



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**SMART CITY BRNO - PRŮZKUMY**

SMART CITY BRNO - RESEARCHES

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**PAVEL WITASSEK**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. SVETOZÁR NOSKO**

BRNO 2019

## Zadání bakalářské práce



21972

Student: **Witassek Pavel**  
Program: Informační technologie  
Název: **Smart City Brno - průzkumy**  
**Smart City Brno - Researches**  
Kategorie: Uživatelská rozhraní  
Zadání:

1. Prostudujte možnosti a literaturu na téma tvorby průzkumů a sběru názoru obyvatel. Zaměřte se například na tzv. pocitové mapy a/nebo jiné postupy.
2. Vyberte vhodný postup pro sběr názoru tak, aby bylo možné spolehlivě identifikovat místa, ke kterým mají obyvatelé pozitivní anebo negativní vztah. Systém by měl umožnit také sběr zpětné vazby (názory, nápady, atd.) a vytváření průzkumů.
3. Navrhněte postup implementace a platformu takového systému a vyhodnoťte dosažitelné vlastnosti řešení.
4. Implementujte experimentálně systém a demonstруйте vlastnosti.
5. Vyhodnoťte dosažené výsledky a možnosti dalšího pokračování práce.

### Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Nosko Svetozár, Ing.**  
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.  
Datum zadání: 1. listopadu 2018  
Datum odevzdání: 15. května 2019  
Datum schválení: 6. listopadu 2018

## Abstrakt

Cílem této práce je vytvořit řešení pro město Brno, které městu umožní sběr podnětů a informací o jednotlivých lokalitách od občanů města. Výsledná implementace se skládá ze dvou částí, první část je server s API rozhraním a druhá část je webová aplikace s mapou. Pro zaznamenávání názorů obyvatel je využita pocitová mapa, do které respondenti zadávají body. Město Brno tak získá zpětnou vazbu od obyvatel a navíc u jednotlivých bodů mohou respondenti také uvést textový komentář. Výsledkem řešení je možnost si definovat vlastní průzkumy a město díky tomu nemusí využívat placená řešení, která se navíc musela přizpůsobovat dle potřeb města.

## Abstract

The goal of this thesis is to create the solution for the City of Brno, which will enable the collection of ideas and information about individual areas from the inhabitants of the city. The provided implementation is composed from two parts, the first one is the server with the API interface and the second one is the web application with the map. The mental map is used for the records of the opinions of the inhabitants of the City of Brno, where the responders mark the points. The City of Brno gets the feedback from its citizens and more important, the city gets the possibility to define its own survey and the city does not have to use costly solutions, which would anyway have to be adopted to the city needs.

## Klíčová slova

GIS, průzkum, PHP, Symfony, Angular, sběr názorů, ArcGIS, participativní mapování, REST, Zurb Foundation, mapa, Brno, Typescript

## Keywords

GIS, research, PHP, Symfony, Angular, opinion mapping, ArcGIS, participatory mapping, REST, Zurb Foundation, map, Brno, Typescript

## Citace

WITASSEK, Pavel. *Smart City Brno - průzkumy*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Svetožár Noko

# Smart City Brno - průzkumy

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Svetozára Noska. Další informace mi poskytl pan Mgr. Jan Zvara, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Pavel Witassek

14. května 2019

## Poděkování

Rád bych tímto poděkoval vedoucímu práce za pomoc a rady při tvorbě této práce. Dále bych rád poděkoval panu Mgr. Janovi Zvarovi, Ph.D. za odborné rady během celé práce a pomoc při testování.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Problematika tvorby průzkumů a sběru názorů obyvatel</b>	<b>4</b>
2.1	Mentální mapa . . . . .	4
2.2	Mapa vnímání místa . . . . .	4
2.3	Pocitová mapa . . . . .	5
2.4	Mapová aplikace . . . . .	5
2.5	Google Maps . . . . .	6
2.6	Mapy.cz . . . . .	6
2.7	OpenstreetMap . . . . .	7
2.8	Český úřad zeměměřický a katastrální - ČÚZK . . . . .	7
2.9	ArcGIS Online . . . . .	7
<b>3</b>	<b>Existující řešení</b>	<b>8</b>
3.1	Pocitové Mapy . . . . .	8
3.2	Maptionnaire . . . . .	8
3.3	Social Pinpoint . . . . .	9
<b>4</b>	<b>Existující nástroje a standardy</b>	<b>10</b>
4.1	GeoJSON - RFC 7946 . . . . .	10
4.2	WMS - Web Map Service . . . . .	11
4.3	WMTS - Web Map Tile Service . . . . .	11
4.4	Knihovna Leaflet . . . . .	12
4.5	ArcGIS JS API . . . . .	12
<b>5</b>	<b>Analýza požadavků</b>	<b>13</b>
5.1	Popis problému . . . . .	13
5.2	Požadavky aplikace . . . . .	13
<b>6</b>	<b>Návrh systému</b>	<b>16</b>
6.1	Entity Relationship Diagram – ERD . . . . .	16
6.2	Webové technologie . . . . .	17
<b>7</b>	<b>Implementace</b>	<b>19</b>
7.1	Serverová část aplikace . . . . .	19
7.2	Schéma databáze . . . . .	23
7.3	Klientská část aplikace . . . . .	27
<b>8</b>	<b>Testování</b>	<b>31</b>

8.1	Testování na úřednících magistrátu . . . . .	31
8.2	Testování na respondentech . . . . .	31
8.3	Testování exportu velkého množství odpovědí . . . . .	32
8.4	Možnosti dalšího rozšíření . . . . .	32
<b>9</b>	<b>Závěr</b>	<b>33</b>
	<b>Literatura</b>	<b>34</b>
<b>A</b>	<b>Schéma tabulek v databázi</b>	<b>36</b>

# Kapitola 1

## Úvod

Město Brno potřebuje sbírat názory a podněty od občanů jednotlivých městských částí v prostoru, tyto informace využívá pro potřeby změn veřejného prostranství ve městě. K tomuto sběru využívá pocitové mapy, současné řešení ovšem nevlastní a musí tak platit za licence. Dále současné řešení prodražují drobná přizpůsobení, volba otázek, definice oblasti apod.

Participativní plánování[17] veřejného prostranství poskytuje obyvatelům možnost ovlivnit danou lokalitu tak, aby co nejvíce vyhovovala jejich potřebám. Pocitové mapy umožňují sběr informací o jednotlivých lokalitách od občanů prostřednictvím online mapy, jedná se tedy o nástroj participativního plánování veřejného prostoru.

Poslední pocitovou mapu město Brno uspořádalo v roce 2018[10], poprvé ji město uspořádalo o dva roky dříve[6], tedy v roce 2016. Při posledním mapování město zjišťovalo, kde se obyvatelům líbí a chtějí dané místo zachovat v současné podobě, kde se občanům nelíbí a chtějí změnu, kde obyvatelé vnímají centrum dané čtvrti, kam se má městská část rozvíjet a kde obyvatelům něco chybí. Oproti tomu v roce 2016 město zjišťovalo, kde lidé tráví volný čas, kde jsou na město Brno hrdí a kde chtějí bydlet. Stejně jako při posledním mapování se město zajímalo, kde se obyvatelům líbí či nelíbí.

Cílem této práce je vytvořit řešení pro město Brno, které městu umožní sběr podnětů a informací o jednotlivých lokalitách ve městě od občanů, ať už se jedná o jeden konkrétní park, jedno prostranství či městskou část. Město Brno si bude moci vytvářet vlastní průzkumy, tedy vymezit lokalitu sběru názorů, volit otázky a výsledky průzkumu exportovat ve formátu vhodném pro další vizuální zpracování.

V následující kapitole číslo 2 je popsána problematika sběru názorů obyvatel, participativního plánování a pocitových map. V kapitolách číslo 3 a 4 jsou zmíněna existující řešení a nástroje, které se používají ke sbírání podnětů a názorů od obyvatel formou pocitových map v České republice i v zahraničí. Analýze požadavků města Brna na sběr názorů pomocí pocitových map se věnuje kapitola č. 5 a na základě těchto požadavků byl vytvořen návrh tohoto systému. Tento návrh je popsán v kapitole č. 6. V kapitole 7 je popsána samotná implementace tohoto webového portálu a v kapitole 8 jsou popsány informace o způsobu testování implementovaného systému a možnostech jeho dalšího rozšíření.

## Kapitola 2

# Problematika tvorby průzkumů a sběru názorů obyvatel

Participativní plánování veřejného prostoru je možnost, jak zapojit obyvatele do procesu rozhodování o výsledné podobě dané lokality. Jedná se o nástroj pro návrh veřejného prostoru, aby splňovalo očekávání a požadavky obyvatel, kteří tuto lokalitu navštěvují nebo v ní bydlí. Poskytuje jim možnost aktivně se podílet na výsledné podobě už ve fázi plánování daného prostoru. Aby mohl být výsledný návrh kvalitní, tedy aby splnil požadavky obyvatel, které ovlivní, je potřeba znát jejich názory a podněty.

Cílem sbírání názorů a podnětů je popsat řešené území z různých stran pohledů, jak je vnímají jednotliví obyvatelé. Různé skupiny obyvatel samozřejmě vnímají dané území jinak, proto je potřeba během průzkumu názorů zaznamenávat i od koho daný názor vzešel, jestli se jedná o mladistvé, starousedlíky, maminky s kočárky apod. Součástí průzkumu názorů v prostoru jsou tedy kromě otázek, kde se mi líbí nebo nelíbí, kde rád trávím volný čas i otázky na věk, pohlaví apod.

Ke sbírání podnětů či názorů od uživatelů se dá použít několik metod, které jsou rozepsány níže. Nejpoužívanější je v dnešní době pocitová mapa ve formě webové aplikace, kdy jsou uživatelům zobrazovány otázky a očekává se od nich jejich zodpovězení formou zanesení bodů do mapy. Nemusí se ovšem jednat jenom o body, ale respondenti mohou zadávat i různé tvary.

### 2.1 Mentální mapa

Kognitivní mapu[23] nosí každý z nás v hlavě, jde o mapu okolního světa, jak jej vnímáme. Jedná se tedy o subjektivní pohled na svět, tento pohled je ovšem individuální. Překreslením kognitivní mapy z mysli na papír dostáváme mentální mapu. Překreslení je ovšem na čistý nepopsaný papír, takže se tento postup dá popsat jako volná kresba z paměti. Výsledná mapa je ovšem velmi obtížně transformovatelná do souřadnicového systému. Výhodami jsou její technická a finanční nenáročnost, v případě že mapu nebudeme převádět do souřadnicového systému.

### 2.2 Mapa vnímání místa

Mapa vnímání místa[5] je metoda, při které jsou nejčastěji do předpřipravené mapy zaznamenávána místa dle předem zvolených kritérií, jako jsou oblíbené místo, místo pro děti,



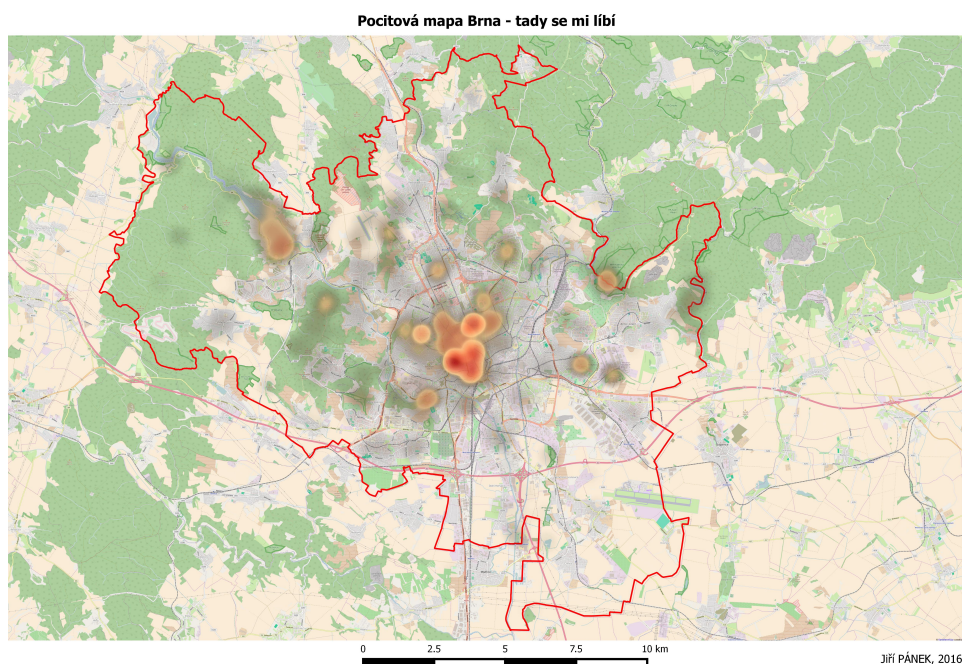
tajemné místo apod. Tato místa jsou zaznamenávána ve skupinkách osob a každá skupina pak ke každému označenému místu zapíše slovní komentář, proč místo vybrala.

## 2.3 Pocitová mapa

Tato metoda je založena na předem připravené mapě, do které uživatelé zaznamenávají své názory například formou barevných špendlíků. Každá barva má předem daný význam, zde se mi místo líbí, nelíbí, zde se cítím bezpečně, tady rád trávím volný čas apod. Výsledná mapa pak dává přehledný grafický výstup, jak občané vnímají daný prostor.[5] V případě online pocitové mapy jsou respondentům kromě otázek na daný prostor položeny i otázky obecné jako věk, pohlaví apod. Tato metoda ovšem nejde příliš do hloubky.

Pocitové mapy[24] jsou v dnešní době využívány obcemi jako nástroj pro získání podpůrných podkladů pro plánování veřejného prostranství, investiční akce apod. Jejich výhodami jsou nejmenší časová a organizační náročnost. Nevýhodou je, že výstupy respondentů nejdou příliš do hloubky, získáme pouze seznam zadaných bodů, kde jednotlivé body mohou být případně doplněny krátkým komentářem.

Pocitové mapy se dají využít pro získání nejrůznějších podkladů pro plánování ve městě. Mohou sloužit k získání informací o dopravě, kdy otázky jsou například: tady se nedostanu pomocí MHD, tady mám problém zaparkovat. Mohou se ovšem také využít i pro získání pokladů o subjektivních bezpečných či nebezpečných lokalitách, tyto informace pak může využít městská policie.



Obrázek 2.1: Ukázka pocitové mapy. Převzato z [6]

## 2.4 Mapová aplikace

V rámci webové aplikace, která má zobrazovat mapy, je potřeba mít zdroj podkladových map. Tato data v dnešní době poskytuje několik společností, každá společnost má ovšem

své podmínky, za kterých je možné mapy používat. Jsou to společnosti jako Google a jejich platforma Google Maps, Seznam s Mapy.cz, Český úřad zeměměřický a katastrální (ČÚZK) nebo komunita OpenStreetMap. Každá platforma má své specifické API, přes které se mapové podklady načítají do aplikací. Typicky probíhá toto načítání pomocí Web Map Tiled Service (WMTS) či Web Map Service (WMS). K dispozici jsou různé typy podkladových map a aplikace v dnešní době nechávají na uživateli volbu, kterou podkladovou mapu chce použít.

Mapové aplikace v dnešní době umožňují standardně přepínat mezi podkladovými mapami, přibližovat a oddalovat a samozřejmě se na mapě posouvat. Mapa je tvořena typicky čtvercovými obrázky v různé úrovni detailu, které jsou dynamicky načítány. O toto načítání se starají typicky poskytované knihovny společnosti, od které aplikace využívá mapové podklady. Samotné načítání probíhá asynchronně a nedochází tak k blokování aplikace čekáním na přijetí dat. U mapové aplikace, která má zaznamenávat informace o místech a vztahu obyvatel k těmto místům, je potřeba, aby se dala tato místa zaznačit do mapy. Pro tuto interakci se využívá další vrstva v mapové aplikaci, kdy nejnižší vrstva je samotná podkladová mapa tvořena čtvercovými obrázky, nad ní mohou být další vrstvy s informacemi jako jsou například názvy ulic či městských částí a do další vrstvy jsou pak ukládány grafiky vyznačených geometrií. K tomuto vyznačování se používají nástroje knihovny na vykreslení mapy. Knihovny typicky umožňují vyznačovat linie, polygony či pouhé body do mapy.

Zaznačené geometrie mohou mít atributy, které uživatel může zobrazit a editovat. Tyto atributy se dají využít k zaznamenání zpětné vazby od uživatelů, jako jsou třeba nápady či názory. Tyto geometrie je pak nutné uložit do databáze a umožnit je následně exportovat do souborů pro další zpracování. K uložení zadaných odpovědí v mapě se používá formát GeoJSON, který je specifikován od roku 2016 pomocí RFC 7946 a umožňuje zakódovat různé geografické datové struktury.

## 2.5 Google Maps<sup>1</sup>

Jedná se o webovou mapovou aplikaci od společnosti Google. Služba obsahuje mapy celého světa, oproti českým poskytovatelům jsou však mapy méně podrobné. Společnost nabízí pro ovládání a vkládání mapového obsahu javascriptové API rozhraní, toto rozhraní[9] je pro komerční potřeby placené. Abychom mohli tuto službu využívat, je potřeba mít API klíč, který zasíláme s každým požadavkem. V dotazu na server uvádíme úroveň detailu, rozměry a vrstvy, které chceme získat.

## 2.6 Mapy.cz<sup>2</sup>

Společnost Seznam.cz vlastní webový mapový portál mapy.cz a zaměřuje se v něm na oblast České republiky. Stejně jako společnost Google nabízí ke svým mapám webové API rozhraní, toto rozhraní ovšem poskytuje zdarma i pro komerční účely. API rozhraní Mapy.cz[7] umožňuje zobrazení map, ale jeho možnosti jsou značně omezené oproti rozhraní od společnosti Google.

---

<sup>1</sup>[cloud.google.com/maps-platform](https://cloud.google.com/maps-platform)

<sup>2</sup>[www.mapy.cz](https://www.mapy.cz)

## 2.7 OpenstreetMap<sup>3</sup>

Jedná se o open-source službu, která nabízí zdarma mapovou službu i pro komerční použití. Služba OpenstreetMap[8] obsahuje několik druhů map, od map cyklostezek, přes mapu světa až po námořní mapu. Komunikace s API probíhá pomocí HTTP požadavků a klient opět specifikuje o jaká data má zájem pomocí upřesnění vrstev, souřadnic atd.

## 2.8 Český úřad zeměměřický a katastrální - ČÚZK

Tato organizace poskytuje přístup k prostorovým datům v rámci svého rozhraní. Část rozhraní poskytuje právě prohlížečské služby, kde je k dispozici základní mapa ČR, ortofoto mapa ČR, a to v souřadnicovém systému Web Mercator. Dále poskytuje prohlížečské data jako česká jména moří a států, geomorfologické jednotky či data z registru RÚIAN.

Přístup k těmto zmíněným datům je možný pomocí různých standardů. Podporovány jsou WMS, WMTS nebo proprietární rozhraní od ArcGIS Serveru, které je založeno na REST technologii a využívá stejně jako WMTS předgenerované mapy do dlaždic.

## 2.9 ArcGIS Online<sup>4</sup>

Jedná se o webový nástroj společnosti Esri, který umožňuje vytvářet vlastní mapové podklady nebo použít již existující. Jsou v něm dostupné aplikace pro tvorbu vlastních mapových aplikací, například aplikací pro vizualizaci sesbíraných dat. Umožňuje používat ArcGIS API a SDK pro vytváření nových webových nebo mobilních aplikací. Na této platformě zveřejňují data organizace jako ČÚZK, a tak jsou uživatelům i vývojářům k dispozici podrobné mapové podklady za Českou republiku.

Magistrát města Brna tento nástroj používá právě k vizualizaci sesbíraných dat a jejich dalšímu zpracování v tomto GIS řešení. Díky tomu je tak možné používat API rozhraní, podkladové mapy nebo serverové služby ArcGIS v dalších aplikacích, například právě ve webové platformě pro sběr názorů obyvatel.

---

<sup>3</sup>[www.openstreetmap.org](http://www.openstreetmap.org)

<sup>4</sup>[www.esri.com/en-us/arcgis/products/arcgis-online](http://www.esri.com/en-us/arcgis/products/arcgis-online)

## Kapitola 3

# Existující řešení

V oblasti participativního plánování veřejných prostranství již existují online řešení ke sběru informací a názorů obyvatel pomocí pocitových map v České republice i v zahraničí. V České republice je nejpoužívanější aplikace [www.pocitovemapy.cz](http://www.pocitovemapy.cz), v zahraničí jsou to řešení [maptionnaire.com](http://maptionnaire.com), [survey123.arcgis.com](http://survey123.arcgis.com), [socialpinpoint.com](http://socialpinpoint.com) a [carticipe.net](http://carticipe.net). Nástroje umožňují sběr informací za pomoci online mapy daného místa i vyhodnocení sesbíraných informací a názorů.

### 3.1 Pocitové Mapy<sup>1</sup>

Řešení Pocitové Mapy[24] umožňuje sběr názorů obyvatel pomocí zaznamenávání do mapy na základě otázek. Ke každému zaznamenanému místu je možné přidat komentář. Zaznamenání je možné jako plocha, linie, plocha jako volná kresba či konkrétní místo jako bod. Po zodpovězení otázek týkajících se pocitů jsou zobrazeny otázky na věk, pohlaví a další upřesňující informace o daném respondentovi. Jedná se o podobný nástroj jako je v zahraniční služba Maptionnaire.

Toto řešení je implementováno jako webová aplikace, která využívá open-source knihovny Leaflet a jQuery. Program nevyžaduje po respondentech registraci ani instalaci softwarových doplňků. K ukládání sesbíraných dat využívá databázi MySQL a formát GeoJSON. Umožňuje změnit podkladovou mapu, využívá mapy od více subjektů jako Seznam.cz, ČÚZK, OpenStreetMap či ESRI. Dále využívá knihovnu Leaflet pro interaktivní práci s mapami.

### 3.2 Maptionnaire<sup>2</sup>

Tato aplikace je využívána v mnoha zemích a jedná se o nástupce softwaru softGIS od finských výzkumníků z Aalto University. Nástroj je opět pojat jako webová aplikace, která zaznamenává pocity respondentů do online mapy. Maptionnaire[1] umožňuje navíc zobrazit v mapě otázky ostatních respondentů, kteří pak mohou spolupracovat a diskutovat mezi sebou. Služba umožňuje klást respondentům různé typy otázek, včetně otázek na než se odpovídá kreslením do mapy nebo zaznačením místa v mapě. Umožňuje zobrazovat specifické otázky na základě předchozích odpovědí respondenta v průzkumu, dále zabezpečit průzkum heslem nebo specifikovat, které otázky jsou povinné a které volitelné. O respondentech si systém ukládá meta-informace jako velikost obrazovky, použitou podkladovou mapu nebo

---

<sup>1</sup>[www.pocitovemapy.cz](http://www.pocitovemapy.cz)

<sup>2</sup>[www.maptionnaire.com](http://www.maptionnaire.com)

čas strávený v dotazníku. Toto řešení je používáno například městy jako jsou Stockholm, Helsinky, Sydney nebo Singapore.

Řešení je postaveno jako webová aplikace, která jako zdroj map využívá službu Mapbox.com. Dále také umožňuje využít vlastní WMS server, nahrát vlastní GeoJSON či GeoTIFF data. Služba nabízí API pro integraci s nástroji třetích stran, nevyžaduje instalaci žádných doplňků a využívá knihovnu Leaflet.

### 3.3 Social Pinpoint<sup>3</sup>

Jedná se o webovou aplikaci, která poskytuje interaktivní mapování názorů včetně diskuze nad názory, které jsou vidět v mapě. Webová aplikace Social Pinpoint[12] umožňuje pro každý projekt definovat jeho parametry, jako jsou zdroj map, výchozí pozice mapy nebo definovat či nahrát vlastní vrstvy mapy. Podporované jsou formáty GeoJSON nebo Shapefile. Jako zdroj map je podporován vlastní ArcGIS server, vlastní mapy na Mapbox či mapy společnosti Google.

---

<sup>3</sup>[www.socialpinpoint.com](http://www.socialpinpoint.com)

## Kapitola 4

# Existující nástroje a standardy

V této kapitole jsou popsány různé nástroje a standardy, které se používají pro práci s mapou. Jsou rozebrány technologie pro načítání mapy ze serveru a standard pro ukládání geografických struktur.

### 4.1 GeoJSON - RFC 7946

Jedná se o formát pro popis geografických struktur pomocí formátu JSON. Pomocí tohoto formátu lze zakódovat geografické struktury bod, linie, polygon, vícenásobný bod, mnohonásobná linie nebo mnohonásobný polygon. GeoJSON[16] formát od svého zveřejnění v roce 2008 zažívá narůstající zájem, především v API rozhraních a webových mapových službách.

GeoJSON objekt[16] je objekt ve formátu JSON, obsahuje položku `type`, která musí obsahovat hodnotu jednoho z typů formátu GeoJSON. Většina objektů tohoto formátu má dále položku se souřadnicemi. Pokud se jedná o objekt reprezentující bod, je položka souřadnice pouze jedna konkrétní pozice. Položka souřadnice může být také pole souřadnic, pole se používá v případě, že je popisován objekt typu vícenásobný bod či linie. Jestliže potřebujeme popsat polygon, tak ten je popsán pomocí linií. Konkrétní pozice je definována jako pole čísel, které musí obsahovat alespoň dvě čísla, nejčastěji vyjadřující zeměpisnou šířku a délku.

```
{
  "type": "Feature",
  "geometry": {
    "type": "LineString",
    "coordinates": [
      [16.604427409387046, 49.23121867778965],
      [16.587776255822615, 49.225613546843775]
    ],
  },
  "properties": {
    "text": "Lorem ipsum"
  }
}
```

Výpis 4.1: Ukázka GeoJSON formátu

## 4.2 WMS - Web Map Service

Webová mapová služba[14] je standard, který vyvinula organizace Open Geospatial Consortium a dále je i touto organizací vyvíjen. Server pracuje na principu klient-server, kdy server poskytuje jednoduché HTTP rozhraní pro získání obrazových dat z geoprostorových databází. Klient v požadavku specifikuje o jaké vrstvy a oblast má zájem, server následně požadavek zpracuje a zašle klientovi odpověď ve formě jednoho či více rastrových obrázků, které jsou připraveny přímo dle jeho požadavku. Tato rastrová data lze pak přímo zobrazit v prohlížeči, protože se jedná o formát obrázků JPEG, PNG apod. Data poskytnutá serverem jsou georeferencována, lze je tedy jednoznačně umístit v rámci výsledné mapy. Ke komunikaci klienta se serverem je využíván protokol HTTP, typicky jeho metody POST a GET.

Protože jsou data serverem poskytována pro každého klienta specificky dle jeho požadavku, nelze využít ukládání dat do mezipaměti a zrychlit tak načítání. Při narůstajícím počtu klientů a různorodých požadavků může být problém s odezvou serveru. Může se tedy stát, že uživatel bude muset na mapová data příliš dlouho čekat, než je pro něj server připraví.

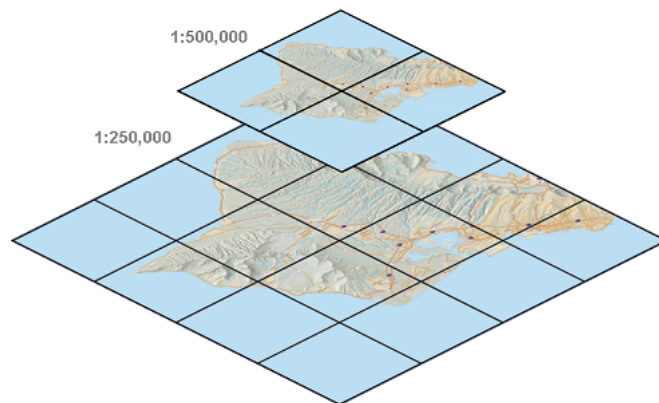
Webová mapová služba je typicky provozována v rámci mapového serveru, který obsahuje databázi geografických dat. Toto rozhraní nabízí velké množství organizací, v České republice se jedná například o katastry, které přes tento protokol zpřístupňují katastrální mapy, ortofoto mapy a další služby.

## 4.3 WMTS - Web Map Tile Service

WMTS[4] je standard, který spravuje Open Geospatial Consortium. Jedná se o alternativní strategii k WMS, kdy mapová data jsou předem předgenerována do formy dlaždic. Tyto dlaždice musí být předem vygenerovány pro různá měřítka a souřadnicové systémy, protože server zasílá pouze dlaždice vytvořené dle požadavku uživatele, kdy požadavkem je konkrétní dlaždice. Server již neprovádí žádné zpracování obrazu. Toto řešení umožňuje využití mezipaměť jak na serveru, tak i u poskytovatelů obsahu či připojení a tím snížit odezvu a zvýšit dostupnost služby.

Tuto službu lze realizovat pomocí webového serveru[21], který má uloženy předgenerované dlaždice v různých měřítkách a v daném souřadnicovém systému. Tyto dlaždice server odesílá pomocí standardního HTTP protokolu klientovi. Kromě samotných mapových dlaždic webový server poskytuje statický XML soubor popisující parametry dané služby, jako je právě informace o dostupných měřítkách, souřadnicovém systému a informace jak žádat o konkrétní dlaždice.

Toto rozhraní nabízí velké množství organizací jako Google Maps, Yahoo Maps či Microsoft, v České republice se jedná například o Český úřad zeměměřický a katastrální, který přes tento protokol opět zpřístupňuje katastrální mapy, ortofoto mapy a případně další služby.



Obrázek 4.1: Ukázka WMTS dlaždic. Převzato z [4]

## 4.4 Knihovna Leaflet

Leaflet je javascriptová open-source knihovna, která je hodně využívána existujícími řešeními. Je to knihovna, která umožňuje interaktivní práci s mapami na počítačích i mobilních zařízeních, má širokou podporu prohlížečů a nevyžaduje žádné externí knihovny pro svou činnost jako třeba jQuery. Samotná knihovna lze rozšířit pomocí doplňků.

V základu umí načítat data pomocí standardu WMS a přidávat geometrie, ale neumožňuje uživateli vytvořit tyto geometrie v mapě. Musí se přidávat v kódu. Pro načítání dat z WMTS je potřeba doplněk od jiného autora, pro vytváření a editaci geometrií uživatelem je k dispozici doplněk, který je opět od jiného tvůrce.

## 4.5 ArcGIS JS API

Jedná se o JavaScriptové rozhraní pro práci s mapovými podklady, umožňuje zobrazovat 2D i 3D mapy. Nejčastěji se používá ve spojení s ArcGIS servery. Obsahuje knihovny pro práci s různými mapovými vrstvami. Dále obsahuje knihovny pro přidávání objektů do mapy, různé druhy mapových vrstev a podkladové mapy je možné získat snadno z ArcGIS Online platformy, která je výchozím zdrojem této knihovny. Umožňuje podporu vizualizace dat nebo práci s lokalizačními službami a další vlastnosti, které mohou využít specialisté v oblasti GIS.

Pro svou činnost nevyžaduje žádné další knihovny, podporuje lokalizaci rozhraní a také zobrazení na mobilních telefonech díky responzivnímu vzhledu. Aktualizace vychází pravidelně několikrát ročně. Pro použití je potřeba mít předplatné ArcGIS pro vývojáře nebo ArcGIS Online. V současné době je aktuální verze z března 2019 a to verze 4.11.



## Kapitola 5

# Analýza požadavků

V této kapitole je krátce popsán problém, který tato bakalářská práce řeší a samotná analýza požadavků, které vyplývají z tohoto problému a byly specifikovány ve spolupráci s úředníky Magistrátu města Brna.

### 5.1 Popis problému

Město Brno potřebuje sbírat názory a podněty od občanů jednotlivých městských částí v prostoru prostřednictvím pocitových online map. Potřebují mít webový portál, ve kterém si budou moci sami vytvářet jednotlivé průzkumy, upravovat je a stahovat si výsledky průzkumů. Tyto výsledky pak budou dále zpracovávat v již existujících nástrojích, které v současné době používají.

Cílem této práce je tedy navrhnout a implementovat webový portál, který splní požadavky na vytváření vlastních průzkumů. Má umožnit definovat si jednotlivé otázky a možné odpovědi, dále zobrazovat průzkumy uživatelům a ukládat si jejich odpovědi a umožnit jejich exportování ve formátu vhodném pro další zpracování.

### 5.2 Požadavky aplikace

Aplikace má mít dvě části, jednu pro širokou veřejnost a druhou pro úředníky města Brna. Část pro širokou veřejnost má sloužit k získávání názorů obyvatel v prostoru pomocí mapy a druhá část má sloužit úředníkům k vytváření průzkumů, jejich administraci a stahování sesbíraných názorů od obyvatel. Tyto názory mohou být na konkrétní místa či lokality, ale tento nástroj by měl umožnit také zaznačit trasy jako například cestu do školy apod.

Každý průzkum má svůj název a případně podnázev, dále obsahuje stav průzkumu, datum začátku a konce průzkumu, uvítací a děkovaný text, který se zobrazuje respondentům a také lokalitu, ve které má průzkum probíhat. Datum začátku a konce průzkumu omezuje dobu trvání sběru dat, která souvisí i se stavem průzkumu, kdy sběr dat má být umožněn jenom, pokud je průzkum aktivní. Průzkum obsahuje sadu obecných a geografických otázek.

Obecné otázky obsahují samotný dotaz, typ otázky, příznak povinné odpovědi, pořadí otázky a seznam možných odpovědí. Existuje několik typů otázek, kdy rozlišujeme otázky s textovou odpovědí, číselnou odpovědí nebo výběrové otázky. Seznam možných odpovědí se zadává pouze u výběrových otázek, ale v jednotlivých možnostech odpovědi je potřeba umět vytvořit textový vstup pro respondenty. U výběrových otázek rozlišujeme, jestli respondent vybírá právě jednu nebo více nabízených odpovědí.

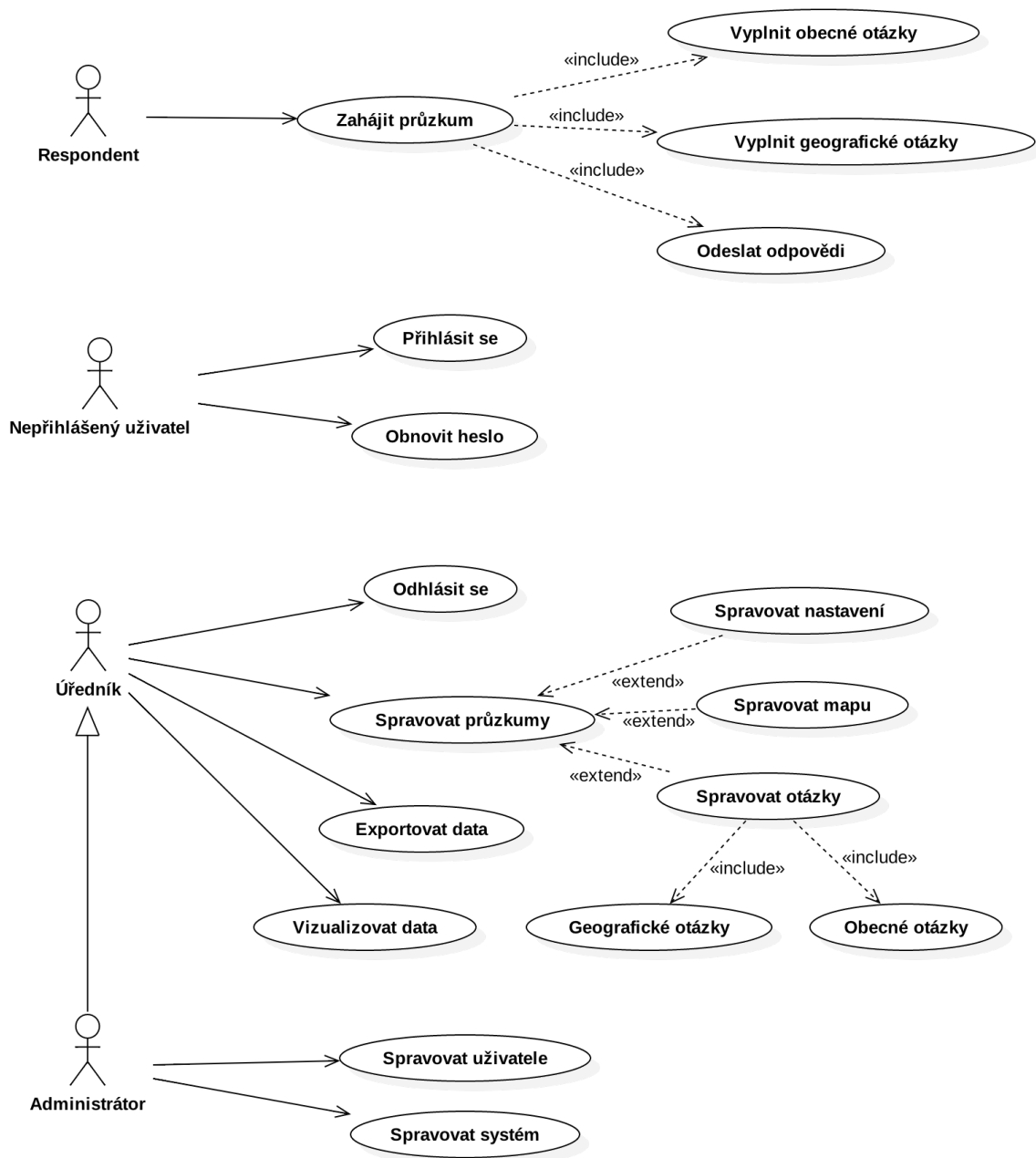
Geografické otázky jsou otázky, na které se odpovídá formou značení míst v mapě. Toto místo může být konkrétní bod, polygon nebo linie. Každá otázka má dotaz, na který se značí odpovědi v mapě formou zadaných značek. Seznam značek zadává úředník a každá značka má určitý typ (polygon atd.), barvu a popis dané značky. Respondent pak v průzkumu u otázky vybere značku a umístí ji do mapy. Ke značce je u některých otázek možné přidat i textový komentář. Přidávání komentářů je řízeno pomocí příznaku u dané otázky, který musí autor průzkumu zadat.

Respondent má možnost vyplnit průzkumu pouze jednou, další vyplnění mu nemá být umožněno, pokud je u průzkumu zadán příznak unikátnosti respondentů. Po spuštění průzkumu budou nejprve zobrazeny obecné otázky a po jejich vyplnění pak mapa s geografickými otázkami. Respondent má mít volbu podkladové mapy, možnost v mapě se pohybovat, přibližovat či oddalovat. Zadávat odpovědi musí být umožněno pouze ve zkoumané lokalitě průzkumu.

Protože Magistrát města Brna nemá žádný použitelný systém pro správu uživatelů, který by umožňoval napojení tohoto systému, je požadováno, aby tento portál měl implementovanou vlastní správu uživatelů pro administraci průzkumů. V systému jsou předpokládány dvě role, a to role administrátora a uživatele. Role uživatel má možnost spravovat průzkumy – vytvářet, upravovat či stahovat výsledky, role administrátor má oproti roli uživatele navíc oprávnění spravovat uživatele. Uživatelé mají mít možnost obnovit si své heslo.

Data od respondentů sesbíraná v průzkumu má systém exportovat ve formátu vhodném pro další zpracování nástroji, které magistrát města Brna již používá. Jedná se o webový nástroj ArcGIS online, který byl krátce popsán v kapitole 2.9. Po ukončení průzkumu má systém místo průzkumu zobrazovat základní vizualizaci sesbíraných odpovědí. Tato vizualizace se bude zobrazovat, pokud je stav průzkumu ukončen, ale není ještě archivován. Průzkum, který je ve stavu archivován nelze vyplňovat, nelze u něj ani zobrazit vizualizaci, ale lze stahovat sesbíraná data od respondentů.

Výstupem analýzy požadavků je diagram případů užití, který je vidět na obrázku 5.1. V tomto diagramu případů užití jsou znázorněny funkcionality systému, jednotlivé uživatelské role a akce, které vedou k dosažení požadovaného cíle.



Obrázek 5.1: Diagram případů užití

## Kapitola 6

# Návrh systému

Cílem tohoto řešení je vytvořit platformu pro sběr názorů od obyvatel města Brna, tuto platformu s ohledem na cílové publikum navrhuji jako webovou aplikaci, která tak bude široké veřejnosti snadno dostupná z libovolného webového prohlížeče. V dnešní době se moderní webové aplikace vytvářejí pomocí třívrstvé architektury, která je tvořena prezentační, aplikační a datovou vrstvou. Prezentační vrstva má za cíl vizualizaci dat a je prováděna HTTP klientem, v rámci kterého mohou být spouštěny i javascriptové aplikace. Pro tvorbu webových aplikací jsou často používány javascriptové frameworky Angular<sup>1</sup> a React<sup>2</sup>. V rámci aplikační vrstvy probíhají výpočty a zpracování dat, tato vrstva tak tvoří samotné jádro aplikace, které obsahuje funkce a logiku aplikace. Často používané technologie aplikační vrstvy jsou PHP, Python či ASP.NET. Datová vrstva slouží ke konzistentnímu uložení stavu systému a nejčastěji se jedná o databáze, relační i objektové.

### 6.1 Entity Relationship Diagram – ERD

Tento diagram slouží k modelování perzistentních dat, se kterými bude systém pracovat. Diagram slouží jako podklad pro tvorbu schématu relační databáze a byl navržen na základě předchozí analýzy. Výsledný ER diagram je zobrazen na obrázku 6.1.

Entita `Uživatel` představuje úředníky systému, kteří mohou mít roli administrátor nebo uživatel. Účastníci průzkumu se v reálném světě nazývají respondenti, a proto i v tomto modelu jsou takto nazváni a mají svou vlastní entitu, aby byli odlišeni od uživatelů představujících úředníky, kteří vytvářejí jednotlivé průzkumy. Každý průzkum obsahuje dva druhy otázek, v tomto návrhu jsem se je rozhodl modelovat jako specializace entity `Otázka`, protože oba typy mají část atributů společnou, ale liší se ve vazbách a některých attributech.

Respondenti vytvářejí odpovědi na obecné otázky a názory na geografické otázky. Kvůli plánovanému rozdělení vytváření průzkumu do několika kroků není stanoven počet otázek u průzkumu na minimum jedné otázky v tomto modelu, toto bude ošetřeno až v samotné aplikaci znemožněním změny stavu, pokud není zadána alespoň jedna otázka.

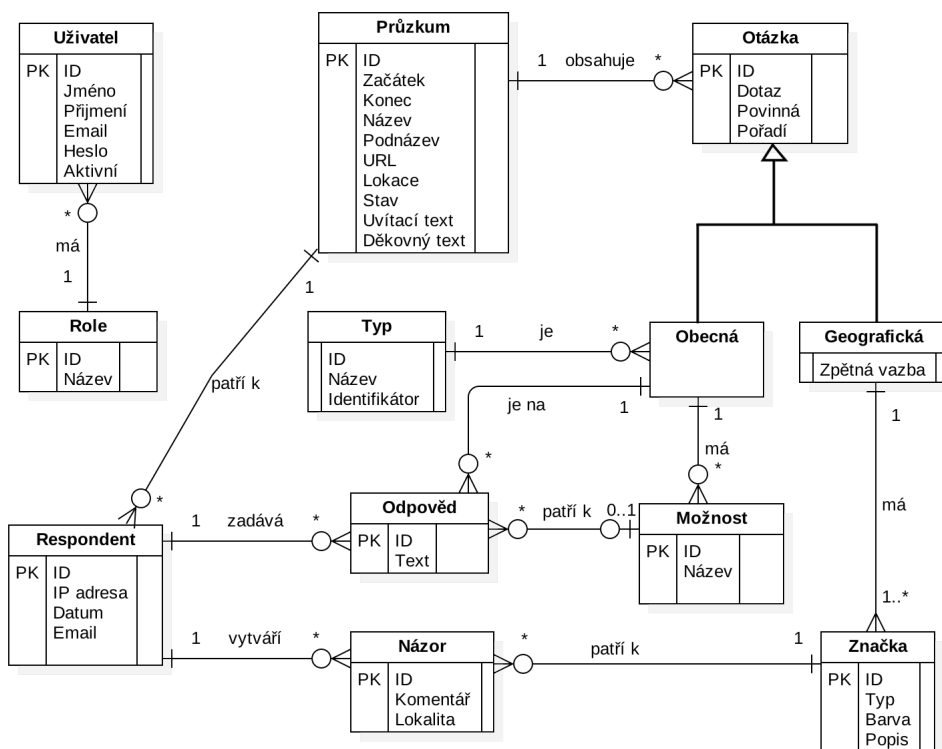
Dle typu obecné otázky pak daná otázka může obsahovat možnosti, které respondent zvolí nebo zadá pouze text jako odpověď. Právě proto otázka nemusí mít žádnou možnost a entita `Odpověď` má vztah jak k obecné otázce, tak i k možnosti. U geografických otázek se přidávají značky a tady je již použito omezení na minimum značek u otázky již v modelu.

---

<sup>1</sup>[angular.io](https://angular.io)

<sup>2</sup>[reactjs.org](https://reactjs.org)

Před samotným začátkem vyplňování průzkumu musí respondent nejprve ověřit svůj email, dle kterého se bude ověřovat počet vyplnění průzkumu, proto diagram obsahuje vztah Respondent-Průzkum a nejsou stanoveny minimální počty odpovědí a názorů na 1. V implementaci je pak počítáno s možností, že průzkum nemusí mít žádné obecné, nebo žádné geografické otázky, portál pak půjde využít čistě k průzkumu bez mapy, nebo ke zjištění názorů v mapě bez obecných otázek, a proto nejsou stanoveny minimální počty odpovědí a názorů u respondenta.



Obrázek 6.1: ER diagram

Díky možnosti zaznamenávat místa do mapy bude možné spolehlivě identifikovat místa, ke kterým mají obyvatelé pozitivní vztah. Výsledný systém bude umožňovat vytvářet jednotlivé průzkumy a k nim otázky. U geografických otázek pak umožní v případě potřeby sbírat zpětnou vazbu od obyvatel.

## 6.2 Webové technologie

Tento webový portál jsem se rozhodl navrhnout jako moderní webovou aplikaci, která bude používat javascriptový framework jehož výběr je popsán dále v textu a API rozhraní postavené na PHP frameworku, jehož výběr je opět popsán dále v textu. Pro ukládání dat bude využita relační databáze.

Single page application[22] (SPA) je webová aplikace, která využívá server jako zdroj dat, ale nedochází k překreslování celé stránky, protože se překreslují jenom ty části stránky, které jsou právě potřeba překreslit. Aplikace je často také rychlejší, protože k překreslení stránky je tak potřeba méně dat. Výhodou tohoto přístupu je, že některé akce uživatele

mohou být provedeny okamžitě, protože není potřeba načítat žádná data ze serveru. Tento typ aplikací využívá javascriptové frameworky, kdy nejnámější jsou Angular a ReactJS. K přenosu samotných dat se využívá AJAX a XHR požadavky.

Angular[11] je javascriptový framework, který usnadňuje vytváření webových klient-ských aplikací. Používá kombinaci vkládání závislostí či deklarativních šablon. Umožňuje vývojářům vytvářet aplikace, které fungují na počítačích i mobilních telefonech, využívá principů objektového programování a využívá jazyk Typescript, což je nadstavba jazyka Javascript. Vývoj je pod kontrolou společnosti Google.

React[13] je opět javascriptový framework pro tvorbu SPA a má relativně málo závislostí. Je založený na komponentách, používá virtuální DOM a zapisuje se v JavaScript XML. Jeho vývoj je pod kontrolou společnosti Facebook. Na rozdíl od ostatních frameworků je potřeba využít další knihovny jako rozšíření pro podporu směrování či interakci s API.

PHP frameworky nám umožňují rychlejší vývoj stabilních aplikací. Ve většině případů využívají koncept Model-View-Controller a často nám usnadňují práci s databází, protože využívají ORM a také není potřeba psát přímo čistě SQL dotazy. K jednotlivým frameworkům často existují i různá rozšíření, která lze používat, jako např. vytváření PDF či odesílání e-mailů, pokud to již neobsahují ve svém základu, což opět zrychluje vývoj aplikace.

Symfony<sup>3</sup> je PHP framework pro tvorbu webových stránek a aplikací založených na jazyce PHP. Na tomto frameworku jsou založeny projekty jako Drupal<sup>4</sup> či phpBB<sup>5</sup>. Tento framework je od začátku vyvíjen s důrazem na jeho rychlost, flexibilitu a rozšiřitelnost. Symfony[3] je aktuálně ve verzi 4.2 a oproti verzi 3.x přináší novou architekturu, strukturu i konfigurace, došlo k odstranění zastaralých částí kódu a tedy ke zrychlení celého frameworku. V nové verzi došlo k přepracování závislostí jednotlivých částí frameworku a Symfony tak lze lépe využít i pro mikro-aplikace, například i na tvorbu API rozhraní.

Další framework pro tvorbu webových stránek v jazyce PHP je Laravel<sup>6</sup>. V současnosti je ve verzi 5.8 a autorem je Taylor Otwell. Laravel[2] nabízí spoustu modulů, například pro validaci dat, směrování nebo práci s databází. Pro práci s databází se využívá stejně jako v Symfony objektově relační mapování.

Tuto webovou platformu jsem se rozhodl vytvořit pomocí JavaScriptové aplikace založené na frameworku Angular, která bude komunikovat s API rozhraním. Pro interaktivní práci s mapou byla zvolena javascriptová knihovna mapového rozhraní od Esri. Vlastní API rozhraní bude využívat serializaci JSON a na základě osobních zkušeností s PHP frameworkem Symfony jsem se ho rozhodl použít pro implementaci samotného API rozhraní.

Navržený systém umožňuje sbírat názory obyvatel v prostoru, využívá k tomu pocitovou mapu, do které respondenti zaznamenávají své pocity pomocí značek a mohou k těmto místům přidat i slovní komentář, který může být názor či nápad k danému místu. Sběr dat probíhá v rámci průzkumu, ve kterém jsou předem definovány otázky. Sbíraná data z jednotlivých průzkumů je možné exportovat a po ukončení průzkumu vizualizovat přímo ve webovém portálu, ale jedná se pouze o omezenou vizualizaci.

Výsledek implementace navrhuji otestovat ve spolupráci s magistrátem města Brna. Konkrétně vytváření průzkumů a stahování dat přímo na úřednicích, kteří budou s daným systémem pracovat. Dále plánuji testování na respondentech, kteří otestují část aplikace týkající se získávání zpětné vazby a názorů od obyvatel, a jako poslední v řadě provést testování exportování velkého množství odpovědí.

---

<sup>3</sup>[www.symfony.com](http://www.symfony.com)

<sup>4</sup>[www.drupal.org](http://www.drupal.org)

<sup>5</sup>[www.phpbb.com](http://www.phpbb.com)

<sup>6</sup>[www.laravel.com](http://www.laravel.com)

# Kapitola 7

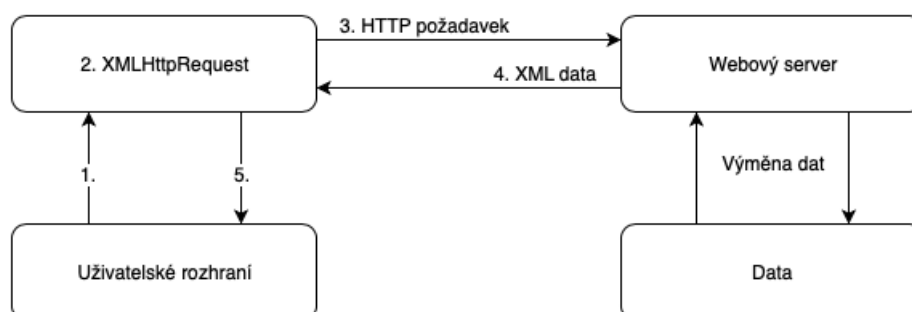
## Implementace

Implementace vychází z analýzy požadavků a návrhu systému popsaného v předchozích kapitolách. Následující kapitoly popisují implementaci jednotlivých částí této webové platformy pro sběr názorů od obyvatel.

### 7.1 Serverová část aplikace

Aplikace v PHP je implementována pomocí frameworku Symfony a poskytuje jednoduché webové API rozhraní pro klienty. Samotné rozhraní je postaveno na REST architektuře a pro autentizaci klientů jsou použity tokeny validované na serveru. REST architektura je bezstavová a nedochází tak k ukládání žádného stavu na serveru. Požadavek na aplikační rozhraní tak musí obsahovat všechny údaje. Pro autentizaci je proto potřeba využít jiný mechanismus než sessions. Mechanismy, které lze použít, jsou například JSON Web Token (JWT), OAuth nebo HTTP Basic.

Asynchronní JavaScript a XML[18] (AJAX) je technika, která umožňuje měnit část webových stránek bez nutnosti danou stránku načíst celou ze serveru. Všechny moderní prohlížeče tuto techniku podporují a není potřeba instalovat žádný doplněk pro její podporu. Základní složkou AJAXu je JavaScript, který manipuluje s jednotlivými částmi stránky. Dále je to objekt `XMLHttpRequest`, který umožňuje asynchronně komunikovat se serverem, kdy tato komunikace je vedena pomocí HTTP protokolu. Protože komunikace probíhá asynchronně, nedochází tak k blokování stránky a uživatel s ní může nadále pracovat bez přerušení. V této implementaci je používána tato technika frameworkem Angular k získávání dat ze serveru, například seznamu průzkumů nebo ukládání odpovědí respondentů.



Obrázek 7.1: Princip AJAXu

JSON Web Token[19] je řetězec, který se skládá ze tří částí, hlavičky, obsahu a podpisu. Tyto tři části se zakódují pomocí kódování base64 a spojí se do jednoho řetězce, ve kterém jsou jednotlivé části odděleny tečkou. Hlavička obsahuje minimálně informace o typu tokenu a použitém hashovacím algoritmu, tělo obsahuje informace o samotném uživateli a expiraci tokenu. Podpis obsahuje otisk vytvořený hashovací funkcí spojením hlavičky a těla, a tedy slouží k ověření integrity tokenu.

Same-origin policy je bezpečnostní politika uplatňovaná v prohlížečích, která umožňuje skriptu posílat požadavky jen na zdroje, které mají stejný origin, tedy že cíl požadavku má stejné schéma, název hostitele a port jako je URL skriptu, který tento požadavek poslal. Toto chování je možné upravit pomocí Cross-Origin Resource Sharing (CORS). CORS[20] umožňuje definovat toto chování pomocí hlaviček HTTP odpovědi ze serveru. Samotné rozvolnění se provádí nastavením hlavičky `Access-Control-Allow-Origin` a případně dalších hlaviček.

V této implementaci je základní jádro PHP frameworku Symfony rozšířeno o balíček `LexikJWTAuthenticationBundle`<sup>1</sup>, který se stará o autentizaci klientů pomocí JSON Web Token v REST požadavcích. Protože samotná PHP aplikace může běžet na jiném hostiteli než klient, je jádro Symfony rozšířeno ještě o podporu Cross-Origin Resource Sharing, kterou přidává balíček `NelmioCORSBundle`<sup>2</sup>. Tento balíček implementuje podporu pro zpracování preflight požadavků a do hlavičky HTTP odpovědi přidává `Access-Control-Allow-Origin` a případně další potřebné položky dle nastavení v konfiguračním souboru.

Kontroléry jsou třídy, které zpracovávají příchozí HTTP požadavky. Pomocí entit a repositářů pracují s daty, provádějí nad nimi operace, modifikují data a vracejí odpovědi ke klientům. V této implementaci se používá JSON serializace a tak příchozí data v požadavcích i odchozí data v HTTP odpovědích jsou serializována pomocí JSON. Jednotlivé metody, které obsluhují příchozí HTTP požadavky jsou identifikovány pomocí cesty a nazývají se koncové body. Seznam těchto koncových bodů je v tabulce 7.1.

- `AbstractApiController` – jedná se o abstraktní kontrolér, ve kterém jsou implementovány metody pro ukládání objektů do databáze a vytváření odpovědí, jejich serializaci a nastavení správného návratového kódu HTTP odpovědi
- `DataExportController` – obsahuje metody pro export sesbíraných dat z průzkumů, data jsou pomocí tohoto kontroléru získána z databáze, formátována a odeslána zpět na klienta
- `GeneralQuestionController` – obsahuje metody pro získání seznamu podporovaných typů obecných otázek
- `MarkerController` – obsahuje metody pro získání seznamu podporovaných typů značek pro geografické otázky
- `PasswordController` – obsahuje metody pro generování klíče k zapomenutému heslu a nastavení si nového heslo
- `ProfileController` – obsahuje metody pro práci s profilem přihlášeného uživatele systému

---

<sup>1</sup>[github.com/lexik/LexikJWTAuthenticationBundle](https://github.com/lexik/LexikJWTAuthenticationBundle)

<sup>2</sup>[github.com/nelmio/NelmioCORSBundle](https://github.com/nelmio/NelmioCORSBundle)



- RespondentController – obsahuje implementaci metod pro práci respondentů s průzkumem, získání obecných a geografických otázek, obecných informací o průzkumu apod.
- SurveyController – implementuje operace pro práci s průzkumem v administraci, vytváření, úpravy a smazání průzkumu
- UserController – implementuje logiku pro správu uživatelů systému, jejich vytváření, upravování a mazání
- VerificationController – obsahuje metody pro opětovné vygenerování potvrzovacího klíče uživatele a potvrzení účtu uživatele
- VisualisationController – obsahuje metody pro získání sesbíraných objektů v mapě dle otázky v průzkumu, tato data o objektech jsou ve formátu GeoJSON

Příchozí požadavky jsou kontrolovány, jestli obsahují validní informace. Kontroluje se, jestli obsahují definované a povinné položky, a že předávané hodnoty jsou ve správném formátu. Tato kontrola je prováděna pomocí komponenty `OptionsResolver`. Tato komponenta umožňuje provádět validaci, normalizaci a nastavení výchozích hodnot. Nastavení výchozích hodnot je využito v této implementaci například u stránkování seznamu průzkumů. Všechny HTTP požadavky jsou validovány přes tuto komponentu, kdy každý požadavek má definovány hodnoty, které musí platit. Po ověření validity dat požadavku jsou tato data přemapována na objekt, který reprezentuje daný požadavek.

```
$resolver->setDefined([
    'status',
    'start',
    'respondentLimit'
]);
$resolver->addAllowedTypes('status', 'string');
$resolver->addAllowedValues('status',
    SurveyStatusEnum::getAvailableStatuses());
$resolver->addAllowedTypes('start', 'string');
$resolver->addAllowedValues('start', $this->isDate);
$resolver->addAllowedTypes('respondentLimit', 'bool');
$resolver->setDefault('respondentLimit', true);
```

Výpis 7.1: Zkrácená ukázka nastavení validace požadavku pro nastavením průzkumu

Pro ukládání dat je použita relační databáze MySQL, ke které se přistupuje pomocí knihovny Doctrine a objektově relačního mapování. Jednotlivé entity a práce s databází jsou popsány v kapitole [7.2](#).

Před vkládáním nebo upravením záznamů v databázi probíhá nad daným objektem validace, validace je častým úkolem pro webové služby. K tomu se používá komponenta `Validator`, kdy pro každou entitu jsou definovány pomocí anotací pravidla, která musí splňovat její položky. Samotná validace se vykoná při zavolání validátoru pro daný objekt a toto volání se provádí v metodách kontrolérů, které mění záznamy v databázi pro daný objekt. V případě, že dojde k výskytu chyb při validaci, dochází k navrácení těchto chyb pomocí HTTP odpovědi se stavovým kódem 400. Validační pravidla jsou mapována pomocí anotací v samotných entitách.

```

/**
 * @Assert\NotBlank
 * @Assert\Date
 * @Assert\GreaterThan(propertyPath="start")
 * @ORM\Column(type="date")
 */
private $end;

```

Výpis 7.2: Ukázka mapování validačních pravidel

Přidávání uživatelů do systému a změna hesla jsou citlivé operace, které je nutné mít potvrzované, a proto jsou v systému zavedeny potvrzovací klíče, které se odesílají na e-mail. Tyto klíče jsou náhodné řetězce o délce 40 znaků, které jsou zahashovány pomocí algoritmu SHA512 a mají omezenou platnost. Při vytvoření nového uživatele odejde tomuto uživateli potvrzovací URL adresa, na které je potřeba potvrdit účet a potvrzovací klíč na zadaný e-mail, přihlášení je do doby potvrzení daného účtu znemožněno. V administraci jde u daného uživatele odeslat nový potvrzovací klíč, ale pouze pokud je účet nepotvrzený.

Změna hesla pak probíhá tak, že nepřihlášený uživatel si požádá o změnu hesla pro účet dle zadaného e-mailu, pokud se podaří najít účet s tímto e-mailem, vygeneruje se klíč, který se odešle na zadaný e-mail. Následně uživatel musí vyplnit nové heslo a zasláný klíč, který se ověří. Dojde k ověření, že je klíč stále platným, jestliže toto ověření dopadne správně, dojde ke změně hesla. V případě vyplnění neplatného e-mailu (neexistující účet), vrací server pozitivní HTTP odpověď, ale neodesílá žádný klíč na e-mail ani neprovádí žádné změny v databázi. Toto chování je zvoleno záměrně, aby nemohlo docházet k prohledávání databáze uživatelů.

Dále se potvrzovací klíče používají u průzkumů, které mají omezení na unikátní respondenty. Toto omezení vyžaduje, aby respondent směl vyplnit průzkum pouze jedenkrát. Toho je docíleno tak, že respondent je ověřován přes e-mail, na který se mu odešle potvrzovací klíč a průzkum se mu podaří spustit až po zadání tohoto klíče. Jestliže už průzkum respondent vyplnil, vrací se ke klientovi chybová zpráva. V případě, že průzkum nemá omezení na unikátní respondenty, tak se klíče negenerují a respondent může průzkum spustit okamžitě.

Koncové body API rozhraní, které slouží pro administraci průzkumů a správu uživatelů, jsou chráněny pomocí autentizace přes JSON Web Token. Samotné vytvoření autentizačního tokenu probíhá po zaslání e-mailu a hesla v HTTP POST požadavku na koncový bod, který má cestu `/api/auth`. Tuto cestu obsluhuje balíček `LexikJWTAuthenticationBundle`, ten vůči nastavenému poskytovateli uživatelských identit ověří, že zadané údaje jsou platné a v případě úspěchu vrátí v odpovědi vygenerovaný token. Do samotného tokenu jsou přidány informace o uživatelských rolích a identifikátor uživatele.

K uchovávání informací o uživateli systému slouží entita `User`, která ovšem musí implementovat rozhraní `UserInterface`. Toto rozhraní definuje metody, které využívá autentizační vrstva frameworku. Pro kontrolu, že je uživatel verifikován a není zakázán, slouží implementovaná třída `UserChecker`, která v případě, že uživatel nesplňuje tyto podmínky vyvolá výjimku `AccountExpiredException` a tím přeruší proces autentizace, čímž dojde k navrácení HTTP odpovědi s kódem 401. Hesla jsou do databáze ukládána v šifrované podobě pomocí algoritmu `bcrypt`.

Název	Metoda	Cesta
surveys_export_data	GET	/api/surveys/{surveyId}/export
surveys_export_opinions	GET	/api/surveys/{surveyId}/export/{questionId}
general_type_list	GET	/api/general/types
marker_type_list	GET	/api/markers/types
users_password_token	POST	/api/users/password/token
users_password	POST	/api/users/password
users_get_profile	GET	/api/users/profile
users_update_profile	POST	/api/users/profile
respondents_get_survey	GET	/api/respondents/{url}
respondents_get_general_questions	GET	/api/respondents/{url}/general
respondents_get_geo_questions	GET	/api/respondents/{url}/geo
respondents_token_create	POST	/api/respondents/token
respondents_token	POST	/api/respondents/verify
respondents_save_responses	POST	/api/respondents/{url}
surveys_list	GET	/api/surveys/{page}
surveys_create	POST	/api/surveys
surveys_get_settings	GET	/api/surveys/{id}/settings
surveys_update_settings	PUT	/api/surveys/{id}/settings
surveys_get_location	GET	/api/surveys/{id}/location
surveys_update_location	PUT	/api/surveys/{id}/location
surveys_get_questions	GET	/api/surveys/{id}/questions
surveys_update_questions	PUT	/api/surveys/{id}/questions
surveys_delete	DELETE	/api/surveys/{surveyId}
users_list	GET	/api/users
users_create	POST	/api/users
users_update	PUT	/api/users/{userId}
users_delete	DELETE	/api/users/{userId}
users_get	GET	/api/users/{userId}
users_verification	POST	/api/users/verification
users_resend_token	POST	/api/users/{userId}/resend_token
visualisations_get_geo_questions	GET	/api/visualisations/{url}/questions
visualisations_get_opinions	GET	/api/visualisations/{url}/questions/{question}
api_auth	ANY	/api/auth

Tabulka 7.1: Tabulka koncových bodů API rozhraní

## 7.2 Schéma databáze

K ukládání perzistentních dat je využita relační databáze MySQL. Pro samotnou práci s relační databází je využit ORM framework Doctrine, který je součástí Symfony. Knihovna Doctrine využívá principu objektově relačního mapování (ORM), což umožňuje pracovat se záznamy v tabulce jako s objekty. Pro každou entitu z ER diagramu se vytvoří třída a pomocí anotací se instanční proměnné namapují na sloupce tabulky.

```
/**
 * @ORM\Column(type="string", length=255, nullable=true)
 */
private $subtitle;
```

Výpis 7.3: Ukázka mapování instanční proměnné

Každá entita má pak svůj repozitář, ve kterém jsou implementovány metody pro práci s danou entitou. Nejčastěji se jedná o metody, které obsahují kód pro dotazování se nad entitami, jedná se většinou o složitější dotazy, než je jenom získání záznamu dle identifikátoru. Jako příklad lze uvést implementovanou metodu `findActiveSurveyByUri`, která vrací informace o průzkumu podle jeho URI adresy. Tato metoda ovšem vrací konkrétní informace o průzkumu pouze v případě, že průzkum je v aktivním stavu a dále je aktuální datum v rozmezí nastaveného začátku a konce průzkumu.

```
public function findActiveSurveyByUri(string $uri)
{
    return $this->createQueryBuilder('s')
        ->select('s.id, s.title, s.subtitle, s.welcome,
            s.respondentLimit, s.thankyou, s.location')
        ->where('s.start <= :currentDate')
        ->andWhere('s.end >= :currentDate')
        ->andWhere('s.url = :url')
        ->andWhere('s.status = :status')
        ->setParameter('currentDate',
            new \DateTime('now', new \DateTimeZone('UTC')))
        ->setParameter('url', $uri)
        ->setParameter('status', SurveyStatusEnum::STATUS_ACTIVE)
        ->getQuery()
        ->getOneOrNullResult(Query::HYDRATE_ARRAY);
}
```

Výpis 7.4: Ukázka metody v repozitáři

SQL injection je typ útoku, který se snaží napadnout aplikaci přes databázovou vrstvu pomocí vsunutí cizího kódu přes nezabezpečený vstup a tím vykonání pozměněného SQL příkazu. Je proto potřeba mít ošetřeny vstupní argumenty, které jsou vkládány do SQL příkazů v repozitářích. Pomocí tohoto útoku lze například během přihlašování místo jména či heslo vyplnit SQL příkaz na změnu hesla, který se provede díky pouhé konkatenci řetězce do původního příkazu. Doctrine nám poskytuje metody, které ošetří nebezpečné znaky. Jsou to například metody `setParameter()`, `setMaxResults()` nebo `setFirstResult()`. Použití těchto metod je vidět ve výpisu 7.4 a ve výpisu 7.5 lze vidět příklad zranitelného kódu pomocí SQL injection, kdy jako parametr lze vložit `Ahoj;DROP TABLE survey;--` a po převedení do SQL dotazu by vznikly 2 dotazy, kdy druhý by smazal celou tabulku s průzkumy. V této implementaci jsou všechny argumenty vkládány přes zmíněné metody z knihovny Doctrine. Implementace ve všech repozitářích využívá ochrany proti SQL injection a všechny parametry jsou vkládány přes zmíněné funkce.

```
return $this->createQueryBuilder('s')
    ->select('s.id, s.title, s.subtitle)
    ->where('s.url = ' . $uri)
    ->getQuery()
    ->getResult();
```

Výpis 7.5: Zranitelný kód pomocí SQL injection

Oproti ER diagramu na obrázku 6.1 z kapitoly 6 jsou v systému implementovány ještě další pomocné entity. Jedná se o entity, které neměly zásadní vliv na návrh systému a proto nejsou v diagramech zobrazeny, jsou to entity pro potvrzení změny hesla, potvrzení

účtu uživatele a ověření unikátnosti respondenta. Celkový seznam entit v implementovaném portálu je následující:

- GeneralQuestion
- GeneralQuestionOption
- GeneralQuestionType
- GeoQuestion
- GeoQuestionMarker
- LostPasswordToken
- Opinion
- Question
- Respondent
- RespondentVerificationToken
- Response
- Survey
- Token
- User
- UserVerificationToken

Při převodu ER diagramu na schéma tabulek, v tomto případě do entitních tříd, ze kterých se vytvářejí tabulky, je potřeba správně provést převod generalizace do tabulek relační databáze. V tomto případě je pro obecné a geografické otázky zvlášť tabulka, nejsou v rámci jedné tabulky s rozlišením jejich typu pomocí sloupce výčtového typu. Toto řešení bylo zvoleno, protože obě specializace jsou disjunktní a totální, navíc je tak zvýšena přehlednost v aplikačním kódu, kdy integritní omezení se nemusejí implementovat v samotné aplikaci, ale nechávají se na správě řízení báze dat (SŘBD).

Entita `Question` je implementována jako abstraktní třída, která je označena pomocí anotace `@ORM\MappedSuperclass()`. Třídy `GeneralQuestion` a `GeoQuestion` využívají dědičnosti z této abstraktní třídy. Toto řešení umožňuje snížit množství duplicitního kódu a dále odstínit datovou vrstvu od samotných objektů. Knihovna `Doctrine` pak automaticky vygeneruje schéma tabulek tak, aby pro obě specializované entity byla samostatná tabulka.

Stejná situace je u verifikačních klíčů pro potvrzení změny hesla, e-mailu respondenta a potvrzení e-mailu uživatele. V tomto případě byla opět vytvořena abstraktní třída `Token`, která je opět označena výše uvedenou anotací a z ní dědí entity `LostPasswordToken`, `RespondentVerificationToken` a `UserVerificationToken`.

Průzkum, který je implementován entitou `Survey`, má proměnnou status, která nabývá pouze omezeného výčtu hodnot. Tento výčet nebyl modelován pomocí samostatné entity, ale pomocí seznamu povolených hodnot, který zastává funkci typu `enum`. Stejná situace je také u entity `Marker`, která má proměnnou type, která nabývá opět malého a omezeného počtu hodnot.

Typ enum, který podporuje databáze MySQL, není v Doctrine podporován, ale je implementován pomocí validace hodnot nad touto proměnnou pomocí callbacku na statickou funkci, která vrací seznam povolených hodnot. Samotná validace probíhá pomocí validátoru, který je součástí Symfony a byl krátce popsán v předchozí kapitole. Funkce pro omezení výčtu hodnot u zmíněných entit jsou implementovány ve třídách `MarkerTypeEnum` a `SurveyStatusEnum`, jedná se o abstraktní třídy, aby nemohly být instanciovány, protože obsahují pouze statické metody a jejich instanciací není v systému žádoucí. Validace těchto sloupců tak není zaručena přes integritní omezení a SŘBD, ale je zajišťována samotnou aplikací.

```
/**
 * @Assert\NotBlank
 * @Assert\Type("string")
 * @Assert\Choice(
 * callback={
 * "App\Entity\SurveyStatusEnum",
 * "getAvailableStatuses"
 * })
 * @ORM\Column(type="string", length=255)
 */
private $status;
```

Výpis 7.6: Ukázka mapování proměnné status

Protože dodatečné úpravy v otázkách průzkumu by mohly způsobit nekonzistenci ve vyplněných odpovědích od respondentů, je zakázáno změnit stav průzkum zpět na koncept, jestliže průzkum již má nějaké respondenty. Toto chování je řešeno pomocí reakce na události, které vyvolává Doctrine. A v případě, že se jedná o změnu lokality v průzkumu nebo změnu otázky v průzkumu, ve kterém jsou již respondenti, je tato akce přerušena pomocí vyvolání výjimky a navrácením chybového stavového kódu HTTP odpovědi.

Čím déle se bude systém používat, tím více průzkumů bude v databázi, ale pouze malá část z nich bude mít aktivní status. Proto je dobré zavést nad sloupcem status index, aby se zrychlilo vyhledávání aktivních průzkumů, což je typická činnost při vyplňování průzkumu respondentem, kdy API rozhraní hledá aktivní průzkum dle zadané URI adresy. Před zavedením indexu nad tímto sloupcem byl proveden EXPLAIN dotaz do databáze, který byl proveden nad SQL dotazem z funkce `findActiveSurveyByUri`. Ukázka této funkce je ve výpisu číslo 7.4. V době provádění dotazu bylo v databázi celkem 10 průzkumů a z nich pouze 3 v aktivním stavu.

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	survey	ALL	NULL	NULL	NULL	NULL	10	Using where

Tabulka 7.2: Výsledek EXPLAIN dotazu bez indexu

Z výsledku EXPLAIN dotazu v tabulce 7.2 je vidět, že došlo k prohledání všech záznamů v tabulce, což s narůstajícím počtem záznamů v budoucnu může způsobovat časem problém s rychlostí načítání informací o jednotlivých průzkumech. Z tabulky vyplývá, že typ ALL značí, že byla prohledána celá tabulka, tomu odpovídá i číslo 10 ve sloupci rows, které značí počet prohledaných záznamů. Následně byl v Doctrine nastaven index nad sloupcem status pomocí anotace, aktualizováno schéma databáze přes Doctrine a znovu spuštěn stejný EXPLAIN dotaz.

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	survey	ref	survey_intro	survey_intro	766	const	3	Using where

Tabulka 7.3: Výsledek EXPLAIN dotazu po zavedení indexu

Z tabulky 7.3 je vidět použití nastaveného indexu nad sloupcem status v tabulce s průzkumy, kdy došlo k omezení prohledávání tabulky pouze dle položek indexu, což nám signalizuje typ ref a jde vidět, že došlo k prohledání pouze 3 záznamů. Index nebyl zaveden nad více sloupci současně, protože ostatní sloupce budou mít zcela odlišné hodnoty v rámci tabulky a index by tak nebyl příliš užitečným, protože by musela být prohledávána téměř celá tabulka.

Další index, který byl nastaven ručně je nad sloupcem URL, který je unikátní. Ostatní indexy nad cizími a primárními klíči jsou generovány automaticky knihovnou Doctrine.

Protože ER diagram je pouze model pro abstraktní a konceptuální modelování dat v systému, je potřeba ho převést na samotné schéma tabulek relační databáze. Tento převod zajišťuje ORM knihovna Doctrine. Výsledné schéma tabulek, které vygeneruje Doctrine v MySQL databázi, je zobrazeno v příloze A.

### 7.3 Klientská část aplikace

Klientská část webové platformy, se kterou budou pracovat respondenti a úředníci Magistrátu města Brno je implementována v JavaScriptu pomocí frameworku Angular. Pro práci s mapou je využita knihovna ArcGIS JS API, která byla popsána v kapitole 4.5. Pro rychlejší vývoj a podporu responzivního vzhledu uživatelského rozhraní byl použit CSS framework Foundation<sup>3</sup>. K přenosu samotných dat ze serveru se využívá AJAX a XHR požadavky.

Do části s administrací průzkumů je potřeba se přihlásit, přihlašování probíhá pomocí e-mailu a uživatelem nastaveného hesla. Z přihlašovací obrazovky je možné přejít na stránku pro obnovu hesla. Po úspěšné autentizaci server v HTTP odpovědi zašle JSON Web Token, ze kterého se uloží do paměti prohlížeče zaslané údaje včetně role a také samotný token. Následně se zobrazí stránka se seznamem průzkumů.

Seznam průzkumů je zobrazen pomocí tabulky, ve které jsou informace o názvu a podnázvu průzkumu, datu začátku a konce, stavu a počtu respondentů průzkumu. Počet zobrazených průzkumů na jedné stránce je omezen na 10 průzkumů a na této stránce je implementováno stránkování, které se zobrazí, pokud je na první stránce alespoň 10 zobrazených průzkumů.

Do jednotlivých HTTP požadavků se přidává uložený token pomocí `HttpInterceptor`, aby se klient autentizoval s každým požadavkem a nebyl tak serverem odmítnut. Jestliže server v nějaké odpovědi zašle HTTP odpověď se stavovým kódem 401, tak dojde k automatickému odhlášení v klientské části a přesměrování na stránku s přihlášením.

Vytváření a úpravy průzkumu jsou zobrazeny na několika stránkách, kdy první stránka obsahuje obecné informace o samotném průzkumu, jako je název, počáteční a koncové datum průzkumu apod. Na další stránce se vybírá lokalita průzkumu, buď výběrem z nabídky městských částí v Brně nebo lze zvolit vlastní lokalitu v mapě. Na poslední stránce se pak přidávají samotné otázky průzkumu, které jsou rozděleny do dvou sekcí – obecné a geografické otázky.

<sup>3</sup>[foundation.zurb.com](http://foundation.zurb.com)

### Obecná otázka

Text otázky \*

Typ otázky \*

Vyžadována odpověď \*

Pořadí otázky \*

Seznam možností \*

Odpověď	Přidat
<input type="text" value="muž"/>	<input type="button" value="Smazat"/>
<input type="text" value="žena"/>	<input type="button" value="Smazat"/>

Položky označené \* jsou povinné

### Geografická otázka

Text otázky \*

Vyžadována odpověď

Textová zpětná vazba

Pořadí otázky \*

Seznam značek \*

Typ	Barva	Popis	Přidat
<input type="text" value="Linka"/>	<input type="text" value="Orange"/>	<input type="text" value="nejčastěji"/>	<input type="button" value="Smazat"/>

Položky označené \* jsou povinné

Obrázek 7.2: Dialogová okna s přidáním otázky

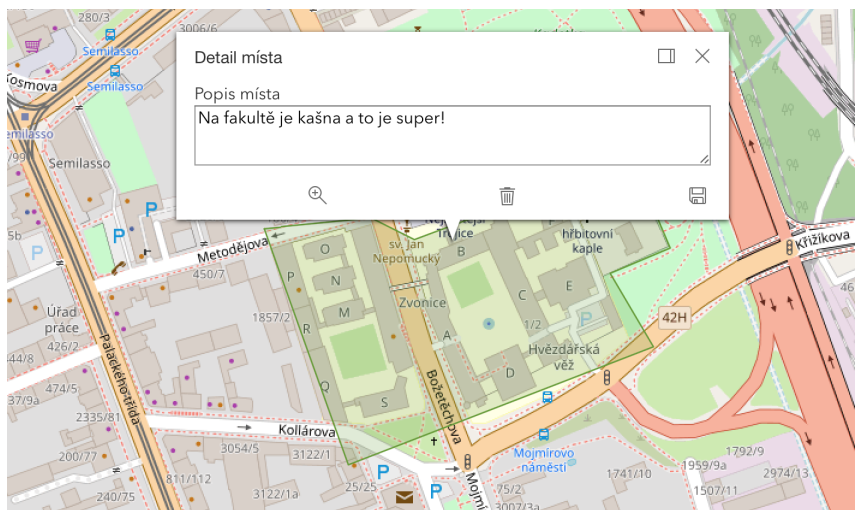
Jestliže je přihlášený uživatel administrátor, pak má k dispozici i stránku se seznamem uživatelů. Na této stránce jsou zobrazeni všichni uživatelé systému, včetně jejich rolí. Administrátor může konkrétního uživatele zablokovat nebo smazat. Přihlášený uživatel si může na stránce s profilem o sobě aktualizovat informace jako jméno, příjmení a e-mail, aby se však změna povedla, je potřeba vyplnit platné heslo. Pro změnu hesla je na stránce profilu odkaz na formulář s obnovením hesla.

Vyplnění průzkumu respondentem je rozděleno do několika částí. Nejprve je zobrazena úvodní stránka, kde se zobrazuje uvítací informace a případně ověření e-mailu v případě, že je nastaveno u daného průzkumu ověřování unikátnosti respondentů. Po ověření e-mailu se spustí průzkum a dojde k zobrazení stránky s obecnými otázkami. Po vyplnění odpovědí se zobrazí stránka s geografickými otázkami a mapou. Po vyplnění všech požadovaných otázek se zobrazí stránka s děkovným textem a s informací o uložení odpovědí na serveru, případně informací o chybě při ukládání a možnost znovu odeslat odpovědi.

Po načtení průzkumu s mapou je mapa automaticky přiblížena na zadanou lokalitu průzkumu tak, aby tato lokalita zabírala celou viditelnou část obrazovky a zároveň je zobrazen i boční panel s aktuální otázkou a seznamem značek této otázky, které respondent může použít k vyjádření názoru. Mapa je zobrazena pomocí knihovny ArcGIS JS API. Po přidání bodu dojde automaticky k zobrazení vyskakovacího okna, kde může respondent zadat komentář (textovou zpětnou vazbu nebo nápad) k zadanému místu, jestliže je to povoleno u dané otázky a případně zadaný objekt smazat.

Respondent má možnost si měnit podkladovou mapu, které jsou získávány ze serverů ČÚZK a ArcGIS Online platformy. Aplikace tak nabízí několik podkladových map a je implementována tak, aby bylo možné v budoucnu v případě potřeby další podkladové mapy přidávat. V současné době webová platforma poskytuje 6 podkladových map, mezi kterými lze přepínat. Všechny podkladové mapy používají stejný georeferenční systém, aby nedocházelo k problémům změny zobrazeného místa při přepnutí mapy, jedná se o Web Mercator. Na stránce s mapou je také ve spodním levém rohu zobrazen prvek pro vyhledávání ulic a míst.





Obrázek 7.3: Vyskakovací okno s povoleným textovým komentářem

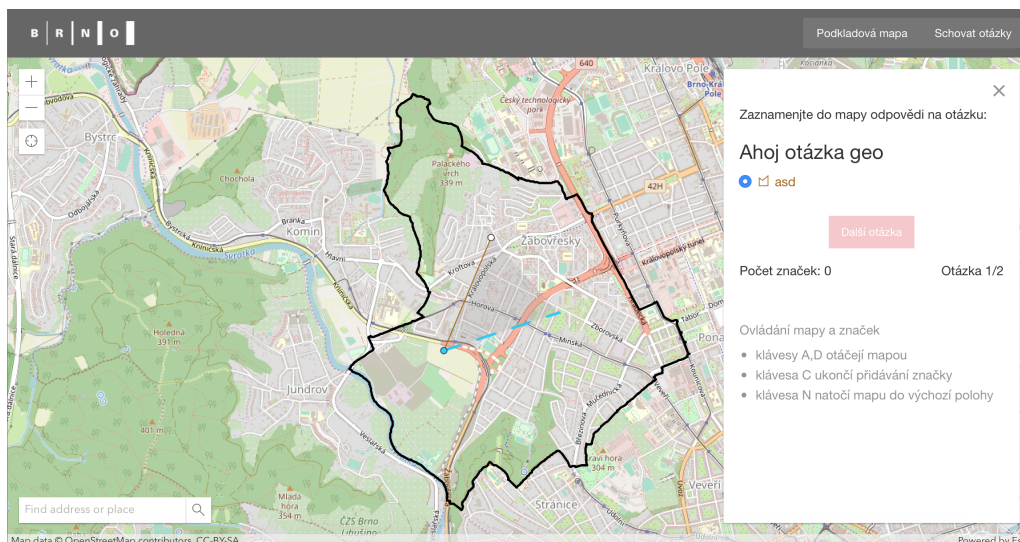
**FeautreLayer** je typ vrstvy v mapě, do které lze ukládat objekty. Tyto objekty musí být stejného geometrického typu (bod, polygon apod.) a sdílí jedno společné nastavení pro vykreslování. Jestliže chceme do mapy zaznamenávat více geometrických typů objektů, tak musíme pro každý typ vytvořit vlastní vrstvu. V tomto případě se tak jedná o vrstvu pro každou nastavenou značku k dané otázce. S narůstajícím počtem vrstev ovšem roste náročnost na výkon počítače, aby se pohyby v mapě zobrazovaly plynule, je potřeba mít malý počet vrstev.

**GraphicsLayer** je jiný typ vrstvy v mapě, do které se ukládají rovnou vytvořené grafiky, kdy každá grafika obsahuje i informace o geometrii. V aplikaci tak stačí mít pouze jednu vrstvu pro zaznamenávání odpovědí od respondentů, a proto je v téhle implementaci použita **GraphicsLayer** namísto **FeautreLayer** jako vrstva pro odpovědi.

Pro samotné zaznamenávání objektů je využita třída **SketchViewModel**, která usnadňuje proces přidávání a aktualizaci grafik v mapě. Pro správné fungování této třídy je potřeba mít k dispozici instanci třídy **GraphicsLayer**, do které tato třída ukládá vytvořené grafiky. Jakmile respondent vybere značku, kterou chce umístit do mapy, nastaví se **SketchViewModel** pro vytváření daného typu geometrie, kterou představuje značka a dle barvy značky se nastaví tato barva i do symbolu, pomocí kterého bude vytvořená grafika zobrazována. Místa lze zaznamenávat pouze do vyznačené lokality, jinak dojde k zobrazení varování při pokusu o přidání mimo lokalitu průzkumu a nedojde k přidání místa do mapy..

Body z grafické vrstvy v mapě je potřeba odeslat na server, kde budou uloženy v databázi. Na server jsou odeslány všechny odpovědi najednou po ukončení průzkumu. Zaznamenané grafiky jsou převedeny do GeoJSON reprezentace, ve které jsou uloženy v databázi. Pro převod z objektů, které používají proprietární formát ArcGIS do GeoJSON reprezentace, je využita knihovna **Terraformer ArcGIS Parser**, která ze zadané grafiky vytvoří GeoJSON objekt. Vytvořený objekt ovšem ještě nesplňuje aktuální standard, protože nemá souřadnice ve WGS84. Po získání validní GeoJSON reprezentace objektu je tak ještě objekt převeden pomocí **Terraformer Core** na objekt se souřadnicovým systémem WGS84.

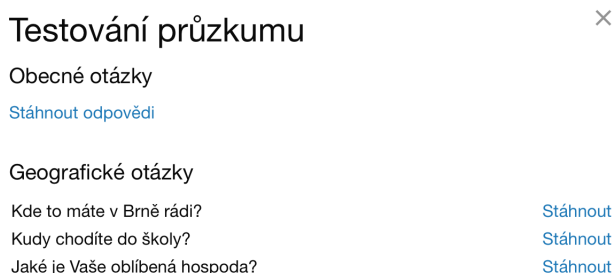
Jestliže se mají sesbírané odpovědi vizualizovat, je potřeba tato data načíst ze serveru. Na serveru však může být spousta odpovědí a proto jsou odpovědi načítány pouze pro aktuálně zobrazenou otázku. I když je takto omezen počet načítaných odpovědí, jsou ještě na



Obrázek 7.4: Ukázka stránky s mapou pro zaznamenání odpovědí

serveru omezeny maximálním počtem záznamů v SQL dotazu, který je načítá z databáze. Dochází tedy pouze k zobrazení části sesbíraných odpovědí a najednou tak může být zobrazeno maximálně 10000 zaznamenaných odpovědí. Načtené názory ze serveru jsou převedeny zpět do ArcGIS reprezentace. Při načtení takového množství grafik do mapy tak dochází k relativně krátkému zaseknutí ovládacích prvků uživatelského rozhraní.

Export zaznamenaných odpovědí probíhá po jednotlivých geografických otázkách, protože jinak by byl export sesbíraných dat příliš náročný na server a celý proces by tak trval dlouhou dobu. Toto řešení navíc umožňuje jednotlivým úředníkům si stáhnout pouze ty otázky, o které mají zájem a nemusí stahovat úplně všechny odpovědi daného průzkumu. Export geografických odpovědí je ve formátu GeoJSON a odpovědi na obecné otázky jsou exportovány ve formátu CSV. Export obecných odpovědí probíhá po určitém počtu respondentů, kdy každý řádek v CSV souboru představuje jednoho respondenta a jeho odpovědi na otázky, které jsou zobrazeny v prvním řádku souboru.



Obrázek 7.5: Ukázka modálního okna s exportem sesbíraných názorů k průzkumu

Webová platforma umožňuje pomocí online mapy sbírat názory obyvatel k jednotlivým místům, dle definovaných značek lze pak jednoznačně identifikovat, jaký mají k zaznačenému místu či lokalitě obyvatelé vztah. V nastavení jednotlivých otázek pak lze povolit sběr zpětné vazby a respondenti tedy mohou k zaznačenému místu přidat textovou zpětnou vazbu.

# Kapitola 8

## Testování

V této kapitole je popsán způsob testování a možnost dalšího rozvoje práce. Nejdříve je popsáno, jak probíhalo testování ve spolupráci s Magistrátem města Brna, který je následován popisem testování na respondentech. Dále je popsán export velkého množství sesbíraných dat a v poslední části je zmíněno několik možností pokračování této práce.

### 8.1 Testování na úřednících magistrátu

V průběhu implementace byl aktuální stav konzultován na Magistrátu města Brna a připomínky byly průběžně zapracovávány. Testování části, se kterou budou nejčastěji pracovat zaměstnanci Magistrátu města Brna proběhlo na hotové implementaci přímo ve spolupráci s úředníky. Část úředníků si vyzkoušela vytvořit celý průzkum, včetně různých typů otázek a následně i export sesbíraných dat, která byla u jiných průzkumů vygenerována předem. Testování neodhalilo žádné zásadní problémy. Došlo k odhalení primárně chyb v textových řetězcích a zobrazení výsledného odkazu pro respondenty dle aktuálního stavu průzkumu. Testování probíhalo pomocí různých prohlížečů, kdy každý úředník používal prohlížeč dle svých preferencí.

Odkaz na adresu průzkumu pro respondenty byl zobrazen bez varování i když průzkum nebyl v aktivním stavu, úředníci se ho pokoušeli vyzkoušet a vnímali jako chybu, že se průzkum nespustil. Uživatelské rozhraní tak bylo předěláno, aby se odkaz zobrazoval pouze tehdy, pokud je průzkum v aktivním stavu. Dále bylo implementováno varování, které úředníky upozorní, že odkaz bude funkční jenom v době trvání průzkumu. Jakmile se nastaví průzkum do ukončeného stavu, dojde k zobrazení odkazu na základní vizualizaci sesbíraných dat z průzkumu.

Testování potvrdilo, že požadované vlastnosti z kapitoly 5, které na tento systém klade Magistrát města Brna jsou dosaženy a úředníci jsou s daným systémem spokojeni. Systém je tak v plánu v nejbližších době nasadit do ostrého provozu, aby sloužil svému účelu, kvůli kterému byl vytvořen.

### 8.2 Testování na respondentech

Testování části se zaznamenáváním názorů obyvatel probíhalo v průběhu implementace, kdy webová platforma byla průběžně a opakovaně ukazována lidem různého věku a pohlaví. V závislosti na jejich zpětné vazbě byla aplikace průběžně upravována, aby byla pro ně snadněji použitelná a intuitivnější.

Toto testování probíhalo zpravidla neformálně, kdy současný stav implementace byl ukazován lidem v okolí, aby si daný člověk mohl s aplikací pohrát. Respondent již během samotného hraní si s implementací většinou sám začal říkat informace, co mu přijde otravné, kterou funkcionalitu by zlepšil apod. Tyto připomínky a náměty tak byly postupně zapracovávány během samotného vývoje portálu.

Následně byla hotová implementace otestována na respondentech, kdy respondent vyplňoval odpovědi k otázkám průzkumu a byl při tom pozorován. Toto testování však již neodhalilo žádné zásadní nedostatky a respondenti hodnotili výslednou implementaci jako celkem intuitivní a neměli problém s vyplňováním průzkumu. Geografické otázky byly například „Kudy chodíte do školy/práce?“ nebo „Kde to máte v Brně rádi?“.

### 8.3 Testování exportu velkého množství odpovědí

Nejdříve byl implementován hromadný export všech sesbíraných odpovědí a následně byla do databáze nahrána testovací data. Tato data obsahovala odpovědi 60 000 respondentů na 30 obecných otázek a 8 geografických otázek. Odpovědi na obecné otázky a seznam respondentů byly použity z datového portálu města Brna a odpovědi na geografické otázky byly vygenerovány, kdy každý respondent odpověděl na každou otázku několika značkami do mapy.

Následně byla předělána implementace, aby šlo exportovat takové množství dat, kdy tato výsledná implementace je již popsána v rámci předchozích kapitol. Změny se týkaly jak uživatelského rozhraní, tak serverové části. Export byl rozdělen na menší části, aby se snížila paměťová náročnost a export menších částí tak trvá i kratší dobu. Aby se neexportovaly hromadně všechny sesbírané odpovědi k danému průzkumu, kterých může být velké množství, tak se exportují odpovědi na obecné otázky po určitém počtu respondentů a odpovědi na geografické otázky po jednotlivých otázkách. Kdy v rámci administrace si uživatel vybírá, kterou geografickou otázku si chce stáhnout.

### 8.4 Možnosti dalšího rozšíření

Webový portál je implementován tak, aby bylo možné ho snadno rozšířit o další funkce v budoucnu, kdy nejdříve je v plánu implementace rozšíření pro nahrávání fotografií k jednotlivým zaznačeným bodům v mapě.

Nahrávání fotografií přinese další možnost bližší specifikace zaznačeného místa, kdy například bude možné nahrávat fotografie přeplněných popelnic apod., a tak ještě více zpřesnit zpětnou vazbu získanou od obyvatel.

Dále lze systém rozšířit o podporu změny jazyka rozhraní, aby systém umožňoval i občanům jiné národnosti, kteří neovládají český jazyk, tento systém používat a magistrát tak mohl sbírat více názorů od obyvatel.

Systém by také v budoucnu mohl být rozšířen o registraci respondentů, kdy by si registrovaní respondenti pak mohli zobrazovat své odpovědi, které vyplnili k jednotlivým průzkumům a případně tyto odpovědi i upravovat, pokud je průzkum stále aktivní.

## Kapitola 9

# Závěr

Cílem této bakalářské práce bylo experimentálně implementovat webový mapový portál pro tvorbu průzkumů, který má umožnit Magistrátu města Brna vytvářet a definovat vlastní průzkumy a sbírat názory obyvatel města. Nejprve bylo potřeba se seznámit s problematikou sběru názorů obyvatel a vybrat vhodnou metodu tohoto sběru, aby se sběr dal realizovat v online prostředí. Seznámil jsem se s různými metodami sběru názorů, kdy se v prostředí internetu nejčastěji používá sbírání názorů pomocí pocitových map.

Následně jsem se seznámil s možnostmi existujících řešení, a to včetně těch, které jsou používány v zahraničí. Ve spolupráci s úředníky z Oddělení dat, analýz a evaluací na Magistrátu města Brna byly specifikovány požadavky na tento portál, které byly zaznačeny do diagramu případů užití. Po samotné analýze požadavků byl vytvořen návrh systému, který obsahuje ER diagram a popis výběru technologií pro implementaci této platformy.

Dle vytvořeného návrhu byl webový portál implementován a během implementace byl průběžně testován a konzultován s Magistrátem města Brna. Systém je implementován jako dvě části, serverová část a klientská část. Klientská část je postavena jako Single page application, která komunikuje se serverovou částí, která je naprogramována v jazyce PHP. Implementovaný systém následně prošel testováním na respondentech různého věku a pohlaví, dále úředníky Magistrátu města Brna a byl testován i export velkého množství sesbíraných názorů od obyvatel. Úředníci byly s funkčností systému velmi spokojeni a plánují ho v nejbližší době nasadit do ostrého provozu a sbírat pomocí něj názory od obyvatel.

Výsledná aplikace tedy splňuje zadání této bakalářské práce, kdy je možné vytvářet vlastní otázky, jak obecné, tak i geografické. Tyto otázky jsou součástí průzkumu, který probíhá jenom v požadované lokalitě a v zadaném období. Po ukončení průzkumu je možné si zobrazit jednoduchou vizualizaci sesbíraných dat a samotná data exportovat ve formátu vhodném pro další zpracování.

V blízké době je v plánu implementovat rozšíření geografických otázek, aby se k jednotlivým odpovědím v mapě mohli kromě textového komentáře nahrávat i fotografie. Dále lze systém rozšířit o podporu více jazyků, aby ho mohli používat i obyvatelé Brna, kteří neovládají příliš dobře český jazyk.

# Literatura

- [1] *Features* — *Maptionnaire*. Maptionnaire, [Online; navštíveno 12.10.2018].  
URL [maptionnaire.com/features](http://maptionnaire.com/features)
- [2] *Laravel - The PHP Framework For Web Artisans*. [Online; navštíveno 10.1.2019].  
URL <http://laravel.com>
- [3] *The technological benefits of Symfony in 6 easy lessons*. [Online; navštíveno 12.1.2019].  
URL <http://symfony.com/six-good-technical-reasons>
- [4] *Web Map Tiled Service*. [Online; navštíveno 6.3.2019].  
URL <http://desktop.arcgis.com/en/arcmap/latest/map/working-with-arcmap/fundamentals-for-creating-tile-packages.htm>
- [5] *Metody mapování veřejných prostranství a území*. Centrum občanského vzdělávání, 2012, [Online; navštíveno 15.12.2018].  
URL [www.participativnimetody.cz/metody-mapovani-verejnych-prostranstvi-a-uzemi.html](http://www.participativnimetody.cz/metody-mapovani-verejnych-prostranstvi-a-uzemi.html)
- [6] *Brno - Pocitová mapa (2016)*. Magistrát města Brna, 2016, [Online; navštíveno 13.12.2018].  
URL [www.brno.cz/pocitova-mapa-2016](http://www.brno.cz/pocitova-mapa-2016)
- [7] *API Mapy.cz*. Seznam.cz, 2018, [Online; navštíveno 15.12.2018].  
URL [api.mapy.cz](http://api.mapy.cz)
- [8] *Develop - OpenStreetMap Wiki*. OpenStreetMap, 2018, [Online; navštíveno 15.12.2018].  
URL [wiki.openstreetmap.org/wiki/Develop](http://wiki.openstreetmap.org/wiki/Develop)
- [9] *Overview | Maps JavaScript API | Google Developers*. Google, 2018, [Online; navštíveno 14.12.2018].  
URL [developers.google.com/maps/documentation/javascript/tutorial](http://developers.google.com/maps/documentation/javascript/tutorial)
- [10] *Výsledky pocitové mapy Brna a jeho čtvrtí*. Magistrát města Brna, 2018, [Online; navštíveno 13.12.2018].  
URL [brno2050.cz/vysledky-pocitove-mapy](http://brno2050.cz/vysledky-pocitove-mapy)
- [11] *Angular - What is Angular?* Google, c2010-2018, [Online; navštíveno 10.1.2019].  
URL [angular.io/docs](http://angular.io/docs)

- [12] *Interactive Mapping for Engagement - Social Pinpoint - A Place to Engage Your Community*. Social Pinpoint, c2014-17, [Online; navštíveno 11.10.2018].  
URL [www.socialpinpoint.com/product-interactive-mapping-engagement](http://www.socialpinpoint.com/product-interactive-mapping-engagement)
- [13] Banks, A.; Porcello, E.: *Learning React*. Sebastopol, CA: O'Reilly Media, 2017, ISBN 978-1-491-95462-1.
- [14] Beaujardiere, J.: *OpenGIS® Web Map Server Implementation Specification*. USA: Open Geospatial Consortium, 2006, [Online; navštíveno 14.12.2018].  
URL [www.opengeospatial.org/standards/wms](http://www.opengeospatial.org/standards/wms)
- [15] Bordons, C. C.: *Web Map Tile Services Tiny Tile Server*. Bakalářská práce, University of Applied Science Rapperswil, Rapperswil-Jona, Švýcarsko, 2013, [Online; navštíveno 10.3.2019].  
URL <https://eprints.hsr.ch/323/1/TinyTileServer.pdf>
- [16] Butler, H.; Daly, M.; Doyle, A.; aj.: *The GeoJSON Format*. RFC 7946, August 2016, [Online; navštíveno 10.11.2018].  
URL <http://www.rfc-editor.org/rfc/rfc7946.txt>
- [17] Craig, W. J.; Harris, T. M.; Weiner, D.: *Community Participation and Geographical Information Systems*. London: CRC Press, 2002, ISBN 9780429203961, [Online; navštíveno 15.12.2018].  
URL <https://www.taylorfrancis.com/books/e/9780429203961>
- [18] Darie, C.: *AJAX a PHP: tvoříme interaktivní webové aplikace profesionálně*. Brno: Zoner Press, 2006, ISBN 80-868-1547-1.
- [19] Jones, M.; Bradley, J.; Sakimura, N.: *JSON Web Token (JWT)*. RFC 7519, May 2015, [Online; navštíveno 16.2.2019].  
URL <http://www.rfc-editor.org/rfc/rfc7519.txt>
- [20] Kesteren, A.: *Cross-Origin Resource Sharing*. W3C recommendation, W3C, Leden 2014, [Online; navštíveno 18.2.2019].  
URL <http://www.w3.org/TR/2014/REC-cors-20140116>
- [21] Masó, J.; Pomakis, K.; Julià, N.: *OpenGIS® Web Map Tile Service Implementation Standard*. USA: Open Geospatial Consortium, 2010, [Online; navštíveno 15.12.2018].  
URL <http://www.opengeospatial.org/standards/wmts>
- [22] Mikowski, M. S.; Powell, J. C.: *Single page web applications*. Shelter Island, NY: Manning, 2014, ISBN 978-161-7290-756.
- [23] Pánek, J.: *GeoParticipace*. Olomouc: Univerzita Palackého v Olomouci, první vydání, 2014, ISBN 978-80-244-4359-1, [Online; navštíveno 10.12.2018].  
URL [www.development.upol.cz/wp-content/uploads/2014/08/GeoParticipace.pdf](http://www.development.upol.cz/wp-content/uploads/2014/08/GeoParticipace.pdf)
- [24] Pánek, J.: *Pocitové mapy*. ©2015-2016, [Online; navštíveno 1.10.2018].  
URL [www.pocitovemapy.cz](http://www.pocitovemapy.cz)





# Příloha A

## Schéma tabulek v databázi

