vkládat dynamickou aktivitu na webové stránky. PHP je jazyk, který se používá, aby byl vytvořen dynamický výstup ze serveru. Tento výstup je potenciálně jiný pokaždé, když prohlížeč požádá o stránku (Nixon, 2014, s. 35). Využití nachází tento programovací jazyk při tvorbě webových aplikací. Je poskytován zdarma, je výkonný a snadno použitelný. Lehce se jej naučí každý. Také má rozšíření a k jeho použití je k dispozici různé frameworky, které umožňují vytvořit celou řadu dynamických funkcí (Forbes, 2012, s. 3).

Většina webových stránek na internetu je pouze statická, což znamená, že je tvořena textem, organizována určitým způsobem, kterým je prohlížeč zobrazuje. Statické stránky se mění pouze, pokud je osoba upraví (Forbes, 2012, s. 3). Soubor, který byl upraven předtím, než byl odeslán prohlížeči, se nazývá dynamická webová stránka. Dynamická webová stránka se může lišit pokaždé, kdy je zobrazena prohlížečem, jelikož server upravuje stránku předtím, než ji pošle prohlížeči ke zobrazení uživateli. Stránka je upravena dle instrukcí, které programátor zadal pro tuto stránku (Forbes, 2012, s. 4).

Soubor PHP je ukládán s příponou .php. Tato přípona říká serveru, na kterém stránka běží, že předtím, než odešle soubor prohlížeči, se má podívat dovnitř, jestli se není zapotřebí něco upravit. PHP jazyk je používán pro vytváření dynamického obsahu webové stránky. Běžný text PHP stránky je statický ale server dynamicky převede text, který se objeví ohraničený `<?php ?>`, do statické podoby poté, co vyhodnotil, co kód znamená (Forbes, 2012, s. 5).

Programovací jazyk PHP byl původně vytvořen Rasmusem Lerdorfem v roce 1995. Hlavní implementace je dnes vydávána PHP Group a slouží jako formální vzor pro jazyk PHP. PHP je software, který je poskytován zdarma pod licencí GNU General Public License (GPL), která zahrnuje restriktce pro použití PHP. Původně zkratka PHP znamenala Personal Home Page, dnes se uvádí jako význam Hypertext Preprocessor (Forbes, 2012, s. 6).

PHP je jednoduchý a zároveň výkonný programovací jazyk pro vytváření obsahu HTML. Původně byl tento jazyk navržen pro vytváření dynamického obsahu webových stránek a pro tento úkol je stále vhodný. Pro generování HTML je nutné mít PHP analyzátor a webový server, skrze který se posílá zdrojový kód. PHP je také populární pro generování dynamického obsahu z databázového připojení, XML dokumentů, grafiky a PDF souborů. Dále může být PHP použito jako skriptovací jazyk pro příkazovou řádku

pro systémové administrativní úkoly, např. pro zálohování. PHP běží na většině operačních systémech, od variant Unixu po Windows a macOS. Může být použito na serverech Apache, Nginx a Open BSD (Tatroe, 2020, s. 1,2).

Jedná se o jazyk velmi flexibilní a nelimitovaný. Jeho výstupem nemusí být pouze textové soubory jako HTML, ale může být generován jakýkoliv formát: PDF, GIF, JPEG a PNG. Velkou výhodou PHP je pak široká podpora databází. PHP podporuje všechny velké databáze včetně MySQL, PostgreSQL, Oracle, Sybase, MS-SQL, DB2 a ODBC-compliant, ale i mnoho menších. Dále PHP poskytuje knihovny PHP kódu pro vykonávání častých úkolů, jako je databázové připojení, řízení chyb a další. (Tatroe, 2020, s. 2, 3).

3.4.2 MySQL

MySQL je zdarma poskytovaná relační databáze. MySQL byla vyvinuta v 90. letech, aby efektivně zpracovávala informace v počítačích (Davis, 2007, s. 6). S více než 10 miliony instalací je MySQL pravděpodobně nejpoužívanějším databázovým systémem pro webové servery. Databáze je strukturované uložení záznamů v počítačovém systému a záznamy jsou organizovány tak, že mohou být rychle vyhledány (Nixon, 2014, s. 165).

Databáze MySQL obsahuje jednu, nebo více tabulek, každá z nich pak obsahuje záznamy, nebo řádky. V těchto řádcích jsou sloupce, nebo pole, které obsahují samotná data (Nixon, 2014, s. 165) . MySQL má vlastní uživatelské rozhraní, které umožňuje přesouvat data a měnit databázovou konfiguraci (Davis, 2007, s. 122).

3.4.3 Propojení PHP a MySQL

PHP je výborným nástrojem pro vytváření webových aplikací, ale samo o sobě neuchovává informace, k tomu je potřeba databáze. Většinou je spolu s jazykem PHP používán databázový systém MySQL, který uchovává informace uživatelů vložené skrze PHP. MySQL automatizuje většinu běžných úloh spojených s uchováváním informací a jejich zobrazováním v závislosti na kritériích, je snadno přístupné z PHP a společně fungují velmi dobře. Další výhodou je, že PHP i MySQL běží na celé řadě různých typů počítačů a operačních systémů včetně Mac OS X, Windows a Linux (Davis, 2007, s. 2).

3.4.4 Výhody použití PHP společně s MySQL

PHP a MySQL fungují dohromady velmi dobře, jelikož byly cíleně vyvíjeny souběžně a bylo s nimi počítáno pro společné použití. V důsledku tohoto cíleného vývoje je programovací rozhraní mezi nimi logicky spárováno (Davis, 2007, s. 3). Obojí je licencováno pod open source licencí, mohou tedy být zdarma kýmkoliv použity. Zkušenější uživatelé mohou provádět změny ve zdrojovém kódu, a tak měnit způsob, jakým jazyk a program fungují. Kolem PHP a MySQL se vytvořila široká komunita na internetu, ke kterým se může kdokoli přidat a členové mu pomohou s jeho problémy. PHP a MySQL mají jednoduchý a efektivní design umožňuje rychlé zpracování procesů (Davis, 2007, s. 3).

3.4.5 MariaDB

Databáze MariaDB se oddělila od MySQL v roce 2010. Jedná se o velmi podobnou databázi, jen nabízí více funkcí a má větší potenciál vývoje (Dyer, 2015, s. 10). Výhodou této databáze je podpora uložení geografických a geometrických dat a zároveň poskytuje funkce pro práci s tímto typem dat od verze 10.2.8 vydané v roce 2017 (aktuální verze je 10.5.9) (MariaDB, 2021).

3.5 Framework

Framework je platforma pro vývoj softwarových aplikací. Poskytuje základy, na kterých mohou vývojáři softwarů vytvářet programy. Framework může obsahovat předdefinované třídy a funkce, které mohou být použity na zpracování vstupů, správu hardwarových zařízení a interakci se systémovým softwarem. Tyto předem připravené funkce zjednodušují proces tvorby, jelikož programátoři nemusejí vše pokaždé vymýšlet, když vyvíjejí novou aplikaci. Služby frameworku slouží jako základ programování, dále pak obsahuje knihovny, kompilátor a další programy používané při vývoji softwaru. Webový framework na straně serveru je softwarový framework, který usnadňuje psaní a údržbu webových aplikací. Poskytuje nástroje a knihovny, které usnadňují běžné úkoly vývojářů včetně vhodného přesměrovávání URL adres, interakcí s databází, uživatelskou autorizací, formátování výstupů (např. HTML, JSON, XML) a také zlepšení zabezpečení proti webovým útokům (Framework, 2013).

3.5.1 Nette Framework

Nette je český framework, který usnadňuje práci předpřipravenými třídami, funkcemi a přehledným formátováním kódu. Poskytuje šablonovací systém, ladící nástroje, databázovou vrstvu a zabezpečení před zranitelnostmi. Podporuje formáty HTML5, AJAX, a je poskytován zdarma (Dokumentace Nette 3.1, 2021).

Tento framework funguje podobně jako většina PHP frameworků. Nette rozdělí architekturu stránky na následující části: model, presentery a šablony. V adresáři Model jsou uloženy třídy (modely), ve kterých je obsažena aplikační logika a které jsou určeny zejména pro interakci s databází. Tyto třídy (modely) komunikují pouze s presentery a jinak jsou odděleny od zbytku aplikace. Šablony pro webové stránky jsou umístěny v adresáři Templates. Dané šablony jsou řešené formou Latte systému, který chrání výstup před zranitelnostmi, šetří čas, jelikož poskytuje mnoho funkcí zrychlujících práci a funkcí vytváření přehledného kódu na rozdíl od spojování PHP a HTML jazyků bez použití frameworku. Latte zároveň urychluje PHP zápis, předkládá své šablony přímo do jazyka PHP. Další částí architektury je v adresáři Presenters. Zde se nachází řadič, který zařizuje zpracování uživatelských požadavků. Podle požadavků řadič zavolá model a následně požádá o vykreslení výsledku skrze Latte šablony. Presenter třídy je složen z rendrovacích metod pro předávání dat z databáze do šablon (Dokumentace Nette 3.1, 2021).

Součástí Nette je i ladící nástroj Tracy, který pomáhá s rychlým hledáním a opravováním chyb. Jestliže se v kódu vyskytuje chyba, stránka se nenačte, ale Tracy vypíše chybovou stránku, kde je možné vidět, ve kterém řádku, kterého souboru nastala chyba (Dokumentace Nette 3.1, 2021).

4 Praktická část

4.1 Požadavky na aplikaci

Pro rozšíření funkcí portálu Včelstva online bude vytvořena mobilní aplikace využívající data z prodeje medu. Cílem je podpora prodeje medu ze dvora zaměřením se na pomoc včelařům s marketingem. Aplikace by měla být vytvořena pro použití co nejširší možnou veřejností, a proto by měl být kladen důraz na snadné na použití i pro starší lidi, jelikož zákazníci včelařů jsou silně heterogenní skupina napříč celou populací.

4.2 Požadavky na API dle funkčnosti aplikace

API by mělo zprostředkovávat přenos dat pro mobilní aplikaci z portálu Včelstva online. Mobilní aplikace bude potřebovat souřadnice prodejen, informace o prodejnách a prodeji medu.

4.3 Včelstva online

Včelstva online je portál, který hobby včelařům poskytuje nástroje pro evidenci včel. Portál je nabízen zdarma pro odbornou i laickou veřejnost. Cílem portálu je podpořit chov včel a zajistit jejich ochranu v krajině na území ČR. Z dlouhodobého hlediska pak je cílem portálu podpořit rozvoj aplikací a moderních technologií v oblasti chovu včel pro podporu precizního zemědělství v ČR. Prvotním impulsem k vytvoření platformy bylo zprostředkování vzájemné komunikace mezi včelaři a zemědělci, protože tyto dva obory se vzájemně velmi silně ovlivňují. Na portálu je proto umožněno anonymní dohledávání včelstev v okolí plánovaných postřiků podle půdních bloků.

Přístup je portál možný bez registrace a plní především informační funkci. Uživatelům je umožněno vyhledávat v mapě prodejny medu, sledovat data z úlových vah referenčních včelařů a také jsou poskytnuty informace o včelách a o medu. Pro registrované uživatele je však dostupných více funkcí, např. vedení úlového deníku, sdružování se do včelařských spolků, uložení a kontrola dat z vlastní úlové váhy. Registrace na portál je možná z pozice včelaře, zemědělce, nebo včelařského spolku. Portál Včelstva online se snaží být otevřen pro všechny výrobce vah a jejich příslušenství. Kterýkoliv výrobce může požádat o zařazení do seznamu úlových vah pro připojení.

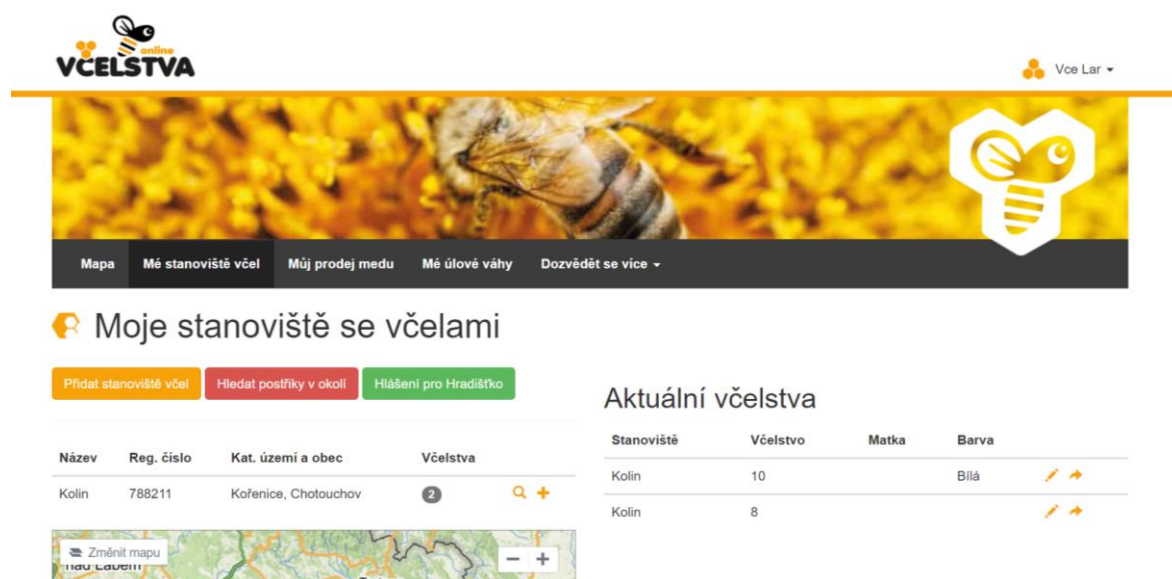
4.3.1 Včelař

Včelařům jsou k dispozici nástroje pro evidenci stanovišť včelstev, vedení úlových karet, dále také usnadnění komunikace se státními úřady skrze funkce generování PDF formulářů pro hlášení na „Hradištko“, nebo na obecní úřad.

Po registraci může ihned včelař používat dostupné funkce. Může si přidat stanoviště se svými včelstvy a na stanovišti si přidělit počet umístěných včelstev (obrázek 1). Po zapsání je stále možnost kdykoliv stav změnit, v případě, že včelstvo bylo přemístěno, nebo nastala změna počtu včelstev. Na stránce stanoviště jsou zobrazovány informace o aktuálně umístěných a v okolí se nacházejících včelstvech. Mimo to mohou být zobrazeny informace o včelstvech, které na stanovišti byly umístěny historicky. Údaj o včelstvech v okolí je anonymní, včelařům je pouze zobrazována informace, kolik je v okolí do 10 km od jeho stanoviště cizích stanovišť se včelími úly a kolik je tam umístěno včelstev.

Na stránce o stanovišti je také poskytnuta možnost zapsání fenologického pozorování, tj. pozorování květenství druhů rostlin pro pozdější korelaci s velikostí a kvalitou snůšky. Po zápise na portál jsou data přehledně uložena a uchovávána. Následně tyto záznamy o pozorování budou použity pro další výzkum o úživnosti včel v krajině, tzn. porovnání dat z úlové váhy s daty o fenologickém pozorování v okolí stanoviště včelstva.

Obrázek 1 Moje stanoviště se včelami



Název	Reg. číslo	Kat. území a obec	Včelstva
Kolin	788211	Kořenice, Chotouchov	2

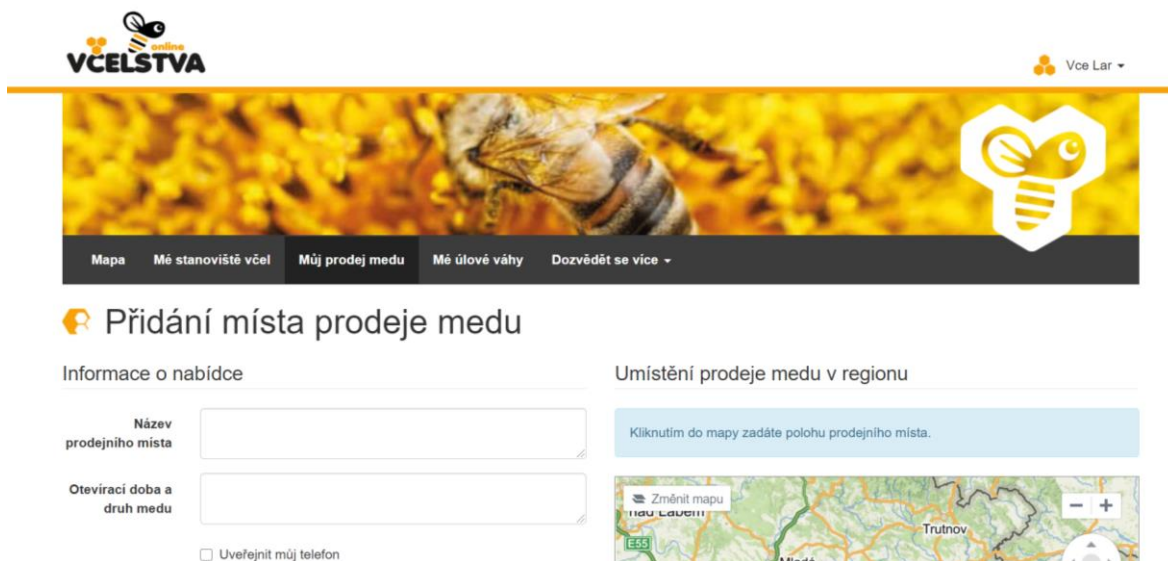
Stanoviště	Včelstvo	Matka	Barva
Kolin	10		Bílá
Kolin	8		

Zdroj: Včelstva online

4.3.2 Prodej medu

Na stránce Můj prodej medu si včelař může přidat podrobnosti o své prodejně kliknutím na tlačítko Přidat místo prodeje medu (obrázek 2). Zde vyplní informace o nabídce a adresu prodejny. Takto je možné vytvořit více prodejních míst.

Obrázek 2 Prodej medu



The screenshot shows the 'Včelstva online' website interface. At the top, there is a navigation bar with the logo 'VČELSTVA online' on the left and 'Vše Lar' on the right. Below the navigation bar is a banner image of a bee on a yellow background. Underneath the banner is a dark navigation menu with the following items: 'Mapa', 'Mé stanoviště včel', 'Můj prodej medu', 'Mé úlové váhy', and 'Dozvědět se více'. The main content area is titled 'Přidání místa prodeje medu'. It is divided into two columns. The left column is titled 'Informace o nabídce' and contains two text input fields: 'Název prodejního místa' and 'Otevírací doba a druh medu'. Below these fields is a checkbox labeled 'Uveřejnit můj telefon'. The right column is titled 'Umístění prodeje medu v regionu' and contains a light blue instruction box that says 'Kliknutím do mapy zadáte polohu prodejního místa.' Below this is a map showing a region with labels for 'Mladá', 'Trutnov', and 'E55'. The map has a 'Změnit mapu' button and zoom controls.

Zdroj: Včelstva online

4.3.3 Pozorování včelstva – úlový deník

Všichni včelaři usilují o efektivní chov včel. Včelařství je precizní činnost, která je ovlivňována mnohými faktory. Včelaři si faktory měří a zapisují je do úlového deníku (ProBee, 2017). Do úlového deníku jsou zapisovány pravidelné kontroly (obrázek 3). Je evidováno, kolik je obsedáno uliček, nebo rámečků včelami, dále jaké množství zásob je vytvářeno, jestli včely prodělávají onemocnění, případně jak je léčeno. Někteří včelaři se také soustředí na vzorce chování včel, na rojivost, mírnost, sezení atd. Záznamy jsou uloženy a jejich dlouhodobým sbíráním je možné vytvářet predikce.

Online úlový deník (obrázek 4) na portále Včelstva online byl vytvořen na základě diskusí s včelaři, se kterými je dlouhodobě spolupracováno. Včelaři mohou zapisovat do úlového deníku na stanovišti na papír, nebo použít jeden ze softwarů sloužících

Obrázek 3 Kontroly včelstva a úlu

Kontroly včelstva a úlu

Kontrola ze dne 02.05.2019

Počet nástavků: 4 Matka klade: *Ano* Obsedají uliček: -- || **Mírnost: 1** **Sezení: 1** **Rozvoj: 1** **Hygiena: 1**

Plod: -- Zásoby: -- Pyl: --

Medný výnos: -- Příště: *nevyplněno*

Kontrola ze dne 01.05.2019

Počet nástavků: 4 Matka klade: *Ano* Obsedají uliček: -- || **Mírnost: 1** **Sezení: 1** **Rozvoj: 1** **Hygiena: 1**

Plod: -- Zásoby: -- Pyl: -- **Poznámka:** *Odebral jsem 2 pl.plodu i s matkou + 2 pl. D1/2 se včelama do prezentačního úlu pro ČT2 - Dpbré ráno*

Medný výnos: -- Příště: *nevyplněno*

Kontrola ze dne 21.04.2019

Počet nástavků: 4 Matka klade: *Ano* Obsedají uliček: -- || **Mírnost: 1**

Plod: -- Zásoby: -- Pyl: -- **Poznámka:** *Přidán nást. D1/1 - 6 pl. + 4 st.r. jako 2 od spodu - Stav 1D1/2 + 1D1/1 + 2 x D1/2*

Medný výnos: -- Příště: *nevyplněno*

Zdroj: Včelstva online

Obrázek 4 Úlový deník

Mapa Mé stanoviště včel Můj prodej medu Mé úlové váhy Dozvědět se více ▾

Úlová karta: Včelstvo č. 10

Číslo matky	Původ matky	Barva matky	Typ úlu	Rámková míra	Stav včelstva
	<i>Není uvedeno</i>	Bílá	r	39x40	Zdravé

Aktuální umístění od	Název stanoviště	Reg. číslo	Katastrální území a obec	Předpokládaný konec stanoviště
26.12.2020	Kolin	788211	Kořenice, Chotouchov	<i>Nebyl stanoven</i>

Zdroj: Včelstva online

k tomuto účelu. Jestliže jsou včelaři vybaveni technologickými zařízeními, jako je např. tablet, zapisování do portálu je možné provést přímo na stanovišti. Včelaři však preferují zápis na papír a přepsat si údaje na portál až doma. Parametrů, které je možné

zapisovat na portál je mnoho a je možné přidat další. Portál nabízí funkci vedení záznamů o léčení včelstva, které je možné vytisknout si v PDF formátu. Tyto záznamy jsou povinni včelaři uchovávat pro případ kontroly Krajské veterinární správy.

Pro přesné měření parametrů týkajících se chovu včel, slouží úlová váha. Ta měří parametry, které není možné přesně kontrolovat vizuálně. Základní funkcí úlové váhy je měření hmotnosti, rozšiřitelné o měření teploty vně či uvnitř úlu, vlhkosti apod. Rozhodnout se, od kterého výrobce si včelař váhu pořídí, je pro každého individuální záležitostí. Většinou je rozhodnuto dle toho, co včelař potřebuje sledovat a zároveň kam bude úlová váha umístěna.

4.3.4 Zemědělec

Pro snadné upozornění včelařů na termín provádění postřiků, se mohou zaregistrovat zemědělci (obrázek 5). Posílání upozornění je anonymní a zároveň velmi užitečné pro včelaře, kteří mohou dopředu své včely ochránit.

Obrázek 5 Přidávání postřiků

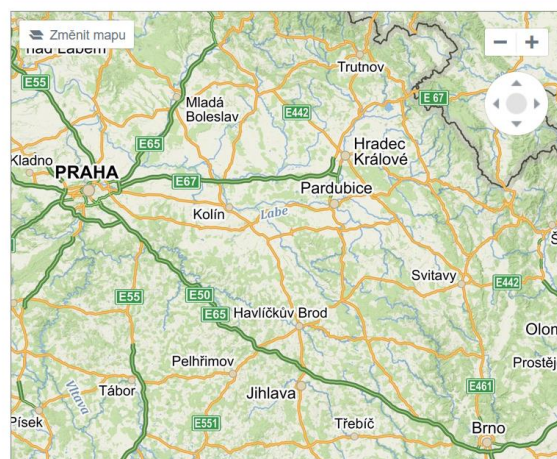
Nahlášení plánovaného postřiku včelařům

Údaje o plánovaném postřiku

Datum postřiku*	<input type="text"/>
Plodina*	<input type="text"/>
Účinná látka	<input type="text"/>
Vliv na včely:*	<input type="radio"/> Bez vlivu <input type="radio"/> Nebezpečný <input type="radio"/> Zvlášť nebezpečný
Další informace o postřiku:	<input type="checkbox"/> Kvetoucí plodina (včely ji mohou vyhledávat) <input type="checkbox"/> Postřik bude probíhat mimo letovou aktivitu včel <input type="checkbox"/> Uveřejnit můj telefon
Katastrální území	<input type="text"/>
Parcela	<input type="text"/>
Půdní blok	<input type="text"/>
<input type="button" value="Uložit plánovaný postřik"/>	

Místo postřiku na území ČR

Kliknutím do mapy zadáte místo postřiku. Údaje o katastrálním území a parcele se po kliknutí načtou automaticky.



Zdroj: Včelstva online

4.3.5 Včelařský spolek

Včelaři jsou často členy včelařských spolků, zejména pokud jsou začátečníci, protože včelařské spolky pořádají různá školení, pomáhají se zvládnutím legislativy a také posílají upozornění týkajících se nadcházejících změn v legislativě. Portál Včelstva online umožňuje zaregistrovat se v roli včelařského spolku (obrázek 6). V pozici administrátora je možné přihlásit jednu, nebo více osob. Administrátoři včelařského spolku k sobě následně přiřadí své členy. Těm zůstávají všechny funkce pro včelaře dále dostupné, nicméně někteří včelaři si svůj úlový deník a zapisování léčení nechtějí vést online. Pro tyto včelaře je výhodné být přiřazen ke své organizaci, jelikož administrátoři za ně mohou zadávat léčení, kontrolovat data z jejich úlové váhy a vyplnit formulář pro „Hradištko“. Také například upozornění na plánovaný postřik nemusí přijít přímo od zemědělce, ale může být posláno poslat včelařským spolkem, po komunikaci se zemědělci.

Obrázek 6 Včelařský spolek

Členové včelařského spolku

Počet členů: 3 Počet stanovišť: 6 Počet včelstev: 23

[Přidat nového člena](#) [Nastavení hromadných e-mailů](#)

Příjmení	Jméno	E-mail	Registrační číslo			
Jet	Regio	ha.kratochvilova+sppp@gmail.com	788996	Odebrat	Stanoviště	Hlášení pro Hradištko
Kkk	H	ha.kratochvilova+sp@gmail.com	788999	Editovat	Odeslat heslo	Stanoviště Hlášení pro Hradištko
Nna	A	ha.kratochvilova+spo@gmail.com	877889	Editovat	Odeslat heslo	Stanoviště Nemá vytvořené stanoviště.

[Tisk hlášení pro Hradištko všech členů](#) [Tisk seznamu členů spolku](#) [Tisk seznamu stanovišť členů spolku](#)

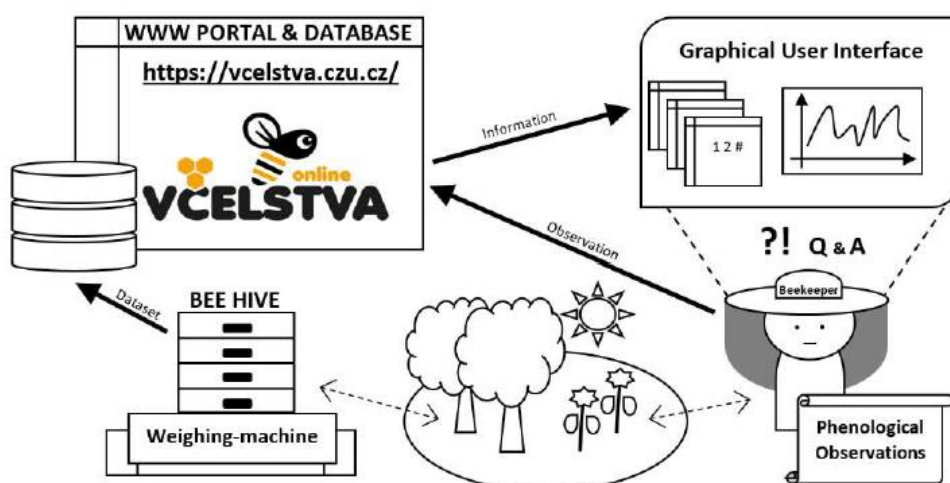
Zdroj: Včelstva online

4.3.6 Technologie

Na obrázku 7 je schematicky zobrazeno rozvržení portálu Včelstva online. Včelař (Beekeeper) se rozhodne začít používat software, aby mohl mít přehledně uložena všechna

data, která si zapisuje, na jednom místě. Zaregistruje se tedy na portálu Včelstva online skrze grafické uživatelské rozhraní (Graphical User Interface). Tímto rozhraním mu portál poskytuje informace (Information), může nahlížet na data ze svého úlu a zapisovat fenologická pozorování (Phenological Observation) ze svého stanoviště. Portál (www portal & database) je složen z webové stránky (<https://vcelstva.czu.cz/>) a dat uložených v databázi. Databáze uchovává veškerá data, která včelař zapisuje a která jsou mu na jeho vyžádání poté zobrazena. V databázi jsou zároveň uložena data, která posílá úlová váha (Weighing-machine) za použití IoT (Internet of Things) technologií. Data (Dataset) jsou posílána mobilní sítí, na portále dekódována algoritmem od výrobce a poté uložena pro zobrazení uživateli.

Obrázek 7 Portál Včelstva online



Zdroj: Bartoška, 2020

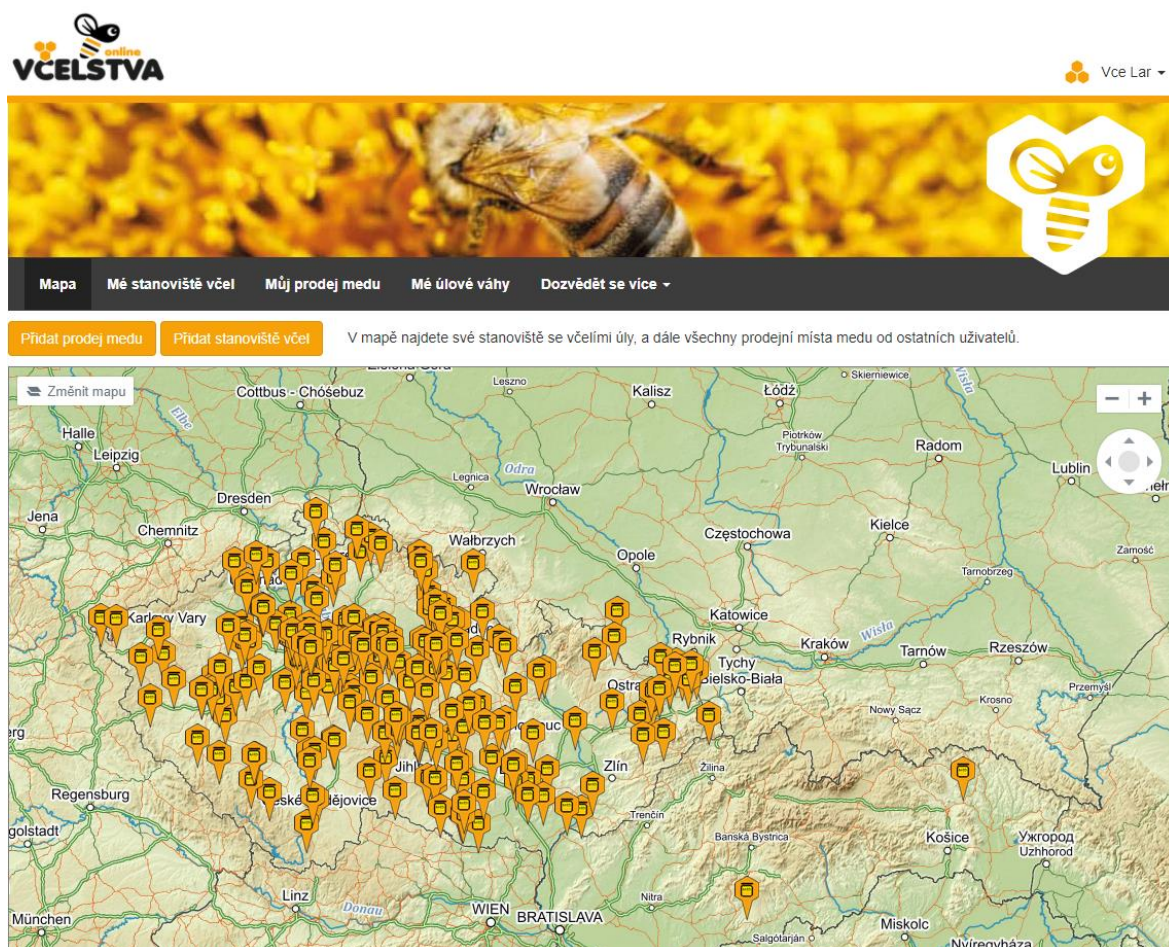
V databázi jsou uložena data o uživateli, jejich přihlašovací údaje, záznamy z úlového deníku a záznamy o fenologických pozorování. Data z úlové váhy jsou posílána přes IoT síť Sigfox a na portálu následně rozšifrována dle algoritmu výrobce. Každý výrobce vah používá odlišný způsob šifrování, proto je důležité při přidávání nového typu váhy do portálu správně data rozšifrovat.

Data uložená v databázi se zobrazují skrze grafické rozhraní na webové stránce vcelstva.czu.cz. Webová stránka je vytvořena ze použití programovacích jazyků PHP, JavaScript, HTML a CSS. Při psaní kódu webové stránky byl použit framework Nette Latte.

4.3.7 Prodej medu na portálu Včelstva online

Často využívanou funkcí portálu je zveřejnění prodejny medu (obrázek 8). Někteří včelaři nemají zájem o funkce jako je vedení úlového deníku, nebo záznamy o léčení, ale registrují se na portál proto, aby mohli zviditelnit své místo prodeje medu. Stejně tak velká část včelařů nechce věnovat příliš času ani financí marketingu a propagaci svých výrobků. Proto uvítají možnost zdarma uveřejnit svou prodejnu. Při přidání prodejny medu včelař vyplní krátký formulář, který se skládá z názvu a adresy prodejního místa, otevírací doby a druhu prodávaného medu, případně je možné zveřejnit včelařův telefon. Většina včelařů se zaregistrovanými prodejny medu volí možnost zveřejnění telefonu, jelikož nemají pevně stanovenou otevírací dobu. Často ani nemají oddělenou prodejnu, kde by seděli celý den, ale v případě zájmu zákazníků se s nimi domluví na čase a med prodávají ze dvora, což je právě faktorem umožňujícím flexibilní otevírací dobu.

Obrázek 8 Prodej medu



Zdroj: Včelstva online

4.4 Mobilní aplikace

Portál Včelstva online se stále rozšiřuje. Na přání včelařů, kteří prodávají med, bude vytvořena mobilní aplikace na podporu prodeje medu ze dvora. Včelaři jsou velmi schopní, co se týká péče o včely, spousta jich ale není tak zdatná v marketingu. Někdy se stává, že mívají velký výnos medu a s ním spojené starosti co s přebytky. Část přebytků prodají svým stálým zákazníkům, ale nejsou schopni o sobě dát dostatečně vědět, aby přebytky prodaly všechny. Tyto a řada dalších obtížností spojených s prodejem medu ze dvora by měla být v budoucnu vyřešena mobilní aplikací.

Mobilní aplikace zobrazuje prodejní místa medu. Tato místa jsou zobrazována v mapě na portálu Včelstva online. Tento portál, ale používají zejména včelaři, veřejnost jen zřídka. Portál je totiž určen primárně pro včelaře, a proto je náročnější zdůrazňovat pro širokou veřejnost pouze seznam prodejen medu. Aplikace bude určena pouze pro prodej medu, a tudíž by měla být přehledná pro cílové uživatele, tedy konzumenty medu, kteří chtějí domácí kvalitní med přímo od včelaře.

Včelaři, kteří chtějí mít zobrazenou svoji prodejnu medu v aplikaci, se musí zaregistrovat na portále. Po přihlášení si na stránce Můj prodej medu mohou přidat místo prodeje medu. Zde se vyplňují informace o nabídce a adresu.

4.5 Prodej medu ze dvora

Z možností, které včelaři mají, se jeví nejvhodnější využít vlastní prostory pro prodej medu ze dvora. Tento způsob má řadu výhod, zejména neplacení nájmu za aditivní prostor a volba prostor dle potřeb včelaře. Při tomto způsobu distribuce se neplatí za zprostředkování prodeje dalšímu subjektu (Šmídková, 2018, s.76).

4.5.1 Propagace

Nejlevnější varianta propagace je přímo v místě prodeje, například na vlastním pozemku. Vývěsní štít, nebo cedule informující kolemjdoucí, nebo projíždějící, potenciální zákazníky o možnosti zakoupení medu (Šmídková, 2018, s.76). Informovat blízké okolí je výhodné, jelikož je pro něj prodejna dobře dostupná. „O tuto stacionární reklamu by měl včelař pečovat a pravidelně ji obnovovat, často se setkáváme s cedulemi poničenými povětrnostními vlivy, nečitelnými, nebo neaktuálními kontaktními údaji.“ (Šmídková,

2018, s. 76). Výhodou prodeje medu ze dvora je také zvýšení zájmu zákazníků o včelařství samotné. Protože v místě prodeje je možné zájemcům předvést nástroje výroby medu, dokonce pokud je v blízkosti stanoviště včel, je možné ho zákazníkům ukázat a umocnit tak zážitek z návštěvy.

4.6 Fungování mobilní aplikace

Mobilní aplikace bude zobrazovat uživatelům prodejny medu a informace o nich. Po otevření aplikace se uživateli zobrazí hlavní stránka, na které jsou prodejny medu zobrazeny v mapě. Uživatel může prodejny medu vyhledávat. Při rozkliknutí prodejny se uživateli zobrazí informace o prodejně, které včelař zadal na portálu Včelstva online. Mobilní aplikace pracuje s Google Maps, což jsou mapy, které nabízejí freeware pro použití. Aplikace funguje pouze pro zobrazování informací. Vkládání je možné pouze přes portál, jelikož ten nabízí pro včelaře mnoho funkcí a jsou zvyklí s ním takto pracovat.

4.6.1 Stahování dat

Pro mobilní aplikaci jsou stahována data z databáze portálu Včelstva online. Stahují se data o prodejnách medu, o majitelích prodejen medu a jejich souřadnice. Aplikace potřebuje souřadnice pro zobrazení prodejen v mapě. Na portále Včelstva online jsou souřadnice uloženy ve formátu JTSK, jelikož na portále se používají mapy společnosti Seznam.cz.

4.7 Vytvoření API pro mobilní aplikaci

Pro vznikající mobilní aplikaci bylo vytvořeno API pro stahování dat o prodejnách medu. Pro API byl vytvořen nový presenter pojmenovaný ApiPresenter. Nejprve se začalo implementací public vlastností, které jsou potřebné v celém presenteru.

V textu práce jsou ukázány úseky kódu, které jsou nezbytné pro přesné popsání fungování API. Bohužel není možné demonstrovat fungování API v kontextu zbylého kódu portálu Včelstva online, protože obsah kódu portálu není majetkem autorky této práce.

4.7.1 Potřeby mobilní aplikace

Mobilní aplikace, která je vytvářena, je určena pro podporu prodeje medu ze dvora. Tato aplikace bude zobrazovat data z databáze portálu Včelstva online (tabulka 2). Budou to informace o prodejnách medu, které jsou zaregistrované na portále. Tyto prodejny budou zobrazovány v mapě. Aplikace tedy z databáze portálu Včelstva online bude potřebovat souřadnice prodejen medu a informace o prodejnách.

Tabulka 2 Prodejna medu v databázi

Sloupec	Typ
id	int(10) unsigned <i>Auto Increment</i>
datum_vytvoreni	datetime <i>NULL</i>
nazev	varchar(500) <i>NULL</i>
vcelar_id	int(10) unsigned
informace	varchar(500)
uvrejniti_telefon	tinyint(1) unsigned
x	double
y	double
lat	decimal(12,9) <i>NULL</i>
lng	decimal(12,9) <i>NULL</i>
adresa_id	int(10) unsigned
smazana	tinyint(1) unsigned [0]
smazana_datum	datetime <i>NULL</i>

Zdroj: databáze Včelstva online

Data prodejen veřejná, tak žádné přihlášení není potřeba a data se tedy získávají pouze přes API, které vygeneruje JSON a v aplikaci se s ním dále pracuje. JSON prodejny vypadá následovně:

```
{
    "id":53,
    "lat":50.187924999999999989768184605054557323455810546875,
    "lng":14.931817999999999981309883878566324710845947265625,
    "nazev":"Rozkoš 26",
    "informace":"Kdykoliv",
    "ulice":"Rozkoš 26",
    "obec":"Kostomlaty nad Labem",
    "psc":"28921",
    "telefon":"+420777776594",
    "web":null
}
```

Data z databáze se stahují v nejnovější verzi pokaždé při startu mobilní aplikace, během toho se stáhnou informace o všech prodejnách. Souřadnice prodejen jsou uloženy v lokální databázi.

4.7.2 Zpracování požadavku na API

ApiPresenter je výchozí presenter pro implementaci vlastního API. Tento presenter zpracovává volání pomocí HTTP metod GET, POST, PUT a DELETE. Pokud požadavek obsahuje tělo ve formátu JSON a příslušná metoda má parametr \$json, dostane metoda data již dekodovaná (objekty jako stdClass, pole jako array). Metoda může vracet data ve formátu JSON metodou { @link sendJson }.

Akci presenteru zprostředkovává metoda run se vstupujícím parametrem \$request (žádost), která volá jednotlivé metody, v pořadí scanMethods, initParameters, startup, callMethod. V případě chyby v žádosti zachytí výjimku skrze AbortException, což je funkce Nette. Metoda run vypadá následovně:

```
/**
 * Executes the presenter action.
 *
 * @param Request $request
 * @return IResponse
 */
public function run(Request $request)
{
    try {
        $this->scanMethods();
        $this->request = $request;
        $this->initParameters();
        $this->startup();
        $this->callMethod($this->action, $this->method, $this->params);
        return NULL;
    } catch (AbortException $ex) {
        return $this->response;
    }
}
```

Metoda scanMethods() z hlavičky HTTP požadavku zjišťuje o kterou metodu se jedná. Použity jsou metoda Nette getMethod():string, ta vrací metodu, kterou byl udělán požadavek. A metoda getName(), která vrací originální název souboru, jak byl odeslán. Následně tato metoda předává parametry funkci addMethod, která se zavolá. Popsané metody vypadají takto:

```

/**
 * Scans all available action methods.
 */
protected function scanMethods()
{
    $src = new ReflectionClass($this);
    foreach ($src->getMethods() as $rm) {
        if (!$rm->isPublic() || $rm->isAbstract() || $rm->isStatic())
        {
            continue;
        }
        $matches = Strings::match($rm->getName(),
'#^do(get|post|put|delete) ([a-z0-9][a-z0-9_]*)\z#i');
        if (!$matches) {
            continue;
        }
        $this->addMethod(Strings::lower($matches[1]),
Strings::lower($matches[2]), $rm);
    }
}

/**
 * Adds action method.
 *
 * @param string $method
 * @param string $action
 * @param ReflectionMethod $rm
 */
protected function addMethod($method, $action, ReflectionMethod $rm)
{
    if (!isset($this->methods[$action])) {
        $this->methods[$action] = [];
    }
    $this->methods[$action][$method] = $rm;
}

```

Metoda `initParameters()` zařizuje inicializaci parametrů a akce. Na začátku se načtou parametry pomocí Nette metody `getParameters()`, která všechny proměnné poskytnuté presenteru načte a přiřadí z parametrů požadavku. Proměnné jsou poskytnuty presenteru většinou přes adresu zdroje (URL z *angl.* Uniform Resource Locator). Jestliže jsou parametry napsány správně, funkce vrátí parametry, akci a metodu. Metoda se načte zavoláním Nette metody `getMethod()`. Metoda `initParameters()` je napsána takto:

```

/**
 * Initializes presenter parameters and action.
 */
protected function initParameters()
{
    $params = $this->request->getParameters();
    $action = isset($params[self::ACTION_KEY]) ? $params[self::
ACTION_KEY] : self::DEFAULT_ACTION;
}

```

```

        if (!is_string($action) || !Strings::match($action, '#^[a-zA-Z0-9][a-zA-Z0-9_]*\z#')) {
            $this->error('Invalid action name.');
```

```

        }
        if (Strings::match($this->httpRequest->getHeader('Content-Type'), '#^application/json\b#')) {
            try {
                $params['json'] = Json::decode($this->httpRequest-
>getRawBody(), Json::FORCE_ARRAY);
            } catch (JsonException $ex) {
                $this->error('Malformed JSON content (' . $ex-
>getMessage() . ')');
```

```

            }
        } else {
            Debugger::log(sprintf("Invalid content type: %s", $this-
>httpRequest->getHeader('Content-Type')), ILogger::WARNING);
            Debugger::log(sprintf("Content is: %s", $this->httpRequest-
>getRawBody()), ILogger::WARNING);
        }
        $this->params = $params;
        $this->action = Strings::lower($action);
        $this->method = Strings::lower($this->httpRequest->getMethod());
    }
}

```

Následující volání probíhá z metody run a je volána metoda callMethod, která kontroluje, že není identifikován žádný z problémů:

- může se jednat o neznámou akci, v tomto případě IHttpReponse::S404_NOT_FOUND, tzn. metoda nebyla nalezena, chyba na straně serveru.
- Jestliže metoda v hlavičce požadavku není rozpoznána, vypíše se chybová hláška IHttpReponse::S405_METHOD_NOT_ALLOWED.
- Zároveň metoda musí být veřejná, nesmí být abstraktní ani statická, jinak se vypíše IHttpReponse::S500_INTERNAL_SERVER_ERROR.

Kód metody callMethod je formulován takto:

```

/**
 * Calls action method if exists.
 *
 * @param string
 * @param string
 * @param array
 */
protected function callMethod($action, $method, array $params)
{
    if (!isset($this->methods[$action])) {
        $this->error('Unknown action.', IHttpReponse::S404_NOT
_FOUND);
    }
    if (!isset($this->methods[$action][$method])) {
        $this->error('Method not allowed.',
IHttpReponse::S405_METHOD_NOT_ALLOWED);
    }
}

```

```

    }
    $rm = $this->methods[$action][$method];
    if (!$rm->isPublic() || $rm->isAbstract() || $rm->isStatic()) {
        $this->error('Invalid action handler.',
IHttpResponse::S500_INTERNAL_SERVER_ERROR);
    }
    $rm->invokeArgs($this, ComponentReflection::combineArgs($rm,
$params));
}

```

Následně se parametry předají funkci Nette `invokeArgs($callable, pole $args[])`, která zavolá `$callable` s argumenty `$args` a předá jim vrácenou hodnotu, když dojde k chybě, vrátí Nette výjimku s varováním.

4.7.3 Odpověď aplikaci

Po zpracování požadavku se odešle JSON metodou `sendJson($data)` odpověď ze statické třídy Nette (obrázek 13). Po poslání odpovědi se ukončí presenter funkcí `terminate()`. Provedení metod vypadá následovně:

```

/**
 * Sends JSON data to the output.
 *
 * @param mixed $data
 * @throws AbortException
 */
public function sendJson($data)
{
    $this->sendResponse(new JsonResponse($data));
}

/**
 * Sends response and terminates presenter.
 *
 * @param IResponse $response
 * @throws AbortException
 */
public function sendResponse(IResponse $response)
{
    $this->response = $response;
    $this->terminate();
}

/**
 * Correctly terminates presenter.
 *
 * @throws AbortException
 */
public function terminate()
{
    throw new AbortException();
}

```

4.7.4 Vrácení požadavku

Mobile presenter vrátí všechny prodejny medu, které se nacházejí v blízkosti zadaných souřadnic. V tomto presenteru je pouze metoda `doGetProdejnyMedu` s parametry `lat` (zeměpisná šířka), `lng` (zeměpisná délka) a `distance` (vzdálenost v metrech). Metoda nejprve zavolá statickou třídu `GeoUtils2::toJtsk` s parametry `lat` a `lng`. Tato třída následně převede souřadnice z mobilní aplikace z formátu WGS84 do formátu JTSK, který je používán portálem Včelstva online, a ve kterém s nimi presenter dále pracuje.

Ve formátu JTSK, který jsou používány na portálu Včelstva online, jsou uložena data v databázi. Formát JSTK byl zvolen kvůli použití `mapy.cz` na portálu. Nicméně mobilní aplikace bude používat mapy společnosti Google. Tyto mapy Google jsou používány po celém světě, a proto nemohou používat pravoúhlý systém souřadnic jako formát JTSK, ale používají formát WGS84.

Následně se spočítají souřadnice plochy, ve které se budou hledat prodejny. Ty se načtou funkcí `findByLocation`. Výsledné proměnné v seznamu se převedou zpátky do formátu WGS84 přes třídu `GeoUtils2::toWgs84` a výsledek se uloží do pole `$vysledek[]` a pomocí funkce `sendJson($vysledek)` se odešle mobilní aplikaci. Aplikaci se tedy pošlou data v podobě, která byla popsána v kapitole 4.7.1. Celý kód pro mobile presenter vypadá takto:

```
class MobilePresenter extends ApiPresenter
{
    /**
     * @var Model
     * @inject
     */
    public $model;

    /**
     * Vrátí všechny prodejny medu (včetně jejich nabídky), které se
     * nacházejí
     * v blízkosti zadaných souřadnic.
     *
     * @param float $lat Zeměpisná šířka.
     * @param float $lng Zeměpisná délka.
     * @param float $distance Vzdálenost v metrech.
     */
    public function doGetProdejnyMedu(float $lat, float $lng, float
    $distance)
    {
        list($x, $y) = GeoUtils2::toJtsk($lat, $lng);
```

```

    $minX = $x - $distance;
    $maxX = $x + $distance;
    $minY = $y - $distance;
    $maxY = $y + $distance;

    $prodejny = $this->model->prodejnyMedu->findByLocation($minX,
    $minY, $maxX, $maxY);

    $vysledek = [];

    foreach ($prodejny as $p) {
        if (GeoUtils2::distanceJtsk($p->x, $p->y, $x, $y) >
    $distance) {
            continue;
        }

        list ($lat, $lng) = GeoUtils2::toWgs84($p->x, $p->y);
        $vysledek[] = [
            'id' => $p->id,
            'lat' => $lat,
            'lng' => $lng,
            'nazev' => $p->nazev,
            'informace' => $p->informace,
            'ulice' => $p->adresa->ulice,
            'obec' => $p->adresa->castObce,
            'psc' => $p->adresa->psc,
            'telefon' => $p->uverejnitTelefon ? $p->vcelar->admin-
    >telefon : null,
            'web' => $p->vcelar->web,
        ];
    }

    $this->sendJson($vysledek);
}
}

```


5 Závěr

Včelařství má v naší zemi dlouhou tradici a stejně tak přímý prodej medu koncovým zákazníkům. Řada včelařů má stálý okruh zákazníků, který se obměňuje pouze zřídka a odebírá povětšinou konzistentní množství medu. Proto v případě větší snůšky medu nebo při rozšíření počtu včelstev může docházet k problémům s hledáním odbytu pro přebytky. K řešení situace by měla pomoci tato bakalářská práce, v rámci níž vzniklo aplikační rozhraní (API) pro komunikaci mezi portálem Včelstva online a mobilní aplikací. Širší projekt na podporu včelařství, kterého je tento portál součástí, si klade za cíl vytvořit platformu pro včelaře, která by jim usnadnila nejenom péči o včely, ale i následný marketing prodeje medu ze dvora.

V teoretické části byla důkladně rozebrána samotná komunikační technologie API, její vývoj a možnosti jejího využití. Dále se tato část práce důkladně věnuje programovacím jazykům a technologiím nezbytným pro vytvoření API, jako jsou: PHP, MySQL a Nette framework. V praktické části pak bylo rozebráno fungování portálu Včelstva online, pro přiblížení, jaké data jsou vkládána včelaři do databáze a co bude přes API přenášeno. V druhé části je pak popsáno fungování vzniklého API. Tento popis by měl v budoucnu posloužit vývojáři, který bude vytvářet mobilní aplikaci s propojenou k portálu přes toto API. Součástí API je i převod pravoúhlého trigonometrického formátu JTSK do formátu WGS84 využívaného v Google Maps, které budou pro mobilní aplikaci použity.

V rámci této práce bylo vytvořeno API pro mobilní aplikaci portálu Včelstva online. Vytvořené API by se v budoucnu mělo být využito jako komunikační rozhraní mezi databází portálu a mobilní aplikací zaměřenou podporu marketingu prodeje medu ze dvora.

6 Seznam použitých zdrojů

ALAN, Forbes, 2015. The Joy of PHP: A Beginner's Guide to Programming Interactive Web Applications with PHP and mySQL. 3. USA: CreateSpace Publishing. ISBN 9781494267353.

BARTOŠKA, Jan, Tomáš ŠUBRT, Jan RYDVAL, Jan KAZDA a Martina STEJSKALOVÁ, 2020. System Dynamic Conceptual Model for Landscape Fertility of Bees. 38th International Conference on Mathematical Methods in Economics. 6.

BIEHL, Matthias, 2015. API Architecture: The Big Picture for Building APIs. 2. vydání. API-University. ISBN 978-1508676645.

DAVIS, Michele E. a Jon A. PHILLIPS, 2007. Learning PHP and MySQL. 2. USA: O'Reilly Media. ISBN 978-0-596-51401-3.

DYER, Russell J.T., 2015. Learning MySQL and MariaDB. USA: O'Reilly Media. ISBN 978-1-44936-290-4.

Framework [online], 2013. In: . Dostupné z: <https://techterms.com/definition/framework>

GOURLEY, David a Brian TOTTY, 2002. HTTP: The Definitive Guide. USA: O'Reilly Media. ISBN 978-1-56592-509-0.

JACOBSON, Daniel, Greg BRAIL a Dan WOODS, 2012. APIs: A Strategy Guide. USA: O'Reilly Media. ISBN 978-1-449-30892-6.

JIN, Brenda, Saurabh SAHNI a Amir SHEVAT, 2018. Designing Web APIs. USA: O'Reilly Media. ISBN 978-1-492-02692-1.

JURKINA, Maria Ivanovna a Miloš PICK, 2006. Numerické výpočty ve světovém geodetickém referenčním systému 1984 (WGS84). Dobruška: Ministerstvo obrany ČR, Geografická služba AČR.

LAURET, Arnaud, 2019. The Design of Web APIs. USA: Manning Publications Co. ISBN 978-1-617-29510-2.

Maps JavaScript API: Data Layer [online]. In: . Dostupné z: <https://developers.google.com/maps/documentation/javascript/datalayer>

MariaDB [online], 2021. Dostupné také z: <https://mariadb.com/>

MASSÉ, Mark. REST API design rulebook. Sebastopol, CA: O'Reilly, 2012. ISBN 9781449310509.

MILLS, Chris a Vidhi BAGADIA. Server-side web frameworks [online]. 2017 Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Web_frameworks

MITCHELL, Lorna Jane, 2016. PHP Web Services. Druhé vydání. USA: O'Reilly Media. ISBN 978-1-491-93309-1.

NIXON, Robin, 2014. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. 4. USA: O'Reilly Media. ISBN 978-1-491-91866-1.

PREIBISCH, Sasha, 2018. API DEVELOPMENT: A Practical Guide for Business Implementation Success. Richmond, BC, Kanada: Apress Media. ISBN 978-1-4842-4140-0.

Směrnice Evropského parlamentu a Rady (EU) 2015/2366, 2015. In: . Lucemburk: Úřad pro publikace Evropské unie, 337/35.

ŠIMBERA, Jan, 2018. Souřadnicové systémy. Praha. Výukový materiál. Přírodovědecké fakultě Univerzity Karlovy v Praze.

ŠMÍDKOVÁ, Jitka, 2019. Využití ICT nástrojů pro chov včel a prodej medu v regionu. Praha. Diplomová práce. Česká zemědělská univerzita.

ŠULC, Milan a David GRUDL, 2021. Dokumentace Nette 3.1. Nette [online]. Nette Foundation. Dostupné z: <https://doc.nette.org/cs/3.1/>

TATROE, Kevin a Peter MACINTYRE, 2020. Programming PHP. 4. USA: O'Reilly Media. ISBN 978-1-492-05413-9.

Using the Twitter API [online]. In: . Dostupné z: <https://developer.twitter.com/en/docs/twitter-api/getting-started/guide>

Včelstva online [online]. Dostupné také z: <https://vcelstva.czu.cz/>

WEINBERG, Paul N., James R. GROFF a Andrew J. OPPEL. SQL, the complete reference. 3rd ed. New York: McGraw-Hill, c2010. ISBN 978-0071592550.