



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**ZVÝŠENÍ POUŽITELNOSTI KNIHOVNY GEOVISTO**

USABILITY IMPROVEMENT OF GEOVISTO LIBRARY

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**TOMÁŠ KOSCIELNIAK**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. JIŘÍ HYNEK, Ph.D.**

BRNO 2022

## Zadání bakalářské práce



Student: **Koscielniak Tomáš**  
Program: Informační technologie  
Název: **Zvýšení použitelnosti knihovny Geovisto**  
**Usability Improvement of Geovisto Library**  
Kategorie: Uživatelská rozhraní

### Zadání:

1. Prostudujte problematiku zpracování a vizualizace geografických dat na webu. Prozkoumejte existující technologie určené pro tento účel (zaměřte se na uživatelské rozhraní a způsoby jejich ovládání).
2. Prostudujte problematiku použitelnosti uživatelských rozhraní a uživatelské zkušenosti (UX). Prozkoumejte existující knihovny určené pro uživatelsky přívětivou demonstraci použití webových grafických komponent (např. Storybook).
3. Analyzujte použitelnost uživatelského rozhraní nástrojů knihovny Geovisto a detekujte problémy jeho použitelnosti (např. nejasnosti s konfigurací tématických map, obtížnost čtení hodnot geografických objektů, apod.). Pro tento účel připravte vhodnou databázi geografických dat a sadu pohledů demonstrující případy užití pro následné uživatelské testování.
4. Navrhněte vhodné úpravy nástrojů knihovny Geovisto s ohledem na výsledky analýzy z bodu 3 a pokyny vedoucího práce. Navrhněte webovou prezentaci demonstrující možnosti nástroje knihovny Geovisto včetně srozumitelné dokumentace.
5. Implementujte navržené úpravy v nástrojích knihovny Geovisto, proveďte aktualizaci balíků v repository nrm a vytvořte webovou prezentaci poskytující dokumentaci a demonstraci použití těchto nástrojů.
6. Proveďte výsledné testování a zhodnoťte zlepšení použitelnosti nástroje Geovisto.

### Literatura:

- Hynek, J., Kachlík, J. a Rusňák, V.: *Geovisto: A Toolkit for Generic Geospatial Data Visualization*. In *VISIGRAPP (3: IVAPP)* (pp. 101-111).
- Dent, B., D., a spol.: *Cartography: Thematic Map Design*. McGraw-Hill Higher Education 2009, ISBN 978-128-3388-023.
- Johnson, J.: *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Morgan Kaufmann Publishers/Elsevier, 2010, ISBN: 9780123750303.
- Preece, J.: *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, 2015, ISBN: 978-1-119-02075-2.
- Leaflet: *Leaflet API reference* [online]. 2021 [cit. 2021-10-09]. Dostupné z: <https://leafletjs.com/reference-1.7.1.html>

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hynek Jiří, Ing., Ph.D.**  
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.  
Datum zadání: 1. listopadu 2021  
Datum odevzdání: 11. května 2022  
Datum schválení: 18. října 2021

## Abstrakt

Tato práce se věnuje vylepšení uživatelského rozhraní knihovny Geovisto pro vizualizaci geografických dat. Toho je dosaženo identifikováním problémů pomocí uživatelského testování a návrhu jejich řešení. Tato řešení byly implementovány a jejich přínos byl ověřen druhým uživatelským testováním. Výsledkem práce je rozšíření knihovny o legendy, jednotky u číselných hodnot a nástroj pro vizualizaci metadat, která zlepšují uživatelskou zkušenost při práci s knihovnou. Dalším výsledkem je také samostatná webová prezentace knihovny, jejích nástrojů a data miner.

## Abstract

This thesis aims to improve the user interface of Geovisto library, used for geospatial visualization. This is achieved by identifying issues using user testing and designing a solution for them. These solutions were implemented, and the second user testing tested their benefit. The result of the thesis is expanding the library with legends, units for numerical values, and tools for visualization of metadata, which all improve the user experience when using the library. Another result is a standalone web presentation of the library, its tools, and data miner.

## Klíčová slova

tematická mapa, geovizualizace, uživatelské rozhraní, uživatelská zkušenost, webová dokumentace, legenda, metadata

## Keywords

thematic map, geovisualization, user interface, user experience, web documentation, legend, metadata

## Citace

KOSCIELNIAK, Tomáš. *Zvýšení použitelnosti knihovny Geovisto*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hynek, Ph.D.

# Zvýšení použitelnosti knihovny Geovisto

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Hynka, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....  
Tomáš Koscielniak  
9. května 2022

## Poděkování

Rád bych poděkoval svému vedoucímu Ing. Jiřímu Hynkovi, Ph.D. za vstřícný přístup při konzultacích, čas věnovaný mojí práci, poskytnutí studijní literatury, odborné vedení a celkově skvělý přístup k vedení práce. Dále bych rád poděkoval všem, kteří se zúčastnili uživatelského testování.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Zpracování a vizualizace geografických dat na webu</b>	<b>5</b>
2.1	Tematické mapy . . . . .	5
2.1.1	Kvalitativní a kvantitativní . . . . .	5
2.1.2	Kartogram . . . . .	6
2.1.3	Teplotní mapa . . . . .	6
2.2	Data . . . . .	7
2.2.1	Geografická data . . . . .	7
2.2.2	GeoJSON . . . . .	8
2.2.3	Tematická data . . . . .	8
2.2.4	Metadata . . . . .	9
2.3	Vizualizace na webu . . . . .	9
<b>3</b>	<b>Použitelnost uživatelských rozhraní</b>	<b>11</b>
3.1	Uživatelské rozhraní . . . . .	11
3.2	Pravidla pro návrh uživatelského rozhraní . . . . .	12
3.2.1	Konzistence . . . . .	12
3.2.2	Struktura rozhraní . . . . .	13
3.2.3	Text . . . . .	13
3.2.4	Používání barev . . . . .	15
3.2.5	Rychlost a responzivní design . . . . .	15
<b>4</b>	<b>Existující řešení</b>	<b>17</b>
4.1	Technologie pro webovou vizualizaci . . . . .	17
4.2	Autorské systémy . . . . .	18
4.3	Programovací knihovny . . . . .	18
4.4	Knihovna Geovisto . . . . .	19
4.4.1	Uživatelské rozhraní . . . . .	19
4.4.2	Datový model . . . . .	21
4.4.3	Architektura . . . . .	21
<b>5</b>	<b>Analýza</b>	<b>23</b>
5.1	Cílová skupina . . . . .	23
5.2	Testovací sada . . . . .	23
5.2.1	Vytváření případů užití . . . . .	23
5.2.2	Získání tematických dat . . . . .	24
5.2.3	Výsledná testovací sada . . . . .	25

5.3	Uživatelské testování . . . . .	26
5.3.1	Použité metriky . . . . .	26
5.3.2	Identifikované problémy uživatelského rozhraní . . . . .	26
<b>6</b>	<b>Návrh řešení</b>	<b>28</b>
6.1	Legenda . . . . .	28
6.2	Metadata . . . . .	28
6.3	Jednotky a zaokrouhlování hodnot . . . . .	29
6.4	Dokumentace . . . . .	29
<b>7</b>	<b>Implementace</b>	<b>32</b>
7.1	Geovisto Data Miner . . . . .	32
7.2	Rozšíření knihovny Geovisto . . . . .	32
7.2.1	Implementace legend . . . . .	33
7.2.2	Implementace vizualizace metadat . . . . .	34
7.2.3	Implementace jednotek a zaokrouhlování hodnot . . . . .	35
7.3	Webová dokumentace . . . . .	36
<b>8</b>	<b>Testování</b>	<b>39</b>
8.1	Druhé uživatelské testování . . . . .	39
8.2	Výsledek . . . . .	39
<b>9</b>	<b>Závěr</b>	<b>41</b>
	<b>Literatura</b>	<b>42</b>

# Kapitola 1

## Úvod

Geografická data jsou informace, kterými popisujeme vlastnosti nebo soubor vlastností vztahujících se ke konkrétnímu místu na Zemi. Tato data vizualizujeme různými formami a metodami, jako jsou například kartogramy, tečkové a nebo teplotní mapy. K těm také připojujeme různá metadata, která nám poskytují dodatečné informace o samotných geografických datech, případně jejich zdrojích a další údaje.

Pro samotnou vizualizaci dat využíváme různé technologie, například klasické fyzické mapy. Nás ale v kontextu této práce zajímají webové vizualizace geografických dat. Jedná se o uživatelsky přívětivý způsob umožňující zobrazení více vrstev, metadat nebo nastavení úrovně detailu a měřítka. Pod tímto pojmem si můžeme představit běžně používané aplikace Google Maps nebo Mapy.cz. Tyto a jim podobné nástroje poskytují snadný přístup k informacím používaným v komerční a akademické sféře, ale i běžnými uživateli. Existující řešení pro tvorbu vizualizací geografických dat jsou ovšem často zpoplatněná, příliš komplikovaná, s nedostatkem funkcí nebo vyžadují pokročilé programátorské dovednosti.

Ve snaze o zjednodušení tvorby tematických map byla vyvinuta aplikace Geovisto, která poskytuje možnost upravovat vizualizovaná data programově, ale i přímo v uživatelském rozhraní. Lze ji také rozšiřovat pomocí externích modulů díky veřejně dostupnému a rozšiřitelnému zdrojovému kódu. Její základ tvoří samotná mapa implementovaná pomocí rozhraní Leaflet API, na ní navázané nástroje aplikace Geovisto, případně jejich data ve formátu JSON a geografická data ve formátu GeoJSON. Implementována je pomocí programovacího jazyka TypeScript a knihoven Leaflet, D3.js a React. Jednotlivé nástroje lze na sebe vrstvit, zapnout či vypnout a rozsáhle konfigurovat jejich vlastnosti nebo vzhled. Tuto konfiguraci lze také exportovat a importovat.

Cílem této práce je prostudovat problematiku vizualizace geografických dat na webu a použitelnosti uživatelských rozhraní, provést analýzu knihovny Geovisto z tohoto hlediska a na jejím základě navrhnout vhodné úpravy knihovny, které by řešily problémy z ní vyplývající. Po vyhodnocení celkového přínosu možných změn pro zvýšení použitelnosti budou vybrány ty nejrelevantnější úpravy a následně implementovány. Výsledkem práce bude rozšíření knihovny Geovisto o moduly zajišťující implementaci realizovaných vylepšení a webová dokumentace, která je mířena jak na běžné uživatele, tak i vývojáře, kteří mají zájem knihovnu upravovat či rozšiřovat. Bude také obsahovat obecné informace o knihovně a sloužit jako nástroj pro její základní prezentaci a propagaci.

V kapitole 2 a 3 se budu zabývat hlubším zpracováním problematiky vizualizace geografických dat a použitelnosti uživatelských rozhraní obecně. V kapitole 4 zhodnotím aktuální stav knihovny Geovisto a jiných řešení z pohledu témat předchozích dvou kapitol. V 5. kapitole potom provedu analýzu použitelnosti uživatelského rozhraní přímo knihovny Geo-

visto. Tato kapitola bude též obsahovat popis provedení uživatelského testování, které bude sloužit jako hlavní zdroj objektivních informací pro identifikování problémů již zmíněnou analýzou a také jako nástroj pro zhodnocení dosažených výsledků celé práce. Kapitoly 6 a 7 se budou zabývat návrhem řešení a implementací výstupů analýzy probíraných v kapitole 5. Kapitola 8 pokryje testování výsledné implementace, které bude provedeno hlavně pomocí druhého uživatelského testování, jehož výsledky budou porovnány s prvním a z tohoto srovnání bude vyvozen celkový závěr. Ten se bude nacházet v poslední, 9. kapitole a zhodnotí výstupy práce a její přínos.



## Kapitola 2

# Zpracování a vizualizace geografických dat na webu

V této kapitole budou probrána samotná geografická data a jejich digitální formát, problematika vizualizace těchto dat ve webové podobě a také momentálně dostupná řešení vizualizace a kvalita těchto nástrojů z pohledu přívětivosti k uživateli. Než budou představena zmíněná témata, je třeba si prvně stručně rozvést pojem *tematické mapy*.

### 2.1 Tematické mapy

Podle [3] můžeme mapy rozdělit na dva hlavní typy. První typ map se nazývá *všeobecně zeměpisné* a jedná se o takové mapy, které vizualizují objekty z geografického prostředí. Zaměřují se na vlastnosti dané lokace, s vhodnou úrovní detailu podle měřítka mapy, a běžné příklady map tohoto typu jsou mapy topografické nebo mapy států. Můžeme na nich nalézt grafické znázornění nadmořské výšky, vyznačení cest, hranic správních celků, komunikací nebo měst. V kontextu této práce ale potřebujeme znát hlavně druhý typ map, nazývaný právě *tematické mapy*.

Tento druh map slouží specifickému, konkrétnímu účelu. Jejich častým použitím je zobrazování statistických dat, například populace, hustota zalidnění nebo průměrný příjem. Obecně se jedná o vztah mezi místem a k němu relevantními vlastnostmi. V následujících sekcích budou rozvedena základní dělení tematických map a uvedeny jejich příklady.

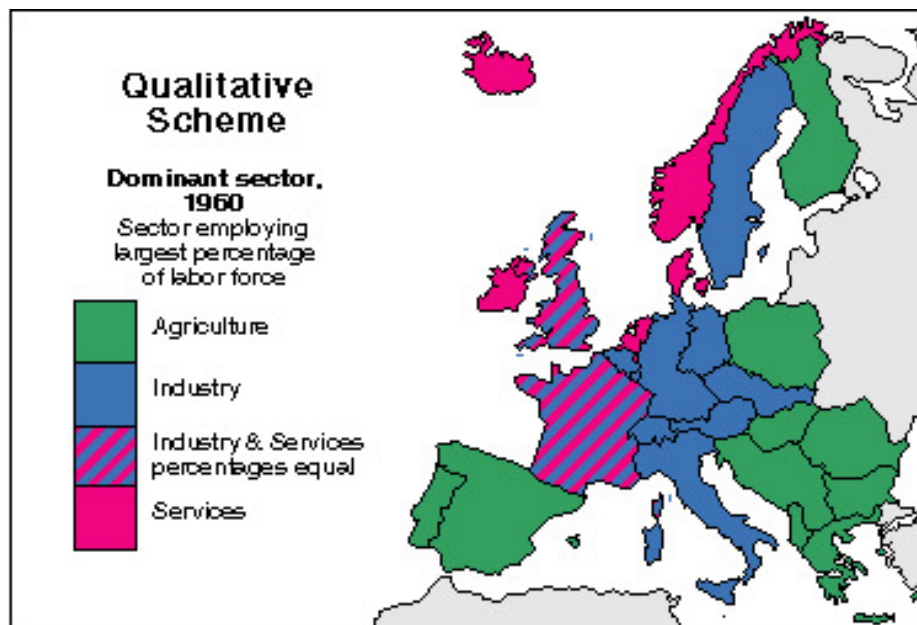
#### 2.1.1 Kvalitativní a kvantitativní

Jako kvalitativní tematické mapy označujeme takové, které nezobrazují vyčíslitelné údaje, ale spíše prostou přítomnost vlastnosti pro danou lokaci. Příklad takové mapy může být výskyt nerostných surovin nebo dominantní sektor v procentuálním počtu zaměstnanců z celkového objemu (obrázek 2.1).

V kontextu této práce jsou ale opět relevantnější mapy kvantitativní. Jedná se o mapy, které zobrazují prostorovou reprezentaci číselných dat. Pod tímto pojmem si můžeme představit konkrétní číselná data ovlivňující vlastnost lokace na mapě, například již dříve zmínovanou hustotu zalidnění v daném místě. Pro tento typ map se používají různé formy vizualizace. Jedny z nejčastěji využívaných podob jsou *kartogramy*, *teplotní mapy* nebo mapy vytvořené *tečkovou metodou*.

---

<sup>2</sup>Zdroj: [https://web.natur.cuni.cz/~langhamr/lectures/vtfg1/mapinfo\\_2/barvy/colors.html](https://web.natur.cuni.cz/~langhamr/lectures/vtfg1/mapinfo_2/barvy/colors.html)



Obrázek 2.1: Příklad kvalitativní mapy zobrazující ekonomický sektor zaměstnávající nejvíc pracovníků v jednotlivých zemích Evropy<sup>2</sup>

### 2.1.2 Kartogram

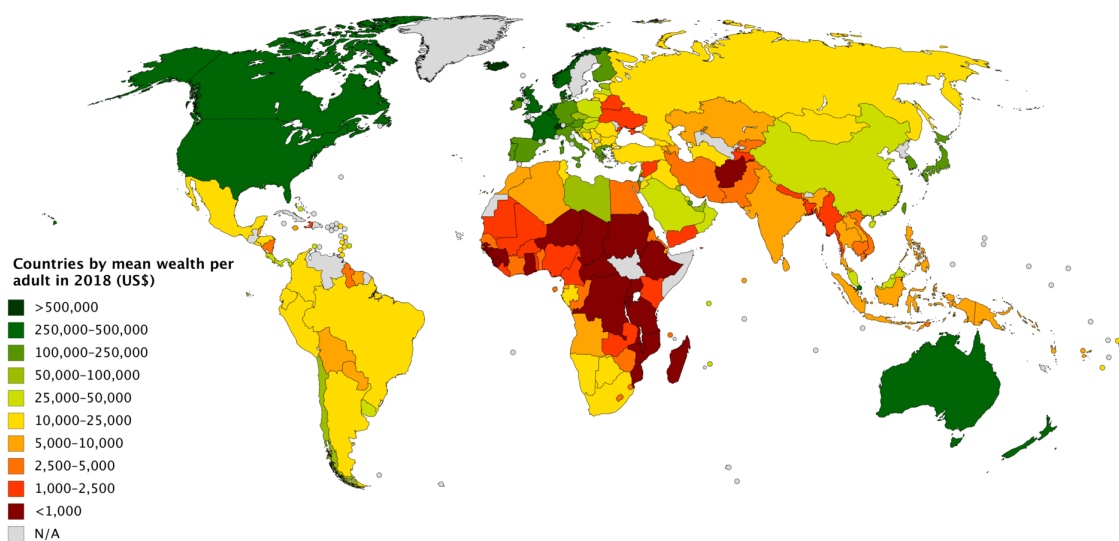
Jedná se jednoduchý typ kvantitativní tematické mapy, který funguje na základě přiřazování různých barevných intenzit lokacím na mapě. Intenzita je určena podle číselné hodnoty vlastnosti této lokace porovnané vůči hodnotám stejného typu ostatních lokací na mapě. Často jsou pro tyto hodnoty zdefinované číselné intervaly, podle kterých je lokacím barevná intenzita přidělena (obrázek 2.2). Používají se intenzity jedné barvy, či spektrum více barev a jejich intenzit.

### 2.1.3 Teplotní mapa

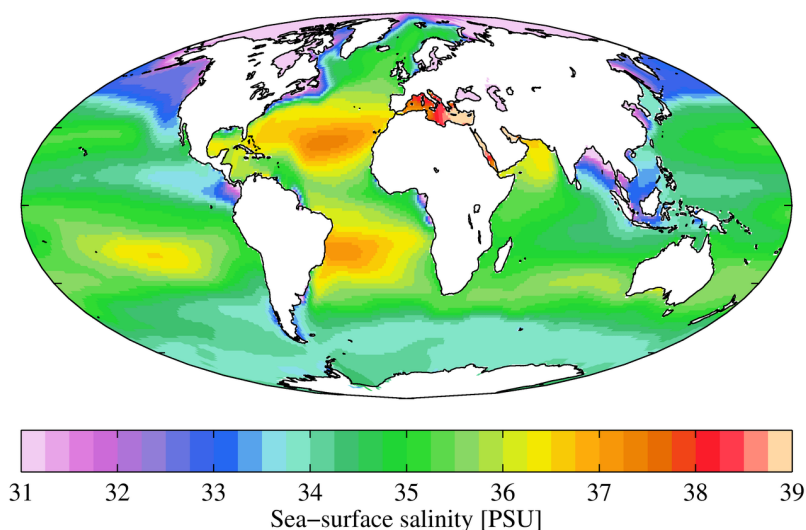
Teplotní mapa je běžně používaný kvantitativní typ tematických map, který zabarvuje lokaci do určité barvy. Ta je v klasické tepelné mapě určena podle velikosti hodnoty. Čím vyšší hodnota, tím teplejší barva a čím nižší hodnota, tím studenější barva (obrázek 2.3). Používají se třeba v meteorologii.

<sup>4</sup>Zdroj: [https://commons.wikimedia.org/wiki/File:Countries\\_by\\_mean\\_wealth\\_per\\_adult\\_in\\_2018.png#/media/File:Countries\\_by\\_mean\\_wealth\\_per\\_adult\\_in\\_2018.png](https://commons.wikimedia.org/wiki/File:Countries_by_mean_wealth_per_adult_in_2018.png#/media/File:Countries_by_mean_wealth_per_adult_in_2018.png)

<sup>5</sup>Zdroj: [https://commons.wikimedia.org/wiki/File:WOA09\\_sea-surf\\_SAL\\_AYool.png#/media/Soubor:WOA09\\_sea-surf\\_SAL\\_AYool.png](https://commons.wikimedia.org/wiki/File:WOA09_sea-surf_SAL_AYool.png#/media/Soubor:WOA09_sea-surf_SAL_AYool.png)



Obrázek 2.2: Příklad kartogramu popisující mediánový majetek s vícebarevným zvýrazněným lokalit<sup>4</sup>



Obrázek 2.3: Příklad teplotní mapy popisující salinitu vody v oceánech<sup>5</sup>

## 2.2 Data

Pro vizualizaci tematických map potřebujeme dostatečné množství dat.

### 2.2.1 Geografická data

Pokud data budeme definovat jako fakta o jevu, tak geografická data jsou fakta vztahující se k vlastnostem povrchu Země, případně fakta o světě kolem nás. Tato data můžeme klasifikovat podle způsobu jejich získání a zdroje na primární a sekundární. Primárními zdroji dat jsou jednotlivci nebo organizace, které využívaná data získali nebo byli zapojeni do jejich získávání. Sekundární jsou opakem primárních, tedy u dat, které používají, nebyli

zapojení do jejich získání. Pro identifikování lokace na povrchu Země se používá zeměpisná délka a šířka, které dohromady tvoří zeměpisné souřadnice [16].

Tato klasifikace je pro nás důležitá, protože zdůrazňuje důležitost *metadat* (2.2.4). Jelikož knihovna Geovisto působí jako sekundární zdroj geografických dat, musíme brát v potaz fakt, že na využívaná data nemáme přímý vliv. Musíme tedy v metadatech nejen poskytnout informace o souboru dat, jako například jeho zdroj nebo časové období vzniku, ale také při jeho výběru dbát na jeho vhodnost pro využití v konkrétním tematickém kontextu.

## 2.2.2 GeoJSON

Pro reprezentování geografických dat ve webovém vývoji používáme jejich digitální formát GeoJSON.<sup>6</sup> Podle jeho standardu [2] je založený na *JavaScript Object Notation (JSON)*. Skládá se z jednotlivých objektů GeoJSON, které mohou být na sebe vázané a reprezentují *Feature*, *Geometry* nebo seznam objektů typu *Feature*.

Objekt *Geometry* reprezentuje body, křivky a povrchy v souřadnicovém prostoru. Jeho hlavním atributem je *coordinates*, což je množina, která se skládá buď z jedné pozice pokud je typu *Point*, množiny pozic pokud je typu *LineString*, nebo *MultiString*, množiny typů *LineString* v případě typu *Polygon* a nebo množiny typů *Polygon*, pokud je typu *MultiPolygon*. Pozice je množina číselných hodnot a musí tvořit alespoň dva prvky v následujícím pořadí: zeměpisná délka, zeměpisná šířka. Jako třetí prvek může být uvedena nadmořská výška, není však povinná.

Objekt *Feature* (obrázek 2.4) reprezentuje geografický objekt vyskytující se v prostoru, přesněji na povrchu země. Hlavním atributem je *geometry*, což je objekt definovaný výše. Mohou mít také atribut *properties*, který slouží k popsání vlastností objektu informačního charakteru. Tyto objekty mohou být sdružovány do objektu *FeatureCollection*.

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

Obrázek 2.4: Jednoduchý příklad formátu GeoJSON<sup>7</sup>

## 2.2.3 Tematická data

Pro vytvoření tematických map potřebujeme datové sady, které nám dodají fakta o jevu, který je vázaný na konkrétní místo na povrchu země. Zjednodušeně mluvíme o informacích

<sup>6</sup>Existují i jiné formáty pro reprezentaci geografických dat, například TopoJSON, ale zaměřil jsem se na GeoJSON, protože je primárně využíván v této práci

<sup>7</sup>Zdroj: <https://geojson.org/>

popisujících vlastnost daného místa. Takovým informacím říkáme právě tematická data [1]. Jako příklad si můžeme uvést statistiku vyjadřující hrubý domácí produkt vázanou na konkrétní státy (výpis 2.1).

```
1  [
2    {
3      "Country": "AUS",
4      "Value": 1380207.833,
5      "Year": "2019"
6    },
7    {
8      "Country": "AUT",
9      "Value": 433258.468,
10     "Year": "2020"
11   },
12   {
13     "Country": "BEL",
14     "Value": 521861.293,
15     "Year": "2020"
16   },
17   {
18     "Country": "CAN",
19     "Value": 1644037.286,
20     "Year": "2020"
21   }
22 ]
```

Výpis 2.1: Příklad zápisu tematických dat ve formátu JSON používaný v knihovně Geovisto. Jedná se o data přiřazující jednotlivým zemím hodnotu hrubého domácího produktu pro daný rok.

## 2.2.4 Metadata

Jedná se o data popisující data [3]. Jak bylo popsáno v podsekcí 2.2.1, jsou důležitou součástí datového celku, protože nám zajišťují validitu geografických a tematických dat. Popisují také vlastnosti dat, rozsah a další informace.

Jejich přítomnost je často klíčová pro správné využívání a získávání poznatků z mapy uživatelem. Pokud uživatel nebude mít dostatečná metadata o mapě, kterou má použít, tak nebude schopen vyčíst jím požadované informace. Může se jednat například o chybějící jednotky pro číselné hodnoty. V důsledku jejich absence se stává celý datový celek nepoužitelným.

Metadata mohou obsahovat zdroj geografických dat a tematických dat, kontakt na osobu či organizaci zodpovědné za vytvořenou mapu nebo rozšiřující informace o samotném tematickém souboru dat (obrázek 2.5). V tomto případě mluvíme primárně o jejich kontextu.

## 2.3 Vizualizace na webu

Jako hlavní médium pro vizualizaci geografických dat sloužil vždy papír. To se změnilo s příchodem internetu v devadesátých letech dvacátého století, který zpočátku umožnil převod papírových map do digitálního formátu. Jak se internet a jeho možnosti dále vyvíjely,

<sup>9</sup>Zdroj: <https://stats.oecd.org/>



Obrázek 2.5: Metadata popisující účel statistických dat, obsahující zdroj dat, kontakt na autora, jednotky a datum poslední úpravy dat.<sup>9</sup>

vyvíjela se s ním i vizualizace geografických dat. Nové technologie umožnily vytvářet interaktivní mapy a manipulaci s geografickými daty, použití animací, satelitních snímků či virtuální zobrazení okolí, například konkrétních ulic, pomocí souboru panoramatických snímků. Jako příklady těchto technologií si můžeme uvést běžně používané služby *Google Maps*<sup>10</sup> a *Google Earth*<sup>11</sup>.

Způsoby vizualizace map na webu si můžeme dle [14] rozdělit na *předem vytvořené mapy* a *mapy na míru*. První pojem označuje mapy, které jsou zpravidla statické, tzn. že po vytvoření už nejsou upravovány. Uživatel si je může stáhnout ve formátu obrázku a jedná se často o digitální podobu fyzických kopií. Druhý pojem se používá k označování map, jejichž obsah si uživatel může přizpůsobit. Jedná se primárně o výběr tematického souboru dat (2.2.3) a geografických dat k vizualizaci (2.2.1). Právě tato metoda nám umožňuje vizualizovat tematické mapy probírané v sekci 2.1.

Toto téma je dobré představit si už zde ve spojení daty a tematickými mapami, ale dále je rozvedeno v kapitole 4, kde jsou popsány konkrétní technologie používané pro vizualizaci geografických dat na webu a již existující řešení, včetně autorských systémů a programovacích knihoven. Dále se v této kapitole rozebírá knihovna Geovisto.

<sup>10</sup><https://www.google.com/maps>

<sup>11</sup><https://earth.google.com>

## Kapitola 3

# Použitelnost uživatelských rozhraní

Tato kapitola se zabývá pojmem uživatelské rozhraní jako takovým a jeho použitelností z hlediska uživatelského zkušlosti. Aby uživatelské rozhraní mohlo být plně využíváno k jeho funkci, musí být dobře navrženo. Špatný návrh nebo jeho realizace vedou ke snížení nejen celkové kvality aplikace, ale i její konkurenceschopnosti oproti jiným aplikacím s podobným využitím a použitelností v konkrétních případech jejího užití.

Špatné uživatelské rozhraní odrazuje potenciální uživatele od aplikace, což má za následek, že se stává zbytečnou. Můžeme disponovat sebelepším řešením problému, na který se aplikace zaměřuje, ale pokud uživatelé budou mít potíže při jejím používání, aplikace nebude využívána k řešení daného problému. Tento fakt ilustruje důležitost správného návrhu uživatelského rozhraní a jeho implementace.

Je třeba zvážit, jaká skupina uživatelů bude aplikaci využívat, a z toho vyvodit požadavky této skupiny na rozhraní. Hlavní roli zde bude hrát faktor technické úrovně a zdatnosti našich potenciálních uživatelů. Například méně zdatní uživatelé budou požadovat jednodušší ovládací prvky než pokročilí a budou vyžadovat více asistence při orientaci v aplikaci.

Proto v této kapitole bude výše uvedená problematika více rozvedena. Bude ustanoven význam pojmu uživatelského rozhraní a pravidla pro jeho design. V této kapitole se čerpalo primárně z [6] a [13].

### 3.1 Uživatelské rozhraní

Pojmem uživatelské rozhraní označujeme součást systému, se kterou uživatelé pracují pro dosažení jejich cílů. Může se jednat o obsluhu strojů nebo nástrojů, nás ale v kontextu této práce zajímá konkrétně interakce uživatele s počítačovými programy. V tomto případě uživatelské rozhraní slouží jako propojení mezi uživatelem a počítačovým programem [15]. Běžného uživatele nezajímají technologie a implementační řešení použitá v daném programu, protože vnímají jenom jeho uživatelské rozhraní jako celý program. Je to jediná část programu, se kterou přicházejí do styku a také se kterou pracují, a podle ní potom hodnotí kvalitu celého programu.

Jak už bylo nastíněno v úvodu této kapitoly, pokud je uživatelské rozhraní špatně navrženo, vede to k negativní uživatelské zkušlosti a nespokojenosti s celou aplikací. Pokud se jedná například o aplikaci používanou zaměstnanci pro výkon jejich úkolů na denní bázi, tak špatné uživatelské rozhraní může vést k frustraci a zvýšenému stresu zaměstnanců. To se pak negativně promítá do jejich výkonu a objevily se i případy, kdy vznikly velké ča-

sové a finanční náklady pro jejich zaměstnavatele způsobené právě špatným uživatelským rozhraním. Vzniklé škody nemusí být ale jen finančně nákladné. Pokud se tyto nedostatky objevují i u důležitých systémů, jako jsou například aplikace pro kontrolu letových provozů či aplikace řídicí provoz elektrárny, mohou mít katastrofální důsledky. [15]

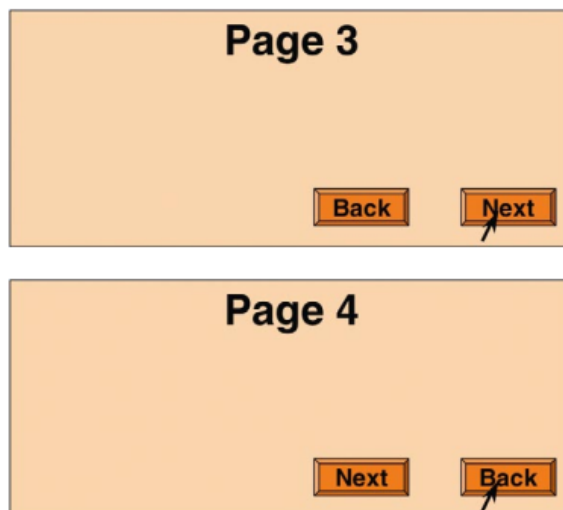
## 3.2 Pravidla pro návrh uživatelského rozhraní

Abychom mohli předejít problémům uživatelských rozhraní popsaným v úvodu této kapitoly a v sekci 3.1, existují pravidla pro návrh uživatelských rozhraní. Pomáhají nám nejen při návrhu našeho vlastního, ale i při odhalování nedostatků už existujících uživatelských rozhraní. Můžeme z nich čerpat ať už jsme v roli někoho, kdo má tyto nedostatky pouze odhalit, tak i v roli programátora, který má odhalené problémy odstranit a nahradit je lepším řešením. Následující pravidla zasadíme do kontextu webových stránek, jelikož na jejich uživatelská rozhraní se tato práce zaměřuje.

Zde uvedená pravidla jsou pouze můj výběr z [6], který jsem považoval za nejvíce relevantní k uživatelskému rozhraní systému pro vizualizaci geografických dat. Další pravidla také definují nebo jinak zpracovávají [13] a [15]. Například [13] se také pak dodatečně zaměřuje na metody zhodnocení výsledného návrhu nebo na proces analýzy potřeb uživatelů.

### 3.2.1 Konzistence

Všechny informace a ovládací prvky uživatelského prostředí by měly být konzistentní. Například pokud máme vícero prvků sloužících stejnému účelu, tak by se měly na každé stránce vyskytovat na stejném místě (obrázek 3.1). Jejich styl, tudíž barva, font, apod., by měly být stejné. Při návrhu a implementaci bychom také měli dbát na standardy, pokud pro daný prvek stránky existují. Konzistence nám pomůže zajistit lepší orientaci uživatele v rozhraní a uživatelé ho budou interpretovat stejně.



Obrázek 3.1: Příklad nekonzistentního umístování tlačítka *Next*, které mate uživatele<sup>1</sup>

---

<sup>1</sup>Zdroj: [6]



### 3.2.2 Struktura rozhraní

Správná struktura nám poskytuje lepší přehlednost a schopnost najít na stránce pro nás relevantní informace. Struktura je také důležitá pro zpracovávání informací, což znamená, že pokud například textu chybí struktura (obrázek 3.2), bude hůře čitelný a jeho přečtení nám zabere delší dobu, případně ho uživatel může dokonce úplně přeskočit.

Unstructured:

**You are booked on United flight 237, which departs from Auckland at 14:30 on Tuesday 15 Oct and arrives at San Francisco at 11:40 on Tuesday 15 Oct.**

Structured:

**Flight: United 237, Auckland → San Francisco**  
**Depart: 14:30 Tue 15 Oct**  
**Arrive: 11:40 Tue 15 Oct**

Obrázek 3.2: Praktický příklad rozdílu v čitelnosti strukturovaného a nestrukturovaného textu<sup>3</sup>

Podobný problém může nastat u číselných hodnot. Pokud neděláme mezery mezi řády (obrázek 3.3), delší číselný zápis se může stát také těžko čitelným a uživatel opět stráví delší dobu čtením daného zápisu. Stejně pravidlo uplatňujeme například u telefonních čísel nebo data narození.

1234 5678 9012 3456  
1234567890123456

Obrázek 3.3: Rozdíl mezi zápisem s mezerami mezi řády a bez nich<sup>4</sup>

Dalším příkladem, kde musíme zdůraznit důležitost struktury, je celkové rozložení informací a prvků. Jak můžeme vidět na obrázku 3.4, vizuální nepřehlednost a špatná hierarchie ovládacích prvků citelně snižuje uživatelskou přívětivost a znesnadňuje použitelnost, protože uživatel stráví zbytečný čas orientováním se v rozhraní a hledáním ovládacích prvků, které zajišťují jím požadované funkce. Na konkrétním případě z obrázku 3.4 můžeme tento nedostatek napravit například centrováním obsahu podle polí se vstupy a lepším vizuálním oddělením jednotlivých skupin funkcí.

### 3.2.3 Text

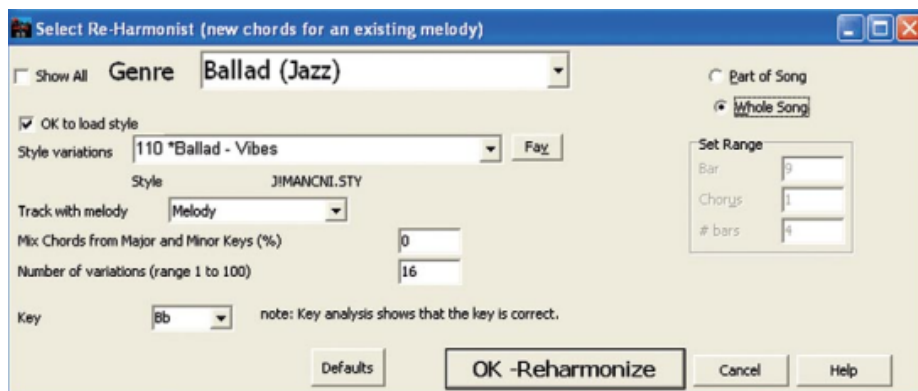
Jako hlavním nositelem informací slouží často text, je proto velmi důležité, abychom ho správně zobrazovali. Text by měl být také dobře napsaný, neměl by obsahovat zbytečné informace, ale rozhodně by neměl opomíjet ty důležité. Jako běžný příklad nám mohou sloužit chybové zprávy. Špatná chybová zpráva nám sdělí buď příliš mnoho informací, informace,

---

<sup>3</sup>Zdroj: [6]

<sup>4</sup>Zdroj: [6]

<sup>5</sup>Zdroj: [6]



Obrázek 3.4: Příklad špatné celkové hierarchie<sup>5</sup>

kteří nejsou relevantní pro uživatele, nebo naopak příliš málo informací, ze kterých uživatel nepozná, o jakou chybu se jedná. Chybová zpráva, která vypíše chybovou hlášku programovacího jazyka nebo generickou zprávu nesouvisející s danou chybou, uživateli nepomůže, ale spíše ho zmate.

Zde také uplatníme pravidla zavedená v podsekcích 3.2.1 a 3.2.2. Text stejné důležitosti by měl být na všech stránkách stejně velký se stejným fontem. Odlišnosti ve stylování textu nám právě pomohou rozlišovat důležitost textu pro uživatele. Výjimky z tohoto pravidla mohou být například bloky se zdrojovými kódy. Dále by měl být text dobře strukturovaný pro lepší orientaci (obrázek 3.2). Měli bychom také zvolit vhodné úpravy textu, které nebudou vizuálně rušivé či nebudou dělat text těžce čitelným (obrázek 3.5). Zde mezi časté prohřešky patří příliš malý text, text přes obrázek či na jiném rušivém pozadí, špatný kontrast oproti pozadí, zarovnávání seznamů na střed nebo přílišná repetice, například v předem zadaných možnostech, ze kterých si má uživatel vybrat.

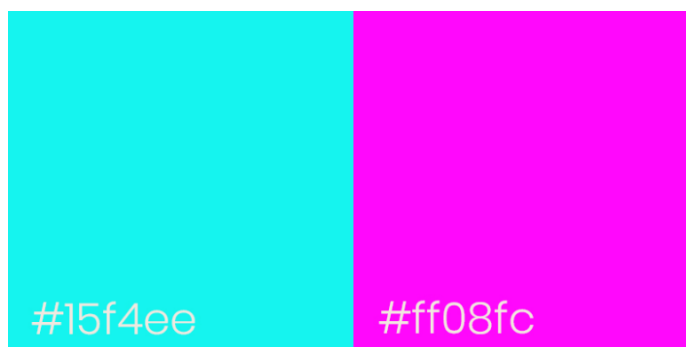


Obrázek 3.5: Příklad nekonzistentní a celkově špatné práce s textem<sup>6</sup>

### 3.2.4 Používání barev

Dalším důležitým faktorem v kvalitě uživatelského rozhraní, tedy hlavně jeho vizuální stránky a přívětivosti, je využití barev a jejich kombinování. Základní informací, kterou bychom měli mít na paměti při každé práci s barvami, je fakt, že lidské oko vnímá rozdíly v kontrastu mezi barvami daleko citlivěji než jas barev [6].

Ke kontrastu mezi barvami ovšem musíme přistupovat opatrně a jeho intenzitu používat s rozvahou vzhledem k využití. Pokud například chceme oddělit barevně pozitivní zprávu zelenou a negativní zprávu červenou, tak zde je vysoký kontrast vhodný. Pokud máme ale třeba pozadí jinak zabarvené, než prvek stránky, který ho překrývá, je vhodné zvolit kontrast menší. Vysoký kontrast by v tomto případě mohl působit vizuálně nevhledně (obrázek 3.6).



Obrázek 3.6: Nevzhledná kombinace barev snižující kvalitu uživatelského rozhraní<sup>7</sup>

Při vizualizaci hodnot pomocí barev, například v grafech nebo u již v podsekcí 2.1.2 zmiňovaném kartogramu, je vhodné, aby jednotlivé odstíny barvy byly na první pohled rozlišitelné. Zde totiž způsob použití barev silně ovlivňuje výslednou použitelnost celého prvku. Běžně používáme různé odstíny jedné barvy, ale pokud máme k vizualizaci mnoho hodnot, je vhodnější použít širší spektrum různých příbuzných barev.

Měli bychom také dbát na potřeby lidí se sníženým barvocitem. Pokud je uživatel barvoslepý, neznamená to, že nevidí barvy, ale má problém rozlišovat určité kombinace barev [6]. Přímo v kontextu knihovny Geovisto toto téma kvalitně zpracovává bakalářská práce [7].

### 3.2.5 Rychlost a responzivní design

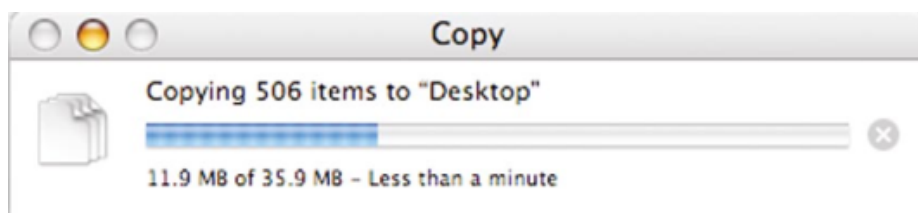
Pod pojmem responzivita zde není myšlena optimalizace webových stránek pro různé velikosti a formáty zobrazení, ale požadavek na informování uživatele o stavu aplikace. Správný návrh uživatelského rozhraní by vždy měl obsahovat okamžité reakce na vstup uživatele, poskytovat alespoň odhad, jak dlouho požadovaný úkol zabere (obrázek 3.7), spravovat frontu úkolů a umožňovat vykonávání dalších interakcí s rozhraním při výkonu daného úkolu. Je záhodno se vyvarovat situací, kdy uživatel čeká na vykonání jím zadaného požadavku a zároveň nemůže v aplikaci dělat cokoliv jiného. Uživatelé by také měli mít možnost zrušit vykonávané operace, zvláště pokud zabírají dlouhou dobu na vykonání.

Abychom mohli zajistit požadavky uvedené v předchozím odstavci, musí být při implementaci aplikace brán ohled na její rychlost a dobu provádění operací v ní. Nutno myslet

<sup>6</sup>Zdroj: <https://www.pennyjuice.com/>

<sup>7</sup>Zdroj: <https://approval.studio/wp-content/uploads/2020/11/neon-color-combination.jpg>

na tento faktor je důležité hlavně u prvků, které uživatel bude potencionálně využívat v interakci s aplikací. Pokud chceme aby uživatel vnímal reakci těchto prvků na jeho vstup jako instantní, musí být vykonána do 0.1 sekundy [6].



Obrázek 3.7: Příklad správného informování uživatele o postupu vykonávání požadavku<sup>8</sup>

---

<sup>8</sup>Zdroj: [6]

## Kapitola 4

# Existující řešení

V této sekci bude představeno již existující řešení tématu rozebíraného v sekci 2.3. Tato řešení lze rozdělit na dvě kategorie. První kategorií jsou rozšiřující knihovny pro programovací jazyky, kterými můžeme vizualizovat geografická data. Jejich nevýhodou je potřeba programovacích dovedností, což z nich často dělá těžko použitelné řešení pro běžné uživatele. Druhá kategorie jsou autorské systémy. Ty poskytují již existující nástroje na vytvoření požadované vizualizace bez programátorských dovedností.

Tyto kategorie jsou uvedeny již zde, ale více budou využity až při popisu knihovny Geovisto. Pro naše účely je totiž důležitější zhodnotit jiné současné řešení z pohledu použitelnosti jejich uživatelského rozhraní, což nám poskytne srovnání pro knihovnu Geovisto a potenciálně i podněty k případnému vylepšení knihovny.

### 4.1 Technologie pro webovou vizualizaci

Základem každé stránky je značkovací jazyk *Hypertext Markup Language*, zkráceně HTML. Tento jazyk určuje strukturu webových stránek pomocí předem definovaných značek. Tyto značky lze do sebe navzájem zanořovat. Můžeme je také rozlišit na otevírací značky a zavírací. Právě tento konstrukt nám umožňuje zanořit jeden typ značky do druhé. Mezi hlavní funkce HTML také patří definice obsahu různého formátu, který chceme na webové stránce vizualizovat. V momentě, kdy webový prohlížeč načte webovou stránku, vytvoří tzv. *Document Object Model*, zkráceně DOM. Jedná se o strom HTML objektů.

Na popis, jak prvky jazyka HTML mají vypadat, používáme jazyk *Cascading Style Sheets*, neboli CSS. Účelem tohoto jazyka je definovat zobrazení HTML značek a jejich obsahu, jako například rozložení, barva, použitý font apod. CSS můžeme používat v samostatných souborech, nebo jej přímo vpisovat do HTML dokumentu a to buď do jeho hlavičky, nebo do HTML značek jako jejich atribut.

Pro vykreslování grafických prvků používáme SVG, značkovací jazyk založený na XML pro vytváření vektorové grafiky. Jeho výhodou je, že SVG je určený pro použití ve webovém prostředí a kvalita jím vykreslených grafických prvků není závislá na velikosti. Jelikož jeho zápis je textový, může být výsledný prvek upraven i v libovolném textovém editoru. [11]

Vykreslovat grafické prvky pomocí programovacího jazyka JavaScript a HTML elementu `<canvas>` nám umožňuje Canvas API. Využívá se pro animaci, vizualizaci dat či zobrazování videí. Této funkcionality dosáhneme rozhraním `CanvasRenderingContext2D`, které vykresluje objekty definované zdrojovým kódem. [10]

Pokud chceme s webovou stránkou provádět interakci, měnit dynamicky její obsah nebo používat pokročilé grafické komponenty jako například animace, potřebujeme programovací jazyk *JavaScript*. Jedná se o velice rozšířený jazyk, který je v době vzniku této práce elementárním nástrojem pro tvorbu webových stránek. Základní způsob, jakým JavaScript modifikuje webovou stránku, je jeho schopnost přistupovat ke stromu HTML DOM a měnit v něm objekty. Může upravovat jejich vlastnosti, například CSS styly, obsah, umožňuje přidávat a odebírat tyto objekty. Používá se také jako prostředek pro reakci na uživatelské vstupy, díky čemuž můžeme vytvářet dynamické webové stránky, které mění svůj obsah či zobrazení v reakci na tyto vstupy. Výhoda této funkce je také, že tyto změny mohou probíhat bez nutnosti znovu stránku načítat.

Na tento jazyk navazuje mnoho samostatných externích rozšiřujících knihoven, mimo jiné i ty, které budou popsány v následující sekci. Pro vývoj webových aplikací v jazyce JavaScript je též často využíván systém *Node.js*, který umožňuje spouštět kód napsaný v JavaScriptu i mimo webový prohlížeč na straně serveru.

## 4.2 Autorské systémy

Jedná se o systémy, které nám umožňují používat jejich funkce bez programátorských dovedností. Zatímco u programovacích knihoven bychom museli tyto funkce implementovat pomocí kódu, zde jsou uživateli poskytnuty ve zjednodušené formě. Jedná se o realizaci těchto funkcí pomocí různých uživatelsky přívětivých prvků jako třeba tlačítka. Jejich hlavní výhodou je tedy vysoká rozšiřitelnost díky absenci požadavku na programátorské dovednosti.

Jako příklad lze uvést [GoogleMaps](https://www.google.com/maps)<sup>1</sup> nebo [Mapy.cz](https://mapy.cz/)<sup>2</sup>. V obou případech se jedná o webové aplikace, které poskytují kromě základní mapy také vrstvy složené ze satelitních snímků nebo vrstvu s dopravními informacemi. Interakce uživatele se zde například omezují na umístování bodů do mapy a vytváření tras. Ani jeden z těchto systémů neposkytuje možnost vizualizovat statistická data pomocí tematických map.

## 4.3 Programovací knihovny

Aby uživatelé mohli používat funkce programovacích knihoven, jsou po nich vyžadovány programátorské dovednosti. Výhodou těchto nástrojů oproti autorským systémům, i přes zvýšené požadavky, je možnost vytvářet vlastní nástroje nebo upravovat již ty stávající.

V této sekci jsou uvedené a popsány knihovny *D3.js* a *Leaflet*, které jsou využívány i v knihovně *Geovisto* a vylepšovány jejich komunitami. Jako další příklad programovací knihovny jsem zvolil *GoogleMaps API*, jejíž zdrojový kód není oproti zmíněným knihovnám veřejně přístupný, je zpoplatněná a vyvíjena společností Google.

### D3.js

*D3* je open-source knihovna, která pomocí technologií HTML, SVG a CSS umožňuje vizualizovat statistická data mapováním těchto dat na DOM [9]. S elementy v DOM lze díky této knihovně snadněji manipulovat. Obsahuje několik možností pro výběr elementů, úpravy jejich CSS atributů nebo možnost dynamicky elementy přidávat a mazat.

---

<sup>1</sup><https://www.google.com/maps>

<sup>2</sup><https://mapy.cz/>

Data umí vizualizovat v různých formách, jako například několika typy grafů nebo právě tematickými mapami. Data lze i různě animovat. Poskytuje také uživateli možnost s vykreslenými daty pracovat. Knihovna umožňuje registrovat posluchače událostí na jednotlivé elementy, což může posloužit pro implementaci vybírání v zobrazení oblasti, posouvání prvků nebo přiblížení prvků kliknutím. Tyto funkce lze využívat díky JavaScriptovému API, které tvoří na sobě nezávislé moduly.

## Leaflet

Leaflet je open-source knihovna implementovaná v programovacím jazyce JavaScript, zaměřená na vytváření interaktivních map. Poskytuje několik základních funkcí. Má vestavěné prvky uživatelského rozhraní pro ovládání mapy, jako například přiblížování a oddalování nebo různé reakce na vstupy z počítačové myši. Podporuje také datový formát GeoJSON a rastrovou i vektorovou grafiku. Lze ji snadno rozšířit jak programově, tak pomocí již dostupných rozšíření vytvořených její komunitou. [8]

## GoogleMaps API

Jedná se o zpoplatněnou službu, která poskytuje JavaScriptové API umožňující přístup k datům používaným v aplikaci GoogleMaps, popsáných v předchozí sekci. Obsahuje podporu formátu GeoJSON, rozsáhlé uzpůsobení estetického vzhledu mapy, ovládací prvky nebo vrstvu značek [4]. Lze zde například vytvořit i kartogram, knihovna na tento účel ale není zaměřená. Celkově je toto API nevhodné na vizualizaci statistických dat pomocí tematických map.

## 4.4 Knihovna Geovisto

Knihovna Geovisto slouží k vizualizaci geografických dat a je postavena na knihovně Leaflet. Obsahuje několik předem vytvořených vrstev pro tvorbu tematických map a pomocné nástroje. Architektura knihovny je založená na komponentách, což umožňuje její snadnější rozšiřování. To je dále podpořeno faktem, že knihovna je open-source a je uložena pod licenci MIT<sup>3</sup>. Poskytuje API na zpracovávání datového formátu GeoJSON.

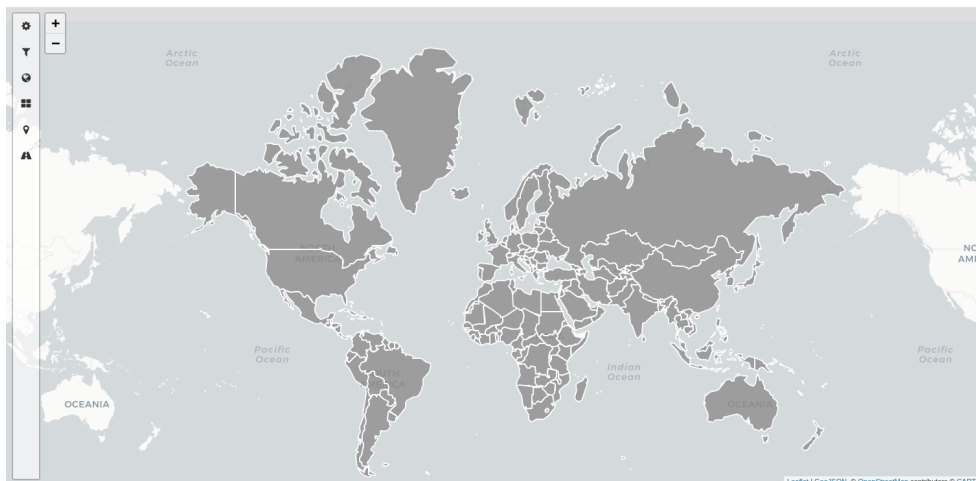
Hlavními vlastnostmi knihovny z uživatelského pohledu je automatické zpracování vstupních dat na výslednou mapu bez potřeby zásahu uživatele, možnost rozsáhle upravovat nastavení a zobrazení jednotlivých vrstev a nástrojů pomocí konfiguračních souborů a funkce umožňující importovat i exportovat tyto soubory pro další využití. Knihovna v době psaní této práce implementuje 4 vrstvy. Základní podkladovou mapu, vrstvu kartogramu *Choropleth Layer*, značek *Marker Layer* a spojení *Connection Layer*.

V této kapitole bude popsáno uživatelské rozhraní knihovny, datový model, architektura a způsob konfigurace. Hlavním zdrojem pro tuto kapitolu bylo použito [5].

### 4.4.1 Uživatelské rozhraní

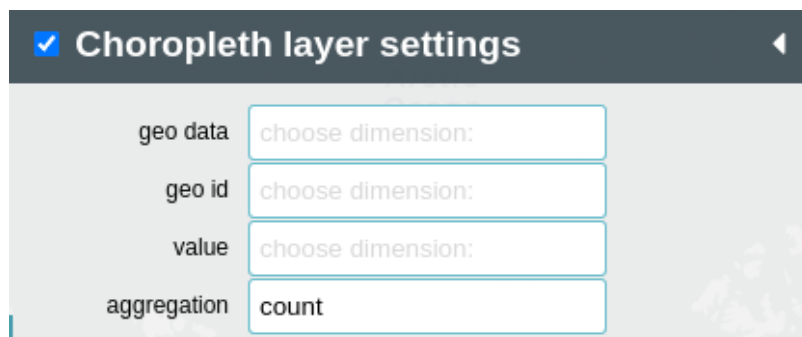
Základ uživatelského rozhraní v knihovně tvoří mapa s možností přiblížit či oddálit obsah. Na ní je potom na levé straně pohledu umístěn panel s ovládacími prvky vrstev a nástrojů (obrázek 4.1). Mapa může fungovat a být ovládána i bez něj.

<sup>3</sup><https://opensource.org/licenses/MIT>



Obrázek 4.1: Pohled na uživatelské rozhraní Geovista s postranním ovládacím panelem.

Postranní ovládací panel (obrázek 4.2) se skládá z menu pro výběr nástroje nebo vrstvy a rozbalujícího se okna, které se objeví po kliknutí na položku menu. Obsahuje nastavení daného nástroje, kde je každá položka specifická pro daný nástroj, nebo vrstvy, které mají většinou společné položky *geo id*, *value*, *geo data* a *aggregation*. Tyto jednotlivé položky měněné pomocí rozbalovacích menu s předem definovanými možnostmi jsou specifické datové dimenze mapující tematická data. Možnosti lze také změnit přímo editací aktuálně vybrané položky či napsat do vstupního pole vlastní. Celý panel je záměrně mírně průhledný, aby v rozbaleném stavu neztěžoval orientaci v mapě.



Obrázek 4.2: Pohled na ovládací panel vrstvy vizualizující kartogram.

Na obrázku 4.2 dimenze *geo id* určuje podle unikátního identifikátoru zvoleného například v rámci standardu ISO 3166-1 alpha-3<sup>4</sup> datovou dimenzi na mapě, ke které budou přiřazeny konkrétní tematická data označená stejným identifikátorem. Další dimenze *value* mapuje z tematických dat číselnou hodnotu, která určuje barevnou intenzitu polygonu, se kterým sdílí identifikátor. *Aggregation* obsahuje metodu agregace hodnot z dimenze *value*. Pomocí *geo data* vybíráme aktuálně používanou sadu geografických dat.

Jednotlivé vrstvy fungují nezávisle na sobě, včetně konfigurace a nastavení dimenzí, jdou navzájem kombinovat a poskytují tím způsob vytváření komplexnějších pohledů.

<sup>4</sup><https://www.iso.org/iso-3166-country-codes.html>



## 4.4.2 Datový model

Knihovna Geovisto pracuje s datovým formátem JSON. Tento formát je využíván pro vytváření konfiguračních souborů. Ty ovlivňují chování mapy, postranního ovládacího panelu i jednotlivých vrstev a nástrojů. Lze je exportovat i importovat. Na obrázku 4.3 je ukázaná struktura konfigurace. Položka *type* označuje typ vrstvy nebo nástroje, *id* je unikátní identifikátor v mapě, *enabled* určuje, zda je výchozí stav vrstvy aktivovaná nebo deaktivovaná a *layerName* je název zobrazovaný v uživatelském rozhraní. Položky sjednocené v seznamu *data* jsou nastavení pro konkrétní vrstvu, v tomto případě pro kartogram.

```
{
  "type": "geovisto-tool-layer-choropleth",
  "id": "geovisto-tool-layer-choropleth",
  "enabled": true,
  "layerName": "Choropleth layer",
  "data": {
    "country": "",
    "value": "",
    "aggregation": "",
    "scaling": "median",
    "range": 7,
    "customColor": true,
    "color": "#e32400"
  }
}
```

Obrázek 4.3: Konfigurace vrstvy vizualizující kartogram zapsaná v JSON formátu.

Druhým typem dat ve formátu JSON jsou tematická data popsaná v sekci 2.2.3. Poskytují hodnoty pro datové dimenze, které lze měnit pomocí postranního ovládacího panelu. Na obrázku 4.4 je ukázána jedna možná struktura těchto dat. Zde zobrazené položky JSON objektu nejsou klíčová slova, jejich název může být libovolný. Reprezentují hodnotu hrubého domácího produktu pro Austrálii a Belgie. Položka *Country* zde obsahuje již zmiňovaný identifikátor podle standardu alpha-3, který přiřazuje celý JSON objekt, sdružující s sebou související tematická data, ve kterém se nachází, příslušnému polygonu se stejným identifikátorem. Položka *Value* obsahuje číselnou hodnotu, která může sloužit jako obsah dimenze Value. *Year* obsahuje datum, pro které je hodnota platná a může být použita při vytváření skupin. Všechny tyto položky se poté zobrazí v postranním ovládacím panelu jako volby pro hodnoty dimenzí.

## 4.4.3 Architektura

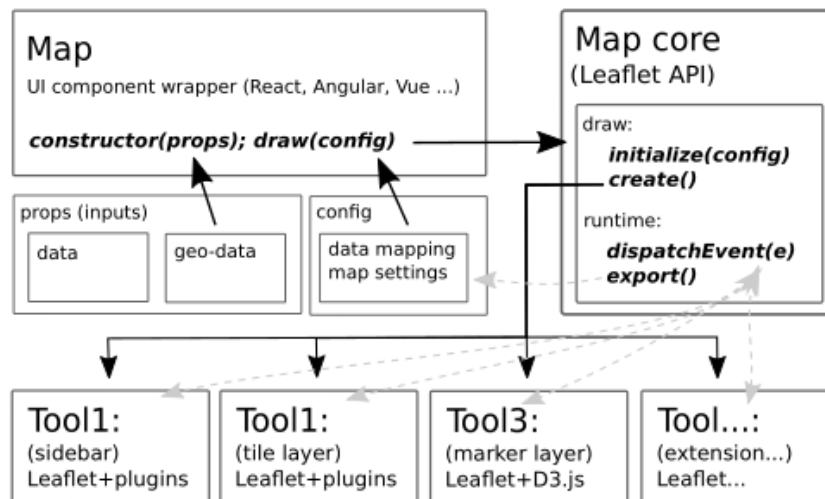
Architektura knihovny Geovisto (obrázek 4.5) se skládá ze samotné mapy, která je poskytována knihovnou Leaflet a skládá se také z jednotlivých vrstev tvořených pomocí vektorového grafického formátu SVG, generovaným také knihovnou D3.js. Leaflet API serializuje a deserializuje konfiguraci mapy a vykreslí HTML objekty do uživatelského rozhraní. Vrstvy mezi sebou komunikují pomocí událostí.

```

[
  {
    "Country": "AUS",
    "Value": 1380207.833,
    "Year": "2019"
  },
  {
    "Country": "BEL",
    "Value": 521861.293,
    "Year": "2020"
  }
]

```

Obrázek 4.4: Tematická data o hrubém domácím produktu za daný rok ve formátu JSON.



Obrázek 4.5: Architektura knihovny Geovisto.<sup>5</sup>

Obalovacím prvkem Geovista může být komponenta knihovny React, nebo jiné knihovny nesoucí stejnou funkci jako React, ale knihovna funguje i samostatně. Zde se používá druh dat nazývaný *props*. Ten umožňuje programově ovlivňovat chování celé aplikace. Komponenta též obsahuje konfiguraci, která umožňuje nastavit chování mapy a jednotlivých nástrojů. Jejím účelem je vytvořit instanci knihovny Geovisto a dodat jí potřebná data, případně vytvořit instance nástrojů. Tato komponenta také vykresluje samotnou mapu popisanou v předchozím odstavci.

<sup>5</sup>Zdroj: [5]

# Kapitola 5

## Analýza

Pro řešení problémů uživatelského rozhraní knihovny Geovisto bylo nejprve potřeba tyto problémy identifikovat. Toho bylo docíleno pomocí uživatelského testování. V této kapitole bude popsáno získání vhodných sad tematických dat, samotné testování a jeho vyhodnocení.

### 5.1 Cílová skupina

Abychom mohli identifikovat problémy rozhraní pomocí uživatelského testování, musíme nejprve zvolit vhodnou skupinu uživatelů. V rámci testování se zaměřujeme čistě na uživatelské rozhraní, programátorskou stránku knihovny opomíjíme a stejně tak se nezabýváme i vytvářením map pomocí dat a konfigurací.

Cílová skupina jsou tedy uživatelé všech úrovní počítačové gramotnosti. Cílem je, aby knihovna byla snadno použitelná pro co největší škálu uživatelů a zvýšila se její rozšiřitelnost. Může být totiž využita nejen jako samostatný nástroj, ale i jako komponenta vložená do webových stránek třetích stran. Příkladem může být tematická mapa vizualizující statistické údaje v článku zpravodajského webu.

Od uživatelů pro účely tohoto testování nejsou vyžadovány programátorské dovednosti, pouze základní znalost anglického jazyka. Jejich hlavním úkolem je pouze získat z tematické mapy požadované informace a vyhodnotit je, což identifikuje chyby uživatelského rozhraní.

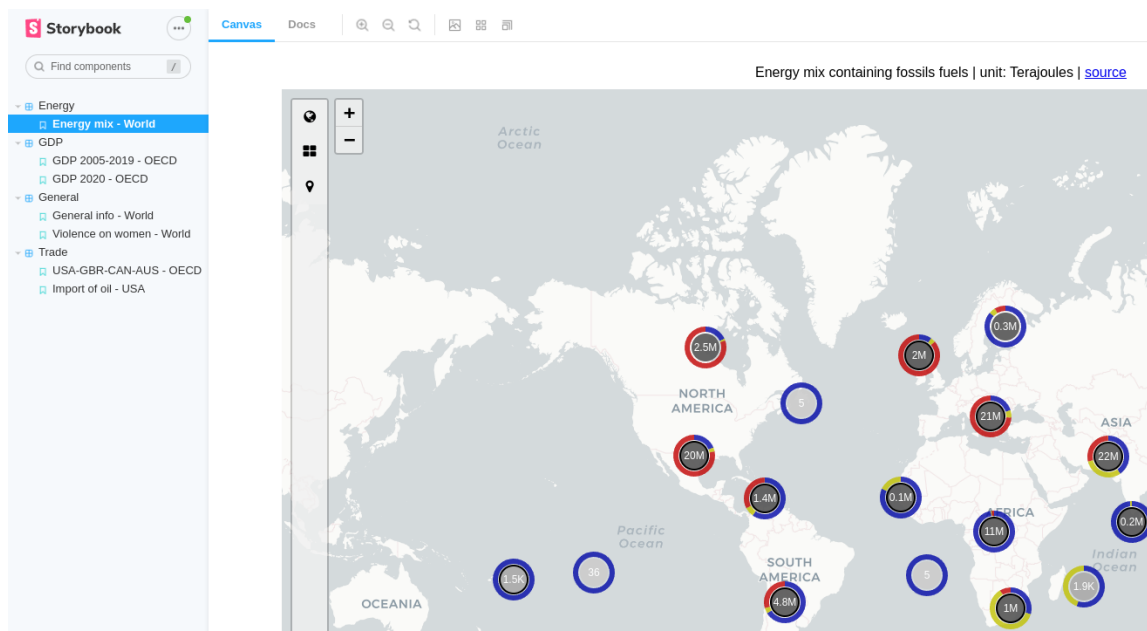
### 5.2 Testovací sada

Před provedením samotného uživatelského testování musíme vytvořit případy užití, které by měly obsahovat reálné informace. Musí být zvolena taková tematická data, aby nebyla příliš komplikovaná a neodváděla pozornost od funkcí uživatelského rozhraní, ale zároveň aby nebyla příliš jednoduchá. Uživatel by se mohl poté přestat soustředit na získávání informací a přehlédnout chyby v rozhraní. Cílem je vytvořit pohled pro každou vrstvu implementovanou v době psaní této práce.

#### 5.2.1 Vytváření případů užití

Pro vytváření pohledů poslouží nástroj Storybook. Umožní nám vytvořit od sebe izolované instance map knihovny Geovisto a tím vzniknou samostatné případy užití (obrázek 5.1). Dimenze instancí jsou pomocí konfigurace předem zvoleny, aby se uživatelé při testování nezdržovali manipulací s dimenzemi, protože ostatně nejsou ani předmětem této analýzy.

Už při vytváření těchto pohledů jsem identifikoval jeden z nejzásadnějších problémů uživatelského rozhraní knihovny Geovisto. V době vytváření těchto pohledů nebyl způsob, jak k danému pohledu připojit a zobrazit metadata, což by vedlo k tomu, že uživatel při testování ani nevěděl, jaká data jsou na mapě vizualizována, a v důsledku toho by nemohl splnit ani jeden úkol testování. Dočasným řešením se ukázalo umístění základního popisu datového setu nad mapu, jak je znázorněno na obrázku 5.1.



Obrázek 5.1: Testovací pohled na vrstvu značek Marker Layer.

## 5.2.2 Získání tematických dat

Ke každému pohledu bylo nutno získat tematická data. Geografická data ve formátu GeoJSON byla použita ve všech případech stejná, protože získávání tematických datových setů bylo nejvhodnější pro mapu světa. Za účelem získání dat jsem vytvořil nástroj, který tato data automaticky stáhne z dostupných API a převede tato data na jednotnou strukturu ve formátu JSON.

### Geovisto Data Miner

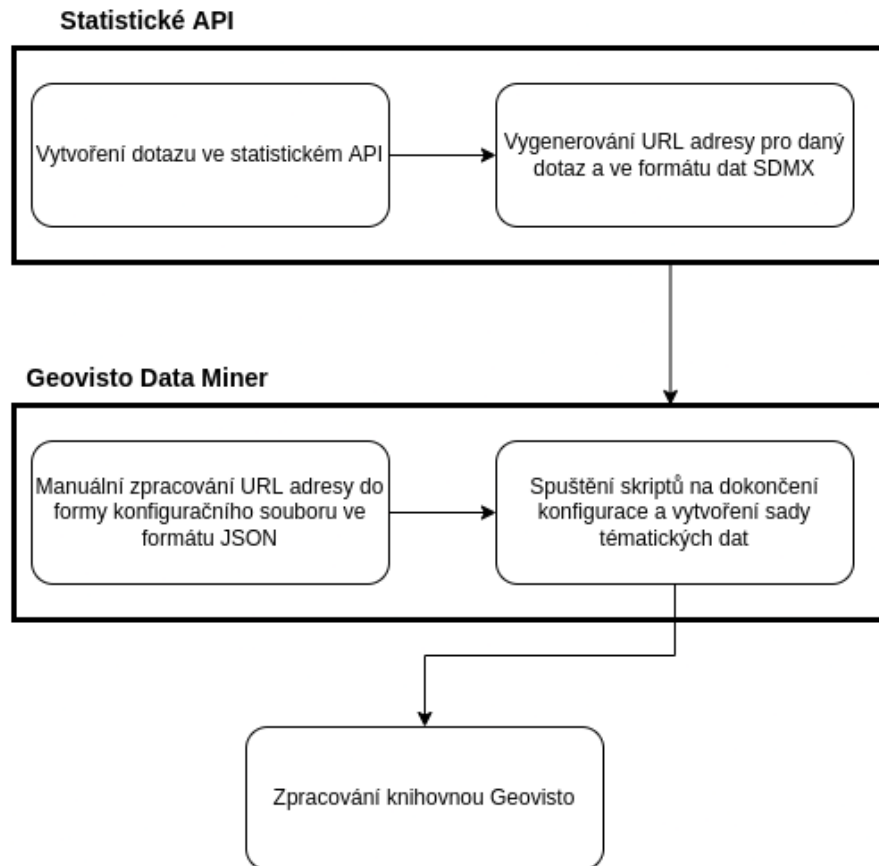
Jedná se o nástroj pro automatizaci a standardizaci tematických dat vyvinutý speciálně pro knihovnu Geovisto. Pracuje s veřejně dostupnými statistickými API jako například UNdata API<sup>1</sup>, což je statistické API organizace spojených národů a obsahuje velkou sadu dat pro skoro každou zemi na světě. Další podobné API je organizace OECD<sup>2</sup>. Jak je vidět na obrázku 5.2, samotný výběr dat automatizovat nejde. Uživatel musí nejdříve vytvořit vlastní dotaz, který chce stáhnout. Poté ze statistického API získá URL adresu pro export dat, přičemž ta musí být podle standardu SDMX<sup>3</sup>, a tuto adresu si uloží. Adresa musí být následně manuálně rozdělena podle dokumentace nástroje do konfiguračního souboru ve

<sup>1</sup><http://data.un.org/Host.aspx?Content=API>

<sup>2</sup><https://data.oecd.org/api/>

<sup>3</sup><https://sdmx.org/>

formátu JSON. Pak stačí spustit skript `get_configurations.sh`, který vypíše strukturu dat s indexy, na kterých je jaká položka uložena. Tyto indexy použijeme pro dokončení konfiguračního souboru, kde je potřeba namapovat index na námi zvolený název datové dimenze. Tímto krokem je konfigurace hotová a výstup získáme spuštěním skriptu `run.sh`. Výsledkem je sada tematických dat.



Obrázek 5.2: Diagram demonstrující postup při získávání tematických dat pomocí nástroje Geovisto Data Miner.

### 5.2.3 Výsledná testovací sada

Pomocí nástrojů popsaných v sekci 5.2 jsem vytvořil 7 testovacích pohledů. Byly tvořeny tak, aby využívaly každou vrstvu, která je v době psaní této práce implementována, což pomůže k odhalení potencionálních problémů i v jednotlivých vrstvách. Všechny pohledy obsahují reálná data, jedná se například o hrubý domácí produkt zemí, které jsou členy organizace OECD, nebo import surové ropy do Spojených států amerických. Součástí sady jsou též pohledy kombinující více vrstev dohromady.

## 5.3 Uživatelské testování

Uživatelské testování proběhlo na skupině uživatelů, ve které byly zastoupeny všechny úrovně počítačové gramotnosti od začátečníků po programátory. Testování proběhlo formou rozhovorů a plněním úkolů vztahujících se k příkladům použití definovaných v sekci 5.2. Úkoly se skládaly z 11 otázek, jejichž odpovědi byl uživatel schopný formulovat po vyhledání požadovaných informací v mapě. Uživateli jsem se průběžně dotazoval při řešení úkolů, zda jim něco na práci s mapou vadí nebo zda jim nějaká funkce chybí. Stejně otázky jsem jim položil po vykonání všech úkolů, jako jistou formu shrnutí.

Skupina se skládala pouze z pěti uživatelů. Důvodem tohoto rozhodnutí byla skutečnost, že už u třetího uživatele jsem zaznamenal, že identifikované problémy se začaly opakovat a testování poté přestalo přinášet nové informace, které bych mohl použít. U dalších dvou uživatelů jsem proto už jen úspěšně ověřil validitu odpovědí prvních tří a rozhodl se testování ukončit. Každý rozhovor trval i se splněním úkolů zhruba 20 minut.

### 5.3.1 Použité metriky

Pro účely druhého testování, které proběhlo po implementaci navržených řešení, jsem použil konkrétní metriky [12], abych mohl objektivně srovnat uživatelskou zkušenost s rozhraním knihovny Geovisto před vylepšeními a po přidání vylepšení a dostal tak validní zhodnocení mé práce.

První metrikou byla míra úspěšnosti. Jedná se o metriku, která sleduje, zda byl uživatel úkol schopný dokončit. Vytvořil jsem tři kategorie – selhání, úspěch s dopomocí, úspěch. První kategorií jsem práci uživatele ohodnotil, pokud nebyl i s mojí dopomocí schopen úkol dokončit a požadované informace najít. Druhá kategorie označuje situaci, kdy se uživatel k informacím nakonec dostal, ale potřeboval moji pomoc, například s vysvětlením, kde najde jednotky číselných hodnot. Poslední kategorii jsem aplikoval na situace, kdy uživatel sám, bez jakéhokoliv mého zásahu získal a zpracoval požadované informace.

Další použitá metrika byla časové měření. Po zadání úkolu uživateli a ověření, zda úkolu rozumí, jsem nechal uživatele pracovat a měřil čas od začátku jeho práce až do konce. Tato metrika mi poskytla objektivní zhodnocení, zda se zlepšila efektivita práce uživatelů s uživatelským rozhraním.

Poslední metrika bylo slovní ohodnocení celé aplikace. Vzhledem k tomu, že uživatelská skupina pro druhé testování se ve složení nezměnila, poskytlo mi to subjektivní zhodnocení úprav, které jsem v knihovně implementoval.

Výstupem testování byly identifikované problémy v uživatelském rozhraní. Dále jsem také uživatelům dal prostor, aby sami mohli navrhnout řešení daných problémů, které by jim přišlo dobré. Sbíral jsem též podněty pro obsah webové prezentace celé knihovny, primárně které části potřebují nejvíce zdokumentovat.

### 5.3.2 Identifikované problémy uživatelského rozhraní

Jak bylo již zmíněno, hlavním požadovaným výstupem analýzy je identifikování problémů uživatelského rozhraní. Poslouží pak k návrhu možných řešení, jejich výběru a implementaci.

#### Absence metadat

Jak už bylo nastíněno v sekci 5.2.1, kde jsem tento problém odhalil sám při vytváření testovacích pohledů, v knihovně chybí funkce, pomocí které by šlo k mapě připojit metadata.

Uživatelé si stěžovali, že nevědí, kde informace v mapě hledat, protože ta zobrazovala pouze číselné hodnoty bez jakéhokoliv dalšího kontextu. Nevěděli proto, o jaké hodnoty se jedná, neznali jejich jednotky a význam. Dočasné řešení, kdy byly tyto informace alespoň ve zkratce zobrazeny nad mapou, měli problém najít, protože bylo mimo plochu mapy. Kromě absence těchto základních informací jim chyběl nějaký delší text, který by datový set popisoval podrobněji a obsahoval například jejich zdroj.

### **Absence legendy**

Dalším problémem, který identifikovalo všech pět uživatelů, byla absence legendy k vizualizovaným datům. Výrazně jim to podle jejich vlastních slov stěžovalo ve vrstvě kartogramu a vrstvě značek orientaci. Neznali význam barev, jejichž skupina tvořila rozmezí, do kterého hodnota patří, u kartogramu a kategorie dat u vrstvy značek. Tato rozmezí a příslušnost barvy ke kategorii museli tedy zjišťovat manuálně proklikáváním položek na mapě, což značně zpomalovalo vykonávání úkolů a vedlo k frustraci uživatelů.

### **Jednotky a zaokrouhlování hodnot**

V tomto případě se jednalo o absenci jednotek u číselných hodnot při nahlížení na například konkrétní polygon na mapě. Uživatelé sice měli jednotky k dispozici v již zmíněném dočasném řešení zobrazování metadat nad mapou, ale v průběhu plnění úkolů na ně zapomínali a museli je znovu zjišťovat. Absence zaokrouhlování číselných hodnot potom vedla k situacím, kdy uživatelé měli číst zbytečně dlouhá číselná údaje, které měly mnoho desetinných míst, i když zde nebyly pro požadovanou informaci vůbec relevantní a spíše překážely.

### **Absence nápovědy či dokumentace**

Knihovna Geovisto v době psaní této sekce nemá žádnou uživatelskou či technickou dokumentaci. Častou výtkou uživatelů bylo, že neexistuje návod na to, jak s uživatelským rozhraním pracovat. Pokud uživatel narazil na nějakou překážku, neměl nástroj, na který by se obrátil pro pomoc. Uživatelé by též ocenili, kdyby se v rozhraní vyskytovaly u jeho ovládacích prvků ikony, které by rozkryly krátkou nápovědu.

### **Problémy konkrétních testovaných vrstev**

Výtky se též týkaly jednotlivých vrstev. U vrstvy značek se jednalo hlavně o jejich seskupování a rozdělování při různých úrovních přiblížení mapy. Seskupování často nedávalo z geografického hlediska smysl, například značky pro země na okrajích kontinentů se seskupovaly s značkami ze sousedícího kontinentu. Uživatelé by též ocenili možnost aktuální stav seskupování uzamknout.

U vrstvy kartogramu uživatelům vadil barevný kontrast, respektive jeho nedostatek. V aktuální implementaci se seskládá z maximálně sedmi odstínů jedné barvy. To značně rozšiřuje škálu hodnot, které daný odstín reprezentuje. To stěžovalo převážně hledání velmi nízkých, či velmi vysokých hodnot. Možné řešení bylo navrženo jedním uživatelem, a to v podobě rozšíření kontrastů o více barev, což odpovídá doporučením pro návrh uživatelských rozhraní v podsekcí [3.2.4](#).

## Kapitola 6

# Návrh řešení

Z výsledků analýzy popsané v předchozí kapitole uvedených v sekci 5.3.2 jsem vybral ty požadavky, které mají největší dopad na kvalitu uživatelského rozhraní knihovny Geovisto a následně jsem vytvořil návrh jejich řešení, která budou implementována. Všechny problémy reálně nebylo možné v kontextu rozsahu této práce řešit a implementovat. V této kapitole budou uvedeny jednotlivé návrhy řešení pro každý zvolený problém. Ty, které se v této kapitole neobjeví, byly vyhodnoceny jako problémy, které dopad na uživatelskou zkušenost s uživatelským rozhráním mají minimální.

### 6.1 Legenda

Nejvíce zmiňovaným problémem byla absence legendy. Nejdříve bylo potřeba vyřešit její umístění. Vzhledem k tomu, že aplikace s mapou se dá vkládat do webů třetích stran, je vhodné, aby se legenda nacházela přímo na ploše mapy. Nabízela se ještě možnost ji umístit do rozbalovacích karet postranního ovládacího panelu, zde by ale mohla být pro uživatele těžko dohledatelná. Její výsledné umístění jsem zvolil přímo na ploše mapy v pravém dolním rohu. Zde legenda nebude rušit pohled na informace prezentované mapou a jedná se o běžně používanou praxi.

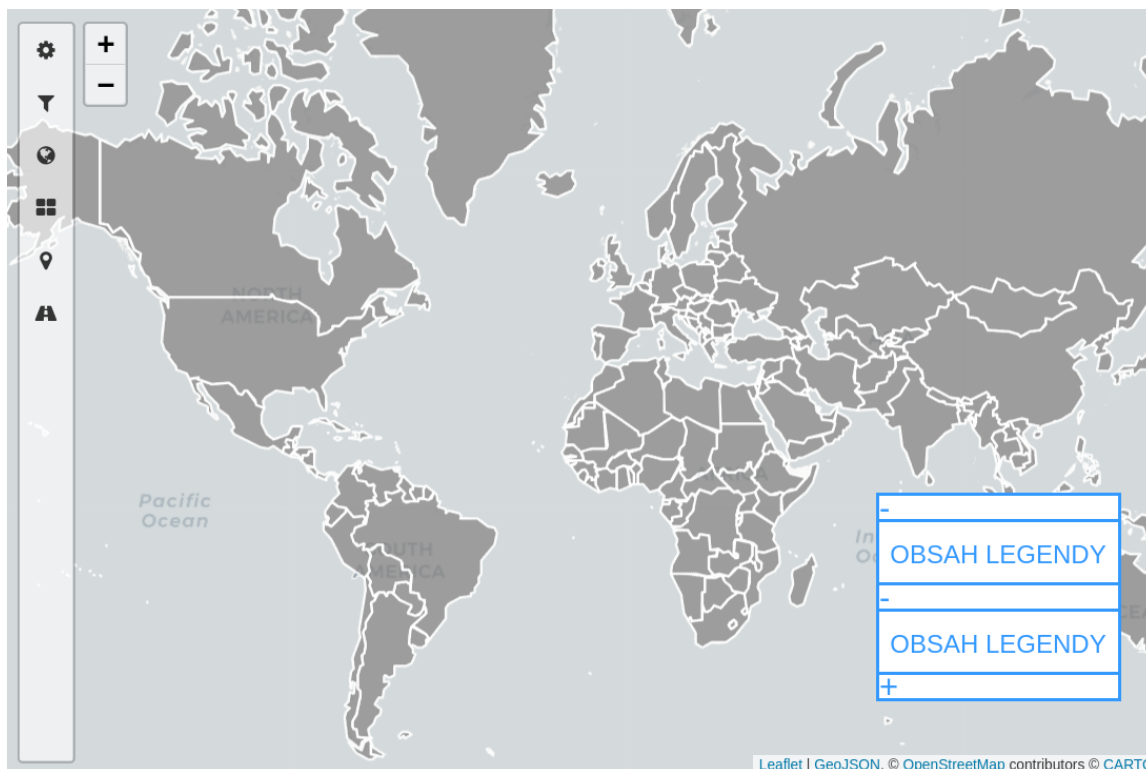
Jako problém se také ukázala potencionální situace, kdyby na mapě bylo aktivovaných zároveň více vrstev vyžadujících legendu, konkrétně jak více legend zobrazit. Návrh počítá, že samotné legendy budou obaleny do rozbalovacích prvků, které budou seskupeny do jednoho celku na mapě (obrázek 6.1).

Podoba samotných legend se liší pro každou vrstvu, v tomto případě je plánována implementace pro vrstvy kartogramu a vrstvu značek. Rozhodl jsem se, že tyto konkrétní podoby bude lepší řešit bez předchozího návrhu až při implementaci, z důvodu možných programátorských limitací a možnost vidět a měnit výsledek v reálném čase.

### 6.2 Metadata

V době vytváření návrhu řešení neexistovala v knihovně Geovisto funkce, která by umožnila zobrazení metadat, tedy dat, která obsahují informace o tematických datech používaných ve vizualizované tematické mapě. Zprvu jsem uvažoval o jejich umístění mimo plochu mapy, což bylo zavrženo opět kvůli možnosti vkládání aplikace do webů třetích stran.





Obrázek 6.1: Návrh zobrazení více legend zároveň. Návrh je umístěn v pravém dolním rohu obrázku a je od aktuálního stavu uživatelského rozhraní odlišený použitím neprůhledného bílého pozadí s modrou hranicí rozbalovacího prvku, který legendy sdružuje.

Jako řešení se tedy nabízelo vytvořit samostatné okno v postranním ovládacím panelu (obrázek 6.2). I v tomto případě jsem se rozhodl řešit konkrétní podobu vizualizace a formát vstupních dat, ve kterém by byla metadata knihovnou zpracována, až při implementaci.

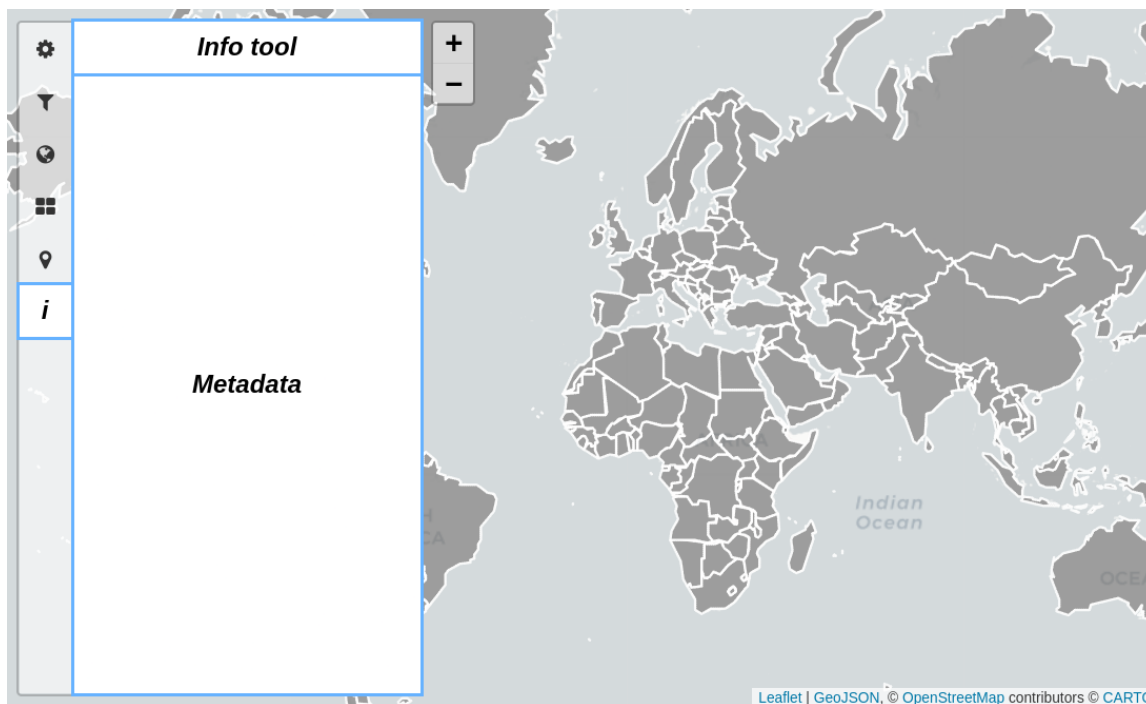
### 6.3 Jednotky a zaokrouhlování hodnot

Jednotky by bylo vhodné zobrazovat všude, kde se vyskytují jim náležící číselné hodnoty (obrázek 6.3). Vzhledem k faktu, že v aktuální implementaci v době psaní této práce nelze pro instanci knihovny Geovisto používat vícero sad tematických dat zároveň, lze nastavovat jednotky globálně pro všechny číselné hodnoty vizualizované v dané instanci. Ty bude možno nastavit v konfiguraci na libovolnou hodnotu.

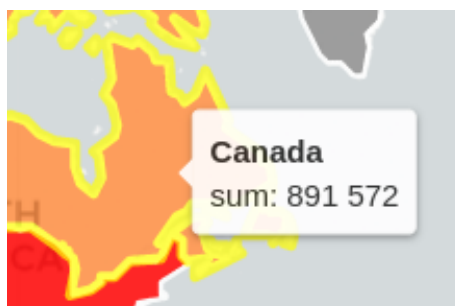
Zaokrouhlování bude též aplikováno globálně na všechny hodnoty v dané instanci. Nastavováno bude v konfiguraci pomocí čísla, které bude udávat, na kolik desetinných míst se má hodnota zaokrouhlit.

### 6.4 Dokumentace

V tomto případě bylo potřeba jako první rozhodnout, na koho dokumentace bude zaměřená, tedy jestli se bude soustředit na uživatele, který z vizualizovaných dat chce pouze získat informace, nebo na uživatele, který chce data vizualizovat. Rozdílní jsou v tom, že první typ



Obrázek 6.2: Návrh zobrazení metadat a jejich umístění v okně postranního panelu. Návrh je odlišený použitím neprůhledného bílého pozadí s modrou hranicí.



Obrázek 6.3: Příklad číselné hodnoty postrádající jednotky.

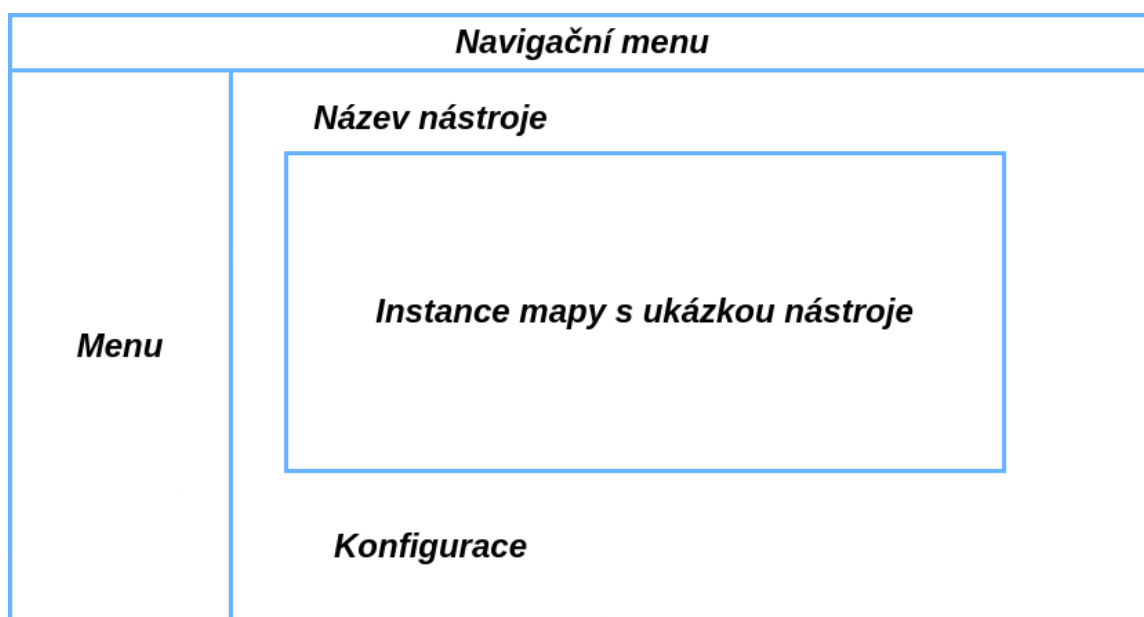
uživatelé pracuje převážně pouze se samotnou mapou a jejími funkcemi, zatímco druhý využívá i postranní ovládací panel, konfiguraci a další pokročilé funkce knihovny.

Pro každý z těchto dvou typů uživatelů je vhodná také jiná forma dokumentace. Pro první typ uživatelů je příhodnější vysvětlit v dokumentaci ovládací prvky knihovny s příloženými konkrétními příklady aplikace popisujícími dané situace, zatímco pro druhý je důležitější popsat například proces vytváření konfigurací, získávání datových sad nebo případně technická dokumentace popisující přímo zdrojový kód knihovny.

Protože se tato práce primárně zaměřuje na uživatelské rozhraní, rozhodl jsem se více soustředit na požadavky prvního typu uživatelů. Dokumentace bude mít formu webové prezentace, kde budou popsány jednotlivé vrstvy a nástroje. Kde to bude vhodné, připojím k danému tématu i přímou ukázkou dané problematiky formou vytvoření této situace v instanci knihovny Geovisto. Plánuji vypustit část s technickou dokumentací, ale zároveň například ke každému popisovanému nástroji přiložit příklad jeho konfigurace nebo ukázat

práci s postranním ovládacím panelem. Samotný web se bude dělit na uvítací stránku se základními informacemi o knihovně a výčtem aktuálně implementovaných a zveřejněných nástrojů. Dále bude mít sekci s dokumentací (obrázek 6.4), kde se bude nacházet popis jednotlivých nástrojů a vrstev.

Výběr konkrétní technologie na vytvoření webové stránky s prezentací knihovny Geovisto a návrh jejího rozložení opět provedu až při implementaci této stránky, protože požadavky na rozložení a technologii se mohou ještě změnit v průběhu implementace ostatních navržených řešení.



Obrázek 6.4: Návrh dokumentace jednotlivých vrstev a nástrojů knihovny Geovisto.

# Kapitola 7

## Implementace

Tato kapitola se zabývá implementací jednotlivých nástrojů popsaných v předchozí kapitole a nástroje pro získání tematických dat popsaného v podsekcí 5.2.2. Budou zde též prezentovány výsledné podoby těchto nástrojů.

### 7.1 Geovisto Data Miner

Pro implementaci nástroje Geovisto Data Miner na získávání sad tematických dat byl použit programovací jazyk Python, hlavně pro jeho snadnou práci s textovými řetězci a datovými API. Python má též širokou nabídku rozšiřujících externích knihoven, mezi nimi hlavně knihovna *pandaSDMX*<sup>1</sup>, která v tomto nástroji zajišťuje komunikaci s různými statistickými API<sup>2</sup> pomocí formátu SDMX<sup>3</sup>. Další zde použitou externí knihovnou je *pycountry*<sup>4</sup> pro standardizaci kódů států na standard ISO 3166-1 alpha-3, protože například statistické API Organizace spojených národů používá jiné standardy.

Před popsáním samotného programu je potřeba popsat jeho konfiguraci. Ta je ve formátu JSON a pomocí modulu *json* je načtena do programu v datové struktuře *dictionary*. Následně se v cyklu prochází jednotlivé bloky konfigurace, představující jednu sadu tematických dat, jak je ukázáno ve výpisu 7.1. Každý takový blok je předán metodě `Request` API knihovny *pandaSDMX*, která z něj vytvoří HTTP dotaz. Ten odešle na statistické API a vrátí jeho odpověď v datové struktuře *series*. Pro získání výsledných tematických dat ve formátu JSON je potřeba ještě převést datovou strukturu *series* na *dictionary*, což vykonává metoda `series_to_dict` modulu *conversion*, ve které také probíhá standardizace kódů států a přiřazení hodnot jejich klíčům.

Program využívá také pomocný modul *utils*, který implementuje výpis informací o běhu programu na standardní výstup, vytvoření JSON souboru s výstupními tematickými daty a zpracování argumentů.

### 7.2 Rozšíření knihovny Geovisto

Knihovna je v době psaní této práce implementovaná v programovacím jazyku TypeScript, a proto je také použitý pro její rozšíření. Knihovnu Geovisto můžeme podle podsekcí 4.4.3

<sup>1</sup><https://github.com/dr-leo/pandaSDMX>

<sup>2</sup><https://pandasdmx.readthedocs.io/en/v1.0/#pandasdmx-user-guide>

<sup>3</sup><https://sdmx.org/>

<sup>4</sup><https://github.com/flyingcircusio/pycountry>

```

1  "OECD_GDP": {
2    "output_name": "gdp_oecd.json",
3    "stat_database": "OECD",
4    "request": {
5      "resource_id": "SNA_TABLE1",
6      "key": "AUS+AUT+BEL+CAN+CHL+COL+CRI+CZE+DNK+EST+FIN+FRA+DEU+GRC+HUN+ISL+IRL+ISR+ITA+
7      JPN+KOR+LTU+LVA+LUX+MEX+NLD+NZL+NOR+POL+PRT+SVK+SVN+ESP+SWE+CHE+TUR+GBR+USA.B1_GA.CXC",
8      "params": {
9        "startTime": "2019",
10       "endTime": "current_year",
11       "dimensionAtObservation": "AllDimensions"
12     }
13   },
14   "pairs": {
15     "Country": 0,
16     "Value": "val",
17     "Year": 3
18   },
19   "single_years": "True"
}

```

Výpis 7.1: Úryvek konfigurace ve formátu JSON reprezentující dotaz na statistické API pro získání tematické sady. Objekt `request` obsahuje rozdělenou URL adresu dotazu. Objekt `pairs` popisuje na jaké klíče se mají získaná data v poli namapovat. Pole `single_years` nástroj nastaví tak, aby z více stejných záznamů, které se liší pouze datem, vybral ten nejnovější.

rozdělit na dvě části. První je její jádro reprezentované třídou `GeovistoMap`. Tato třída zajišťuje vykreslování samotné mapy a jejich nástrojů. Spravuje také jejich aktuální stav `state`. Nástroje, data a globální proměnné jsou uchovávány pomocí atributů mapy `props`.

Druhá část je reprezentována moduly oddělenými od jádra knihovny `Geovisto`. Tyto moduly implementují nástroje a vrstvy. Pomocí těchto modulů jsem také implementoval legendy, vizualizaci metadat a jednotky. Protože jsou tyto moduly od jádra oddělené, je potřeba exportovat instance tříd, které je implementují. Konstruktor těchto tříd potom obsahuje `props`, které udávají identifikátor dané vrstvy. Dále mají tyto třídy vlastní stav `state`, který obsahuje například dimenze pro danou vrstvu. Vrstvy mají též metody pro serializaci a deserializaci stavů, které jsou používány pro nastavování stavu pomocí konfiguračních souborů a nebo pro export aktuálního stavu do nové konfigurace. Pro případ, že není k dispozici konfigurační soubor, vrstvy pracují s výchozími hodnotami `defaults`. Implementace legend a zaokrouhlování metod vyžadovala krom vytvoření samostatných modulů i zásahy do jádra knihovny.

### 7.2.1 Implementace legend

Legendy implementuje třída `LegendTool`. Jedná se o nástroj knihovny `Geovisto`, implementuje proto rozhraní `IMapTool`. Nástroj je singleton, a počítá se tedy s tím, že bude k dispozici pouze maximálně jedna instance tohoto nástroje. Tato třída poskytuje rozhraní pro vytvoření HTML elementů, ve kterých je potom vykreslena samotná legenda. Obsah legendy je vytvořen mapovými vrstvami implementací rozhraní `IMapLegend`. Tento postup byl zvolen, protože při tvorbě nových nástrojů není potřeba takto zasahovat do třídy `LegendTool`. Místo toho se v metodě `createLegend` získají všechny nástroje a u

nich se následně kontroluje, zda implementují rozhraní `IMapLegend`, kterým jsem rozšířil jádro knihovny Geovisto a které deklaruje metody pro správu legend. Pokud ano, je pro daný nástroj či vrstvu vykreslena legenda ve formě přidání HTML elementu na mapu, který je vrácen metodou `getContent` třídy implementující rozhraní `IMapLegend`. Metoda `getContent` může též vrátit hodnotu `undefined`, která zabrání vykreslení legendy. Toto může být použito ve specifických případech, kdy vrstva například nemá ještě při inicializaci data na vykreslení.

Legendy dynamicky reagují na změny ve stavu mapy a každá jednotlivá legenda je vázaná na konkrétní nástroj či vrstvu. Pokud se změní dimenze vrstvy, legenda v reakci na tuto událost změní i svůj obsah podle nového stavu dimenzí. Reaguje též na změny ve viditelnosti vrstvy. Záleží na konkrétní implementaci, ale legenda se objeví až poté, co má k dispozici hodnoty v dostatečném počtu dimenzí pro smysluplné informování uživatele. Pokud je tento počet uživatelem po vykreslení legendy opět snížen, legenda reaguje chybovou hláškou. Zmizí pouze, pokud uživatel vypne zobrazování vrstvy, ke které je vázána.

Abych docílil chování popsaného v předchozím odstavci, musel jsem implementovat dva nové typy událostí v jádru knihovny Geovisto. První reagující na změny ve stavu dimenzí třídou `DimensionChangeEvent`, která se přidává do plánovače událostí `MapEventManager` v místě změn dimenzí. Druhou reagující na změny ve viditelnosti třídou `VisibilityChangeEvent`, která se přidává do plánovače událostí v třídě implementující okna postranního ovládacího panelu `SidebarTab`.

Jak již bylo řečeno, legendy jsou vykreslovány jako HTML elementy na mapu. V metodě `createLegend` se vykreslují všechny zároveň při každém volání této metody. Proto se i při uskutečnění událostí popsaných v předchozím odstavci znovu vykreslí i ty legendy, kterých se změna dimenzí či viditelnosti vrstvy netýkala.

Oproti původnímu návrhu, kdy legendy měly být umístěny do společného rozbalovacího prvku, jsou nakonec implementovány jako samostatné bloky s bílým, částečně průhledným pozadím. Původní návrh se ukázal jako nepraktický a esteticky nevhodný, protože ovládací části rodičovského prvku zabíraly příliš mnoho místa a jednotlivé obsahy legend mohou mít jiné formy a velikosti. Rozhraní `IMapLegend` jsem implementoval pro vrstvy kartogramu a značek. Výsledná implementace je zobrazena na obrázku 7.1.

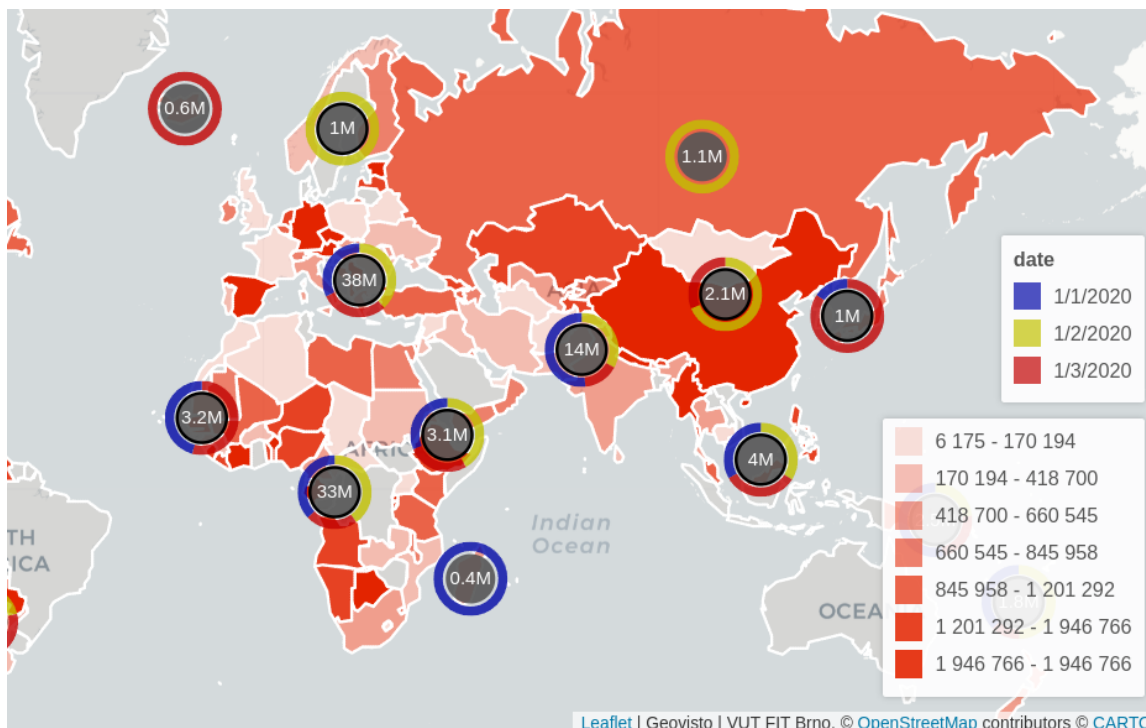
Legendy lze konfigurovat, kdy kromě položek společných pro všechny vrstvy a nástroje, mají specifickou položku `tools`, což je seznam identifikátorů vrstev, pro které chceme legendu zobrazit. Pořadí v tomto seznamu určuje, v jakém pořadí se budou jednotlivé legendy vykreslovat a tím zabráníme situaci, kdy legenda, která se vykreslila jako nejspodnější, po vypnutí a zapnutí viditelnosti její vrstvy, nebude vykreslena jako nejhornější. První v seznamu je vykreslený i jako první v mapě, a tudíž bude vždy nejspodnější.

## 7.2.2 Implementace vizualizace metadat

Po prozkoumání všech možných formátů, ve kterých by se metadata mohla vizualizovat, byl vybrán formát *Markdown*<sup>5</sup>. Jedná se o značkovací jazyk, který se často používá pro psaní dokumentací a také právě pro psaní obsahu webových stránek. Představuje snadný způsob, kterým uživatel může zapisovat metadata a upravovat jejich struktury pomocí značek, který Markdown poskytuje. Uživatel může vytvářet nadpisy, používat dekorace textu nebo vkládat obrázky. Práci mi výrazně ulehčil balíček *markdown-it*<sup>6</sup>, který data ve formátu Markdown převede do HTML.

<sup>5</sup><https://www.markdownguide.org/>

<sup>6</sup><https://github.com/markdown-it/markdown-it>



Obrázek 7.1: Finální implementace legend, zobrazené pro vrstvu kartogramu a značek zároveň. Legendy se vykreslují do pravého rohu odspodu.

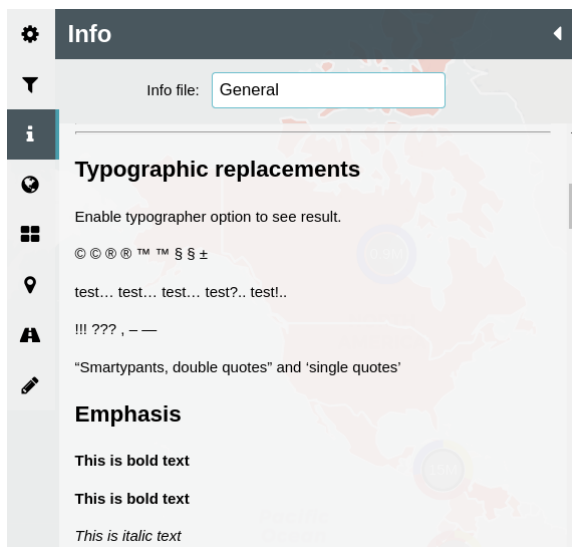
Metadata jsou podle návrhu zobrazena v samostatném okně postranního ovládacího panelu. Základ implementace tvoří třída `InfoTool`. V té se pomocí metody `createMapForm` vytvoří nová instance třídy `InfoToolMapForm`, která obsahuje implementaci okna pro vizualizaci metadat. V této třídě se metodou `getContent` vstupní data v podobě Markdown souboru načtou instancí třídy `InfoDataManager`. Ta se uloží do stavu nástroje `state`. Poté je vytvořena funkce `changeInfoData`, která reaguje na změny v vstupním poli pro výběr zdrojových Markdown souborů. V té se také pomocí balíčku `markdown-it`<sup>7</sup> převedou data do HTML a jsou umístěna do rodičovského prvku v postranním ovládacím panelu. Výsledná implementace je na obrázku 7.2.

Nástroj pro vizualizaci metadat `InfoTool` má konfiguraci, ve které kromě položek stejných pro všechny nástroje se konfiguruje pole `defaultFile`, které určuje výchozí soubor s metadaty. Ten bude v nástroji po jeho vytvoření zobrazen. Pokud toto pole neodpovídá názvu metadat definovaných v `props` nebo není v konfiguraci přítomné, výchozí soubor bude ten, který se načel jako první. Soubory s metadaty se do mapy přidávají pomocí `props` při vytváření instance nástroje `InfoTool` metodou `getInfoDataFactory().markdown()`.

### 7.2.3 Implementace jednotek a zaokrouhlování hodnot

Na rozdíl od řešení popisovaných v předchozích dvou sekcích se zde nejedná o samostatný nástroj, ale o úpravy v jádru Geovista a již existujících vrstvách kartogramu a značek. Změny v jádře se týkaly přidání metody `roundValues` do třídy `AbstractLayerTool`, která poskytuje všem vrstvám metodu na zaokrouhlování čísel. Druhá změna se týká třídy

<sup>7</sup><https://github.com/markdown-it/markdown-it>



Obrázek 7.2: Finální implementace nástroje pro vizualizaci metadat. Příklad obsahuje krátkou ukázkou typografických možností tohoto řešení. Horní vstupní pole určuje aktuálně zobrazovaný soubor. Pod ním se nachází vizualizace jeho obsahu.

MapLayerToolForm, kde je přidána metoda pro vytvoření obyčejného pole na textový vstup, které je instancí třídy LabeledTextInputForm.

Úpravy v již existujících nástrojích se týkaly přidání tří nových dimenzí `unit`, `unitDesc` a `unitEnabled`. První je pro vkládání krátké formy jednotek, druhá pro jejich delší, upřesňující formu a poslední je pro možnost jednotky v jednotlivých nástrojích zapínat a vypínat. Dimenze jsou specifické pro každou vrstvu a umožňují tedy nastavit každé jiné jednotky. Všechny tři dimenze lze ovládat přímo v uživatelském rozhraní skrze postranní ovládací panel, tak i v konfiguraci.

Zaokrouhlování hodnot je statické a specifikuje se v konfiguraci, zatímco jednotky dynamicky reagují na jejich změny pomocí ovládaní v uživatelském rozhraní. Obě funkce se aplikují na všechny místa ve vrstvách, kde se objevují číselné hodnoty, konkrétně ve vyskakovacích oknech při kliknutí na polygon nebo značku a v legendách. Výsledná implementace je na obrázku 7.3.

### 7.3 Webová dokumentace

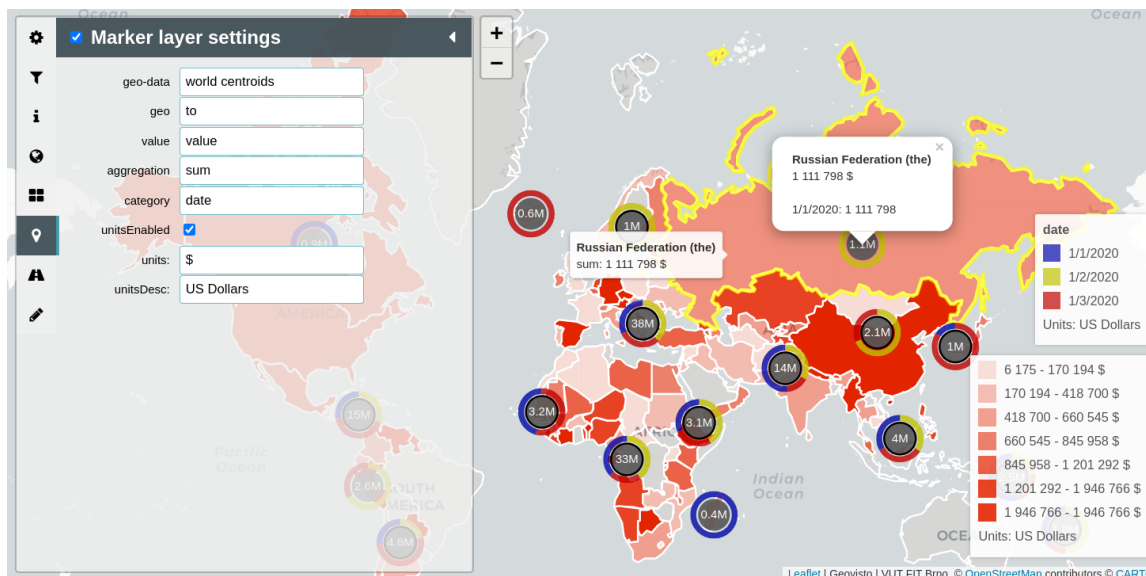
Po prozkoumání různých *frameworků* na tvoření webových stránek byl pro implementaci dokumentace vybrán *Docusaurus*<sup>8</sup>. Jedná se o framework určený na tvorbu dokumentací pomocí formátu Markdown. Jeho výhodou je, že je přímo určený na nasazení výsledné webové stránky do *GitHub Pages*<sup>9</sup>.

Výsledná implementace se skládá z úvodní stránky (obrázek 7.4), prezentace jednotlivých nástrojů a stránky, která obsahuje instanci knihovny se všemi jejími dostupnými funkcemi. Prezentace se skládají z jednotlivých souborů ve formátu Markdown obsahujících jejich popis a také instanci knihovny s daným, aktivovaným, předem nakonfigurovaným, nástrojem (obrázek 7.5). K příkladu je též přiložená jeho konfigurace. Vkládání instancí

<sup>8</sup><https://docusaurus.io/>

<sup>9</sup><https://pages.github.com/>





Obrázek 7.3: Finální implementace jednotek. Na příkladu je vidět nastavení dimenzí v postranním ovládacím panelu a místa jejich vykreslení.

umožňuje formát *MDX*<sup>10</sup>, který poskytuje možnost vkládat do klasického Markdown formátu komponenty knihovny React. Ty slouží jako *wrapper* pro všechny instance knihovny Geovisto.



Obrázek 7.4: Úvodní stránka webové dokumentace knihovny Geovisto.

Pro umístění instance do dokumentačního Markdown souboru je nutno importovat komponentu, která tuto instanci obaluje, a tu následně přidat do lokálně definované komponenty. Těmto Markdown souborům je také potřeba nastavit pořadí v menu pomocí klíčového slova `sidebar_position`. Instance knihovny je vytvořena v samostatné třídě, která implementuje rozhraní knihovny React `Component`. V ní také probíhá dodání všech dat potřebných k vytvoření tematické mapy a zdefinování nástrojů, které budou ve výsledné mapě k dispozici.

<sup>10</sup><https://mdxjs.com/>

Geovisto Tutorial Playground

Intro  
Layers  
  Introducion  
  Choropleth  
  Marker  
  Connection

Layers > Choropleth

## Choropleth

A choropleth map is a type of statistical thematic map that uses intensity of color to correspond with an aggregate summary of a geographic characteristic within spatial enumeration units, such as population density.<sup>1</sup>  
Example below contains data about **Gross domestic product**<sup>2</sup> in countries that are members of **OECD**<sup>3</sup>.

### Example

21 717 076 - 63 951 115 \$
63 951 115 - 228 539 245 \$
228 539 245 - 356 084 868 \$
356 084 868 - 541 063 604 \$
541 063 604 - 1 268 868 332 \$
1 268 868 332 - 2 630 317 731 \$
2 630 317 731 - 2 630 317 731 \$

Units: US Dollars (Millions)

Leaflet | Geovisto | VUT FIT Brno, © OpenStreetMap contributors © CARTO

Obrázek 7.5: Implementace prezentace jednotlivých nástrojů knihovny Geovisto. Zde je popsána vrstva kartogramu.

## Kapitola 8

# Testování

V této kapitole popíši druhé uživatelské testování mnou implementovaných rozšíření knihovny Geovisto a výsledky, které z něj vyplynou. Poznatky dosažené v tomto testování budou přímo srovnávány se závěry prvního uživatelského testování popsaného v sekci 5.3. Toto srovnání bude nejdůležitější pro posouzení, zda se uživatelská zkušenost s knihovnou Geovisto zlepšila.

### 8.1 Druhé uživatelské testování

Skupina uživatelů, která se testování účastnila, se skládala ze stejných uživatelů jako v prvním testování. Skupinu jsem neobměnil, protože bylo potřeba získat srovnání stavu uživatelského rozhraní před a po provedení změn k dosažení validního vyhodnocení jejich přínosu. Stejně uživatele vyžadovaly i použité metriky popsané v podsekcí 5.3.1.

Samotné testování mělo opět formu rozhovorů s plněním úkolů a stejnými testovacími případy užití. Jednotlivé testovací pohledy byly tentokrát už součástí webové dokumentace. Úkoly se od prvního testování nezměnily, jejich počet ale byl snížen z 11 na 4, protože se změnil i účel uživatelského testování. Nyní nebylo záměrem identifikovat nedostatky rozhraní, ale vyhodnotit přínos mnou implementovaných rozšíření. Z toho důvodu většina otázek použitých v prvním testování přestala být relevantní.

### 8.2 Výsledek

Pro objektivní srovnání zvýšení použitelnosti uživatelského rozhraní knihovny Geovisto jsem opět měřil čas, který uživatelé strávili vykonáváním jednotlivých úkolů. Tabulka 8.1 poskytuje srovnání průměrných časů potřebných na vykonání úkolu. První úkol měl za cíl otestovat použitelnost vrstvy značek a druhý přínos jednotek. Třetí úkol se zaměřil na vrstvu kartogramu a přínos legend. Poslední úkol testoval kombinaci vrstvy kartogramu a spojení. Nástroj na zobrazování metadat byl používán ve všech jmenovaných úkolech.<sup>1</sup>

Z tabulky 8.1 lze tedy usoudit, že došlo ke značnému zrychlení vyhledávání jednotek, které byly původně umístěny jako provizorní řešení nad mapou, nyní umístěných v legendách, nástroji na zobrazování metadat a u jednotlivých číselných hodnot. Dále také došlo ke zrychlení orientace ve vrstvě kartogramu a značek, převážně díky legendám.

---

<sup>1</sup>Tyto úkoly v prvním uživatelském testování neověřovaly přínos mnou implementovaných řešení, ale identifikovaly problémy, které moje vylepšení řeší.

Případy užití	První uživatelské testování (sekundy)	Druhé uživatelské testování (sekundy)
1. úkol	58	25
2. úkol	17	3
3. úkol	60	22
4. úkol	31	25

Tabulka 8.1: text

Tyto výsledky je ovšem nutno brát s mírnou rezervou, vzhledem k tomu, že uživatelé byli stejní v obou testováních, a tudíž už jednou uživatelské rozhraní knihovny Geovisto používali. Odstup mezi testováními byl ale 4 měsíce a jak se i při samotném testování potvrdilo, uživatelé už většinou získané dovednosti z předchozího testování ztratili.

Kromě srovnávání času, který uživatelé strávili vykonáváním úkolů, lze vyvodit závěry i z jejich subjektivního hodnocení rozdílu v uživatelské zkušenosti. Všichni uživatelé uvedli, že se jim s uživatelským rozhraním po implementování změn pracovalo lépe. Konkrétně označili legendy za největší přínos, protože například ve vrstvě kartogramu umožňovaly okamžitě přiřadit barvu k hledané hodnotě a pomohly vymezit minimální a maximální hodnotu. Ve vrstvě značek si zase hned spojili barvu s kategorií, kterou v donutovém grafu tvořící značku reprezentovala. U legend též ocenili, že se v nich zobrazují jednotky.

Nástroj pro zobrazování metadat jim zase poskytl kontext k datům, která byla na mapě vizualizovaná, což jim podle jejich slov výrazně chybělo v prvním testování. Pomohlo to také ke zrychlení práce s mapou, protože uživatelé si rychleji uvědomili význam dat. Objevovale se zde často výtka, že ikona v postranním ovládacím panelu zobrazující okno s metadaty nebyla intuitivní a uživatelé si ji všimli až po krátkém čase stráveném prací s mapou. Jednotky stejně jako metadata pomáhaly dodávat vizualizovaným datům kontext. Zaokrouhlování desetinných míst zjednodušovalo čtení číselných hodnot. Uživatelé také ocenili webovou dokumentaci a způsob, kterým reprezentuje jednotlivé nástroje (obrázek 7.5). Pomáhá uživatelům pochopit, jak pracovat s daným typem mapové vrstvy a jakým způsobem z ní číst data.

U vrstvy kartogramu jsem také zaznamenal, že uživatelům se daleko snadněji pracovalo s kartogramem, když v něm byla používána modrá barva místo červené, jako v prvním testování. Někteří uživatelé dokonce nabyli dojmu, že se jednalo o mé vylepšení, protože se jim daleko lépe rozlišovaly odstíny modré barvy.

## Kapitola 9

# Závěr

Cílem této práce bylo zlepšit použitelnost uživatelského rozhraní knihovny Geovisto. Pro dosažení tohoto cíle jsem musel nastudovat problematiku zpracování a vizualizace geografických dat na webu a problematiku použitelnosti uživatelských rozhraní. Následně jsem k těmto problematikám musel prozkoumat již existující technologie, které se jimi zabývají. Jelikož se tato práce zabývá knihovnou Geovisto, bylo nutné ji důkladně prostudovat a naučit se s ní pracovat. Již při tomto procesu jsem identifikoval první nedostatky jejího uživatelského rozhraní. Pro identifikování co nejvíce problémů jsem provedl analýzu rozhraní knihovny pomocí uživatelského testování. K odhaleným problémům jsem následně navrhl jejich řešení, vybral z nich ty, které měly největší přínos pro zlepšení použitelnosti a implementoval je.

Výsledkem je uživatelské rozhraní rozšířené o legendy, nástroj pro zobrazení metadat ve formátu Markdown a veřejně dostupná webová stránka, která poskytuje návod na používání knihovny a jejích nástrojů, včetně příkladů použití. Dále je také rozhraní rozšířeno o jednotky pro číselné hodnoty a zaokrouhlování hodnot na libovolná desetinná místa. Tato rozšíření a webová dokumentace byly při testování označeny uživateli jako velké zlepšení v uživatelské zkušenosti při práci s knihovnou Geovisto. Dalším výsledkem je také Geovisto Data Miner, který umožňuje stahování tematických dat z veřejně dostupných statistických API a jejich standardizaci.

Budoucím možným pokračováním této práce může být řešení dalších problémů uživatelského rozhraní knihovny Geovisto identifikovaných v této práci, například vyřešení problémů v konkrétních mapových vrstvách nebo přidání nápovědy do ovládacích prvků rozhraní. Webovou dokumentaci lze rozšířit o část zaměřenou hlavně na programátory, tedy o technickou dokumentaci.

# Literatura

- [1] BEUTNER, R. *GIS Thematic Data*. 2022 [cit. 2-5-2022]. Dostupné z: [https://library.hws.edu/thematic\\_data](https://library.hws.edu/thematic_data).
- [2] BUTLER, H., DALY, M., DOYLE, A., GILLIES, S., HAGEN, S. et al. The geojson format. *Internet Engineering Task Force (IETF)*. 2016.
- [3] DENT, B., TORGUSON, J. a HODLER, T. *Thematic map design*. Sixth edition. McGraw-Hill New York, New York, NY, 2009. ISBN 978-0-07-294382-5.
- [4] GOOGLE. *Google Maps Platform Documentation*. 2022 [cit. 30-4-2022]. Dostupné z: <https://developers.google.com/maps/documentation>.
- [5] HYNEK, J., KACHLÍK, J. a RUSŇÁK, V. Geovisto: A Toolkit for Generic Geospatial Data Visualization. In: *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SciTePress - Science and Technology Publications, 2021, s. 101–111. Volume 3: IVAPP. DOI: 10.5220/0010260401010111. ISBN 978-989-758-488-6. Dostupné z: <https://www.fit.vut.cz/research/publication/12387>.
- [6] JOHNSON, J. *Designing with the mind in mind: simple guide to understanding user interface design guidelines*. First edition. Morgan Kaufmann, 2010. ISBN 978-0-12-375030-3.
- [7] KAŠPAR, P. *Webová aplikace pro vizualizaci hodnot v geografických souřadnicích*. Brno, CZ, 2021. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Dostupné z: <https://www.fit.vut.cz/study/thesis/23900/>.
- [8] LEAFLET. *Leaflet API reference*. 2022 [cit. 30-4-2022]. Dostupné z: <https://leafletjs.com/reference.html>.
- [9] MEEKS, E. *D3.js in Action*. First edition. Manning Shelter Island, NY, 2015. ISBN 978-1617292118.
- [10] MOZILLA. *Canvas API*. 2022 [cit. 30-4-2022]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API).
- [11] MOZILLA. *SVG: Scalable Vector Graphics*. 2022 [cit. 30-4-2022]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/SVG>.
- [12] NIELSEN, J. *Usability engineering*. Morgan Kaufmann, 1994. ISBN 1-12-518406-9.
- [13] PREECE, J., SHARP, H. a ROGERS, Y. *Interaction Design: Beyond Human-Computer Interaction*. Fourth edition. Wiley, 2015. ISBN 9781119066002.

- [14] SLOCUM, T. A., MCMASTER, R. B., KESSLER, F. C. a HOWARD, H. H. *Thematic cartography and geovisualization*. Third edition. CRC Press, 2014. ISBN 978-1-292-04067-7.
- [15] STONE, D., JARRETT, C., WOODROFFE, M. a MINOCHA, S. *User interface design and evaluation*. First. Elsevier, 2005. ISBN 9780120884360.
- [16] WALFORD, N. *Geographical data: characteristics and sources*. First edition. John Wiley & Sons, 2002. ISBN 978-0471970859.