



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

**INTERAKTIVNÍ NÁSTROJ PRO TVORBU VÍCE-MÉNĚ
SLEPÝCH MAP**

INTERACTIVE TOOL FOR CREATION OF MORE-OR-LESS BLIND MAPS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR BALAZY

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. ADAM HEROUT, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Balazy Petr**
Program: Informační technologie
Název: **Interaktivní nástroj pro tvorbu více-méně slepých map**
Interactive Tool for Creation of More-or-Less Blind Maps
Kategorie: Uživatelská rozhraní

Zadání:

1. Vyhledejte a popište relevantní případy užití slepých (a podobných) map.
2. Seznamte se s prací s Open Street Maps, vyhledejte a popište nástroje pro práci s mapami.
3. Prostudujte problematiku vytváření webových aplikací, zaměřte se na zobrazování map.
4. Vytvořte nástroje pro tvorbu více-méně slepých map z podkladů OSM. Zaměřte se na zpracování na serveru.
5. Navrhněte minimální uživatelské rozhraní pro tvorbu slepých map na přání uživatele.
6. Implementujte prototyp řešené webové aplikace, který umožňování testování a vývoj funkčnosti.
7. Testujte vytvářenou aplikaci s uživateli a iterativně ji vylepšujte.
8. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," in IEEE Pervasive Computing, vol. 7, no. 4, pp. 12-18, Oct.-Dec. 2008, doi: 10.1109/MPRV.2008.80.
- Steve Krug: Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, ISBN: 978-0321965516
- Steve Krug: Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability, ISBN: 978-0321657299
- Joel Marsh: UX for Beginners: A Crash Course in 100 Short Lessons, O'Reilly 2016
- Susan M. Weinschenk: 100 věcí, které by měl každý designér vědět o lidech, Computer Press, Brno 2012

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3, značné rozpracování bodů 4 až 6.

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

Vedoucí práce: **Herout Adam, prof. Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 27. října 2020

Abstrakt

Cílem této bakalářské práce je implementovat webovou aplikaci pro tvorbu více-méně slepých map z OpenStreetMap dat. Při implementaci byla použita sada nástrojů Mapnik, která je určena k tvorbě map a mapových dlaždic. Během implementace probíhalo několik kol testování, po kterých byla aplikace opravována, upravována a vylepšována. Vytvořená aplikace umožňuje uživateli výběr oblasti, výběr prvků mapy a jejich vlastností, stažení mapy ve vybraném datovém formátu a velikosti.

Abstract

The main goal of this bachelor thesis is to implement a web application for creating more-or-less blind maps from OpenStreetMap data. The Mapnik toolkit, which is designed for creating maps and map tiles, was used in the implementation. Several rounds of testing were conducted during the implementation, after which the application was patched, modified and improved. The created application allows the user to select an area, select map elements and their properties, download the map in the selected data format and size.

Klíčová slova

webová aplikace, více-méně slepé mapy, víceméně slepé mapy, OpenStreetMap, Mapnik, React, Flask

Keywords

web application, more-or-less blind maps, more or less blind maps, OpenStreetMap, Mapnik, React, Flask

Citace

BALAZY, Petr. *Interaktivní nástroj pro tvorbu více-méně slepých map*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

Interaktivní nástroj pro tvorbu více-méně slepých map

Prohlášení

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

.....

Petr Balazy
19. května 2021

Poděkování

Tímto bych rád poděkoval panu prof. Ing. Adamu Heroutovi, Ph.D, za vedení této práce, cenné rady a trpělivost. Dále bych chtěl poděkovat rodině a přátelům za podporu při celém studiu.

Obsah

1	Úvod	3
2	Mapy	4
2.1	Styl mapy	4
2.2	Mapová projekce	4
2.3	Online mapy	5
2.4	Existující nástroje pro tvorbu vlastních map	7
3	Projekt OpenStreetMap	11
3.1	Vznik projektu	11
3.2	Data a datové formáty	12
3.3	Jak přispívat	15
3.4	Alternativy	15
4	Využité technologie	17
4.1	React	17
4.2	Flask	18
4.3	WebSocket	18
4.4	Celery	18
4.5	PostgreSQL	19
4.6	Mapnik	19
5	Návrh	22
5.1	Analýza	22
5.2	Návrh uživatelského rozhraní	24
5.3	Návrh pro zpracování na serveru	28
6	Implementace	30
6.1	Cíle implementace	30
6.2	Popis zpracování	30
7	Průběh testování a zhodnocení výsledků	36
7.1	Testování	36
7.2	Zhodnocení testování	38
8	Závěr	40
	Literatura	41

Kapitola 1

Úvod

Slepé mapy zná asi většina už ze základní školy. Tam jsme se s nimi mohli poprvé seznámit už na prvním stupni, když jsme se učili o rozdělení České republiky do krajů. Na druhém stupni už se mohly vyskytovat složitější slepé mapy s hranicemi České republiky doplněné např. o řeky, města, pohoří nebo národní parky. V rámci Evropy potom se slepou mapou států nebo hlavních měst.

Slepou mapu můžeme tedy brát jako mapu určenou primárně ke vzdělávání, na které je zobrazen pouze jeden hlavní prvek (města, řeky, atd), případně doplněný o politické hranice (např. kontinentů, států, regionů).

Pod pojmem více-méně slepé mapy (tento pojem byl zavedený v této práci), můžeme chápat všechny mapy, které vypadají jako slepé mapy, ale jsou rozšířené o další prvky mapy, informace i využití. Ve více-méně slepých mapách můžeme zobrazovat více různých prvků mapy a klidně i text, tyto mapy, tak tvoří spojnicí mezi slepými mapami a těmi klasickými. Takovéto mapy pak mohou sloužit opět k výuce, prezentaci nebo vizualizaci dat. Reálnými příklady použití více-méně slepých map, může být například mapa¹ zobrazující české železniční koridory s velkými městy, mapa² lesů s obcemi s rozšířenou působností nebo mapa³ dálniční sítě a městy.

Cílem práce je navrhnout a vytvořit webovou aplikaci, která umožní uživateli vytvářet více-méně slepé mapy, z OpenStreetMap dat, v různých velikostech a rastrových i vektorových formátech.

Kapitola 2 čtenáře seznámí se základními pojmy týkajícími se map, užitečných pro tuto práci. Popíše jak fungují online mapy rastrové i vektorové a na závěr uvede některé nástroje pro práci s geologickými daty a tvorbu vlastních map. Kapitola 3 poskytuje popis vzniku projektu OpenStreetMap, jeho data, datových formátů a jak se do projektu také zapojit. Kapitola 4 obsahuje popis a zdůvodnění hlavních technologií, které byly při vytváření webové aplikace použity. V kapitole 5 je popsána analýza potřeb uživatelů, návrh uživatelského rozhraní a serveru. Kapitola 6 rekapituluje cíle implementace, popisuje zajímavé a důležité části již implementované webové aplikace. Kapitola 7 líčí průběh, cíle, výsledky, změny všech kol testování a v závěru této kapitoly se nachází zhodnocení celého testování.

¹<https://www.spravazeleznic.cz/documents/50004227/50167205/koridory-zjednodusene.pdf>

²http://www.uhul.cz/images/prehl_map_CR/ORP_les%20resize1.jpg

³<https://www.podalnici.cz/dalnicni-sit-v-cesku/>

Kapitola 2

Mapy

Tato kapitola se věnuje stručnému popisu některých pojmů souvisejících s mapami, tak aby s nimi byl čtenář obeznámen ve zbytku práce, a také popisuje některé aplikace pro tvorbu map a jejich uživatelské rozhraní.

Na začátku této kapitoly se čtenář seznámí s pojmem styl mapy, poté co je to mapová projekce a Mercatorovo zobrazení, následuje sekce s popisem online map a v závěru této kapitoly jsou uvedeny a popsány nástroje pro tvorbu map a stylů.

2.1 Styl mapy

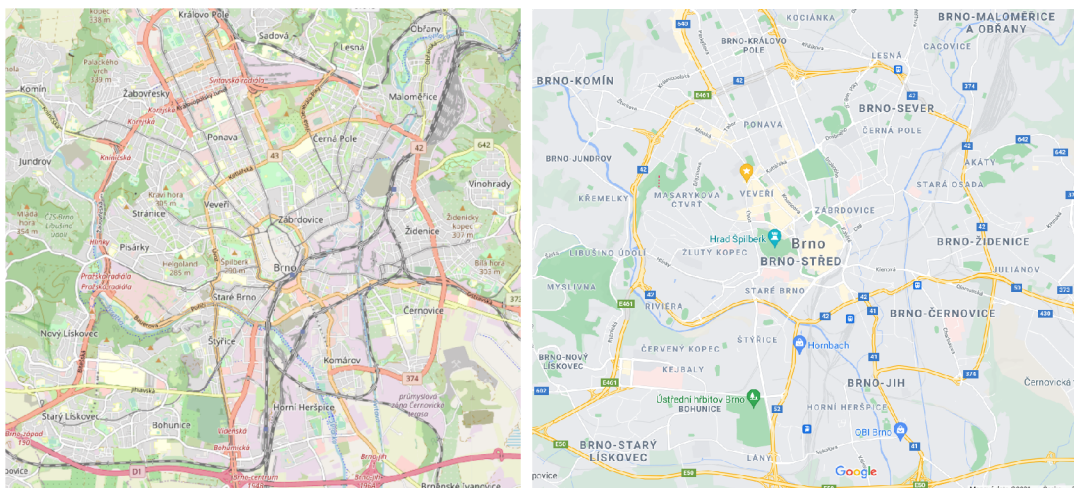
V moderní kartografii můžeme styl definovat jako volný rámec, který organizuje všechny kartografické výrazové prostředky, je využíván a lze jej přiřadit k určitému regionu, kartografické edici nebo tvůrci. Tvůrce mapy je také ten, kdo určuje, jaké prvky mapy (fyzicko-grafické, socioekonomické) se v mapě objeví a jak. Při takovémto výběru by se mělo dbát, aby vizualizované prvky snadno předávaly svou informační hodnotu cílové skupině. Stylem by se mělo nazývat takový soubor parametrů, který byl použit k vytvoření alespoň několika map, jinak lze tvrdit, že využití grafických atributů je náhodné. [1]

Pro stejná data mohou vznikat mapy s různými styly. Tvůrci těchto map, jsou často ovlivněni například osobními názory, místem původu nebo kulturou. Různý přístup můžeme vidět i na obrázku 2.1. Na něm jsou dva výřezy stejných oblastí, s podobným přiblížením. První je z Google maps a druhý z OpenStreetMap, ačkoliv se jedná o obecné mapy, tak se liší ve vzhledu, ale i v zobrazeném obsahu.

2.2 Mapová projekce

Mapová projekce je způsob přenesení bodů z trojrozměrného zemského povrchu na dvou-rozměrný rovinný povrch (mapu). Zemský povrch může být zobrazen na jednoduchou geometrickou plochu, jako je rovina, kužel nebo válec. Tato zobrazení, ale nejsou dokonalá. Zobrazení kulovitého povrchu na rovinný, způsobuje vždy zkreslení délky, úhlu nebo plochy. Většina projekcí zachovává jednu z těchto vlastností, ale zkresluje ty ostatní. Při zobrazování malých území, je zkreslení většinou minimální, ale pro větší oblasti, už je potřeba vybírat to nejvhodnější podle účelu mapy. [5, 7] Podle zkreslení můžeme zobrazení rozdělit na:

- délkojevná (ekvidistantní) – zobrazení, která zachovávají některé vzdálenosti, podle určitého systému čar



Obrázek 2.1: Výřezy mapy města Brna z OpenStreetMap a Google maps.

- plochojevná (ekvivalentní) – zobrazení, která zachovávají poměry ploch, ale zkreslují úhly
- úhlojevná (konformní) – zobrazení, která zachovávají úhly, ale zkreslují plochy
- vyrovnávací (kompenzační) – zobrazení, která zkreslují úhly i plochy, ale v menší míře, jedná se kompromis

Jedno z nejpoužívanějších zobrazení je **Mercatorovo zobrazení**. Návrh vznikl už v 16. století, primárně pro potřeby námořní přepravy a je připisován vlámskému kartografovi Gerhardu Mercatorovi, podle kterého je toto zobrazení i pojmenováno. Jedná se o zobrazení válcové a úhlojevné. Výhodou tohoto zobrazení je zachování úhlu, nevýhodou pak, zkreslení plochy, které jsou čím dále od rovníku, tím větší. [6] Tohoto jevu si lze všimnout na obrázku mapy 2.2 s tímto zobrazením. Grónsko vypadá větší než celá Evropa, ale přitom je Grónsko skoro 5 krát menší. Mercatorovo zobrazení se používá například pro navigační mapy nebo online mapy, jako např. Google maps, Mapbox, Bing mapy nebo OpenStreetMap mapy.

2.3 Online mapy

Tato sekce čtenáři přibližuje způsob fungování online map, podle druhu grafického zpracování dlaždic. Jako hlavní zdroj informací celé této sekce slouží diplomová práce [13]. V podsekcí „Rastrové dlaždice“ jsou pak navíc čerpány informace z článku [16] a v další podsekcí „Vektorové dlaždice“ pak z online článku [3].

2.3.1 Rastrové dlaždice

S rozdělením mapy do rastrových dlaždic (čtvercové nebo obdélníkové obrázky), jak je známe dnes, přišla poprvé firma Google. V roce 2005 představila své mapy s technologií tzv. „Slippy Maps“.

Tyto mapy byly předem rozděleny na části a nástrojem vygenerovány podle Google map stylu do dlaždic. Konvence Google map, byla mít všechny dlaždice čtvercové a stejně

²<https://creativecommons.org/licenses/by-sa/3.0>



Obrázek 2.2: Mapa s Mercatorovým zobrazením. Převzato z [4] pod licencí CC BY-SA 3.0².

velké v rozměrech 256×256 pixelů. Při tomto rozdělení do dlaždic se používá Mercatorovo webového zobrazení, jelikož se mapa dá jednoduše rozdělit právě do čtverců. Vygenerované dlaždice byly uloženy na serveru, v adresářích, podle úrovně přiblížení. Složky obsahovaly: pro 0. úroveň přiblížení 1 dlaždici, pro 1. úroveň 4, pro 2. úroveň 16, až po úroveň 24. kterou tvořilo 4^{24} dlaždic.

Takto vygenerované dlaždice, poté poskytuje mapový server uživatelům. Uživateli, který si prohlíží mapu se stahují a zobrazují v prohlížeči, pouze ty dlaždice, které mají jít v ten moment vidět. O správné sestavení a zobrazování se starají aplikace nebo JavaScriptové funkce v prohlížeči. Pokud se uživatel v mapách posune nebo změní úroveň přiblížení, stáhnou se další dlaždice, které vyplní vzniklé místo.

Postupně na využití technologie „Slippy Maps“ přešli i další poskytovatelé online map, jako např. OpenStreetMaps nebo Bing. Počet úrovní přiblížení, styl mapy a další různé specifikace, si každý provozovatel map určuje sám.

Rastrové dlaždice jsou stále používány např. OpenStreetMap nebo mapy.cz. Dlaždice si jde prohlížet i samostatně například přes OpenStreetMap dlaždicový server. Zde se např. zobrazí dlaždici pro 0 úroveň. V odkazu³, lze měnit čísla a zobrazovat si tak různé dlaždice.

2.3.2 Vektorové dlaždice

Technologii vektorových dlaždic poprvé představila firma Google, v roce 2010, ve své mobilní aplikaci Google Maps a o tři roky později i ve webové verzi. Motivací k vytvoření této technologie bylo zvýšení kvality map a nevýhody, které mají rastrové dlaždice, jako např. pomalé načítání dlaždic, nutnosti vytvářet úplně celý set dlaždic při jakékoliv změně v mapovém stylu a velikost dat, které na serveru zabírají.

Proces vytváření vektorových dlaždic je podobný jako u rastrových. Opět je potřeba nástroj a předpis, ten ale na rozdíl od předpisu používaného u tvorby rastrových dlaždic, neobsahuje informace, jak má co vypadat. Vytvořené vektorové dlaždice, také netvoří hro-

³<https://tile.openstreetmap.org/0/0/0.png>

mady souborů, ale jenom jeden. Ten obsahuje objekty (body, linie, polygony, ...) nebo jen části objektů, pro každou dlaždici. Ke klientovi se pak nepřenáší ze serveru žádné obrázky, ale právě jenom tyto vektorové objekty. Přenesené objekty pak vykreslí funkce, jedné z JavaScriptových knihoven k tomu určených (např. Leaflet GL nebo Mapbox GL JS), podle předpisu, který se nachází na straně klienta. Tento způsob vykreslování vyžaduje výkonnější hardware na straně klienta, než při používání rastrových dlaždic.

Vektorové dlaždice využívají například: již zmíněné Google maps, Maptiler nebo Mapbox.

2.4 Existující nástroje pro tvorbu vlastních map

Tato sekce se zabývá existujícími webovým i počítačovými aplikacemi, ve kterých by uživatel mohl vytvářet více-méně slepé mapy. V sekci jsou popisovány vlastnosti jednotlivých nástrojů, jejich výhody a nevýhody, ale také prvky uživatelského rozhraní, z jejichž výhod se čerpá v kapitole 5 návrh.

2.4.1 Počítačové aplikace

Tato sekce se věnuje počítačovým aplikacím pro tvorbu map a práci s geografickými daty. Aplikaci si uživatel musí vždy stáhnout i s daty ve vhodném formátu, ze kterých bude aplikace mapu vytvářet. Uživatelé jsou často limitováni výkonem jejich počítače, převážně pak velikostí paměti RAM, do které aplikace načítají veškerá data, se kterými pracují.

map-stylizer

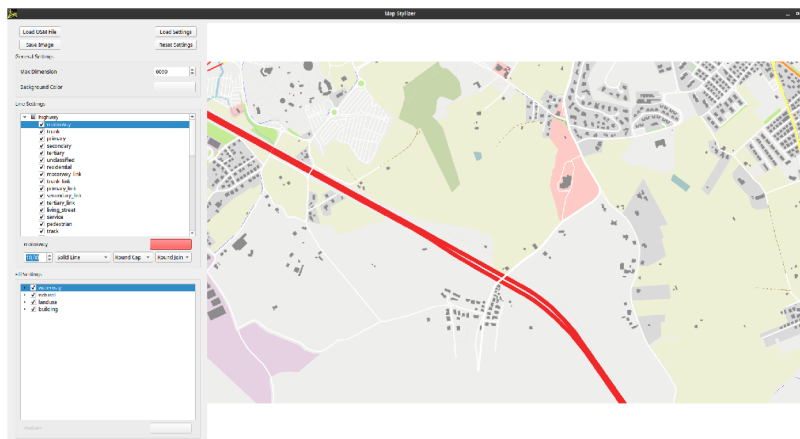
Map-stylizer [9] je velice jednoduchá počítačová aplikace, napsána v programovacím jazyce Python. Umí vytvářet mapy pouze ze souborů s daty, ve formátu OSM XML (tento formát je popsán v sekci 3.2). Aplikace obsahuje jednoduché uživatelské rozhraní (snímek tohoto rozhraní na obrázku 2.3) pro editaci a zobrazení mapy. Na mapu jde vykreslit celá řada prvků (chybějící jdou snadno doplnit manuálně v konfiguraci) kliknutím na zaškrtačací pole a těmto vykresleným prvkům, pak snadno měnit jejich vlastnosti. Liniím barvu, šířku, styl čáry a zakončení, polygonům pak pouze barvu. Vytvořené mapy je možné exportovat do souborů, ve formátech png nebo jpeg. Nevýhodou aplikace je možnost načítat data pouze z malých souborů (tvůrce uvádí doporučenou maximální velikost 30 MB), export výsledných map jen v rastrovém formátu, nemožnost vykreslit text a samostatné body.

Maperitive

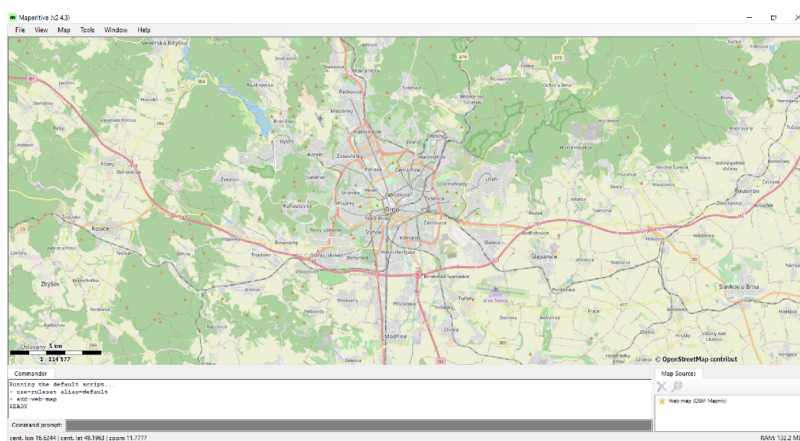
Maperitive [10] (na obrázku 2.4) je nástupce již zaniklé aplikace Kosmos, slouží primárně k vytváření map ve formátech png nebo svg, ale zvládá i automatické vytváření rastrových dlaždic. Lze jej stáhnout zdarma z oficiálních stránek pro operační systémy Windows, Linux i Mac.

Aplikace načítá data pouze ze souborů ve formátu OSM XML. Výhodou aplikace je, že umí stahovat data přímo v aplikaci, přes Overpass API. Pro vykreslení mapy, ze stažených dat, se používá tzv. sada pravidel (ruleset). Ta určuje, jak a která data se vykreslí. Aplikace v základu obsahuje 6 použitelných pravidel:

- Default.mrules – styl OpenStreetMap (starší vzhled)



Obrázek 2.3: Snímek aplikace map-stylizer.



Obrázek 2.4: Snímek aplikace Maperitive s načtenou OpenStreetMap mapu.

- GoogleMaps.mrules – styl Google map (rok 2011)
- Hiking.mrules – styl turistické mapy
- PowerLines.mrules – styl pro zobrazení elektrické rozvodné sítě
- urbanight.mrules – styl pro zobrazení zastavěné oblasti, ve tmavých barvách
- Wireframe.mrules – styl pro zobrazení všech linií šedě a oblastí zeleně

Kromě pravidel, které Maperitive obsahuje v základu, lze vytvářet i vlastní pravidla viz výpis 2.1. Každé pravidlo musí obsahovat povinné bloky a těmi jsou: „features“, „properties“ a „rules“.

V bloku „features“ dochází k přiřazení výběru dat, k proměnným např. všechny linie, které obsahují data s klíčem „waterway“ a hodnotou „river“ se přiřadí pod proměnnou „rivers“.

V dalším bloku „properties“ se definují vlastnosti celé mapy. V uvedeném příkladu je upřesněno, že mapa bude mít bílé pozadí.

Poslední blok „rules“ obsahuje pod bloky značené „target“, v těchto pod blocích se definuje, jaké vizuální vlastnosti budou prvky mít a jaká akce se provede.

Podrobnější vysvětlení, pro tvorbu složitějších pravidel, je k nalezení v dokumentaci⁴, stejně jako několik návodů s různými tipy.

```
features
  lines
    rivers : waterway=river
  areas
    ponds : natural=water AND water=pond

properties
  map-background-color : white

rules
  target : ponds
  define
    fill-color : blue
  draw : fill

  target : rivers
  define
    line-color : lightblue
  draw : line
```

Výpis 2.1: Příklad vlastního pravidla, podle kterého se vykreslí řeky světle modře a rybníky modře.

QGIS

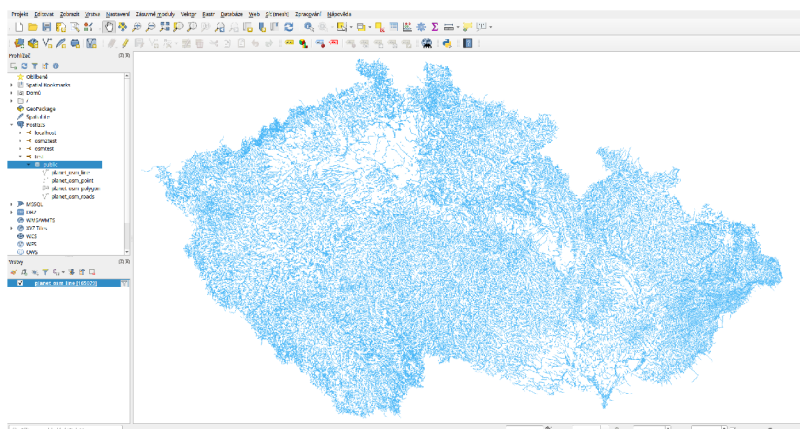
QGIS je komplexní aplikace pro práci s geografickými daty, napsaná v programovacím jazyce C++. Umí načítat data ze souborů ve formátech jako je např. shapefile, csv, OSM XML nebo PBF, ale podporuje také čtení dat z databází PostGIS a MSSQL. Načtená data se dají rozdělovat do vrstev a těm poté nastavovat různé vlastnosti. Styly map lze vytvářet interaktivně, pomocí skriptu v jazyce Python nebo stahovat z internetu a poté aplikovat. Vytvořené mapy umí aplikace exportovat do rastrových formátů bpm, tif, jpg, png, ale i vektorových pdf, svg. [14, 15]

Uživatelské rozhraní (na obrázku 2.5) se skládá z horního panelu, levého panelu, oblasti pro vizualizaci mapy a v případě podrobnějších nastavení se objevuje pravý panel. V horním panelu se nachází různé nástroje pro editaci a nastavení, v levém panelu zdroje dat a vrstvy.

2.4.2 Tvorba vlastních stylů online map

V této sekci je popsán nástroj mapstyle.withgoogle.com, který byl vybrán jako zástupce online vektorových map, u kterých si lze vytvořit vlastní styl, podle kterého se klientovi vykreslují objekty mapy.

⁴<http://maperitive.net/docs>



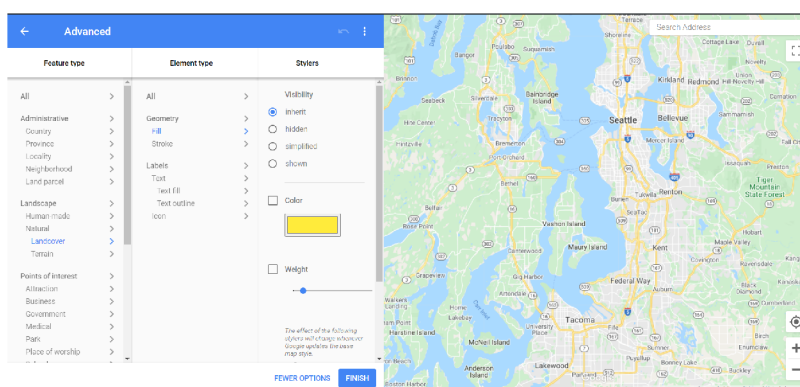
Obrázek 2.5: Snímek obrazovky s uživatelským rozhraním aplikace QGIS, ve kterém jsou zobrazeny vodní toky.

mapstyle.withgoogle.com

Online nástroj⁵ na tvorbu stylů, pro mapy poskytované společností Google. Uživateli je nabídnuto několik předpřipravených stylů mapy, které lze poté upravovat.

Vzhledově jde upravit necelých třicet položek, kde většina z nich sdružuje několik prvků mapy dohromady např. „Landcover“ obsahuje veškerý zemský pokryv a nelze si podrobněji vybrat, který bude vypadat jinak. U všech položek, lze upravovat výplň, štětec, text a ikonu, všechny tyto vlastnosti prvků, pak jde ještě specifikovat viditelností, barvou, důležitostí, saturací a světlostí. Z méně nepochopitelného důvodu může uživatel měnit například výplň státním hranicím, ale jelikož jsou hranice pouze linie, výplň na ně nemá žádný vliv, barvu u nich ovlivní jen barva štětce. Další nevýhodou může být, že změny některých položek se projeví až od určité úrovně přiblížení, protože vektorové nejsou tvořeny do všech úrovní přiblížení, kdyby byly mohlo by to být velice náročné na hardware.

Po dokončení úprav a zmáčknutí tlačítka „Finish“, se objeví vygenerovaný styl ve formátu JSON a odkaz pro vložení do stránky.



Obrázek 2.6: Ukázka webové aplikace mapstyle.withgoogle

⁵<https://mapstyle.withgoogle.com/>

Kapitola 3

Projekt OpenStreetMap

OpenStreetMap [2] je projekt, který vznikl za účelem vytvořit databázi geografických dat z celého světa a tato data pak zdarma poskytovat.

V této kapitole se čtenář seznámí, za jakých okolností projekt vznikl, jaká obsahuje data a v jakých datových formátech je jde stáhnout. Dále je v kapitole popsáno, jak se zapojit do mapování a v závěru jsou pak uvedeny některé alternativy k tomuto projektu.

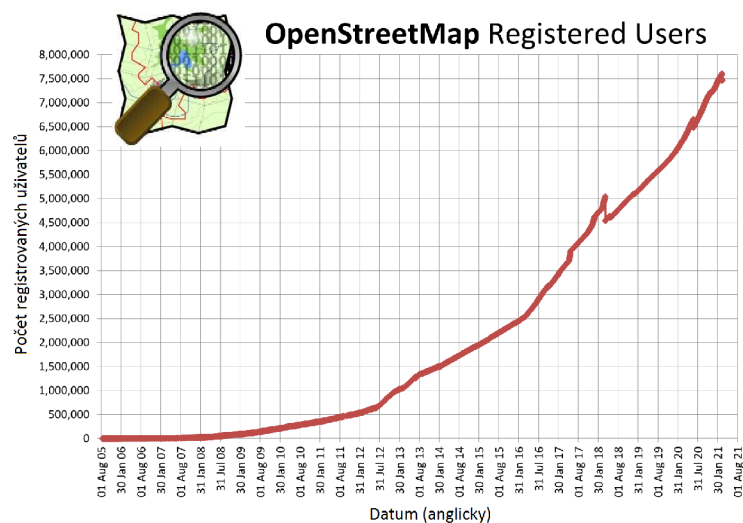
3.1 Vznik projektu

Začátky projektu OpenStreetMap se datují do roku 2004, kdy britský programátor Steve Coast experimentoval s GPS přijímačem. Na začátku používal software, který čerpal data ze zdroje od společnosti Microsoft, ale při své tvorbě nechtěně porušil pravidla licence. Jelikož vědomě licenci nechtěl porušovat a platit obrovskou částku taky ne, pokusil se najít jiný zdroj mapových dat, který by byl volně šiřitelný, ten ale nenašel. Ještě tentýž rok prezentoval svůj vlastní nápad na založení projektu na tvorbu map, do kterého by mohli přispívat daty samotní uživatelé, ostatním účastníkům konference otevřeného softwaru v Londýně. Po své prezentaci se spojil s ostatními účastníky, které projekt zaujal nebo měli podobný nápad, a tak vznikla skupina OpenStreetMap. Začátky byly složité, ale už v roce 2006 vznikla první mapa města Waybridge a o pár měsíců později byla pořádána první společná akce, která měla za úkol zmapovat ostrov Wight. Výsledkem této akce bylo potvrzení, že projekt má své podporovatele a má tedy smysl pokračovat. Vznikla také první barevná mapa takto velké oblasti a nový zvyk pořádat podobné komunitní akce na různých místech planety, které potřebují lepší pokrytí.

Na konci roku 2006 byla oficiálně založena nadace OpenStreetMap Foundation, která tento projekt spravuje. Později, jak se projekt rozrůstal začaly svými daty přispívat soukromé společnosti i města nebo státy [2]. Dne 6. ledna 2013 OpenStreetmaps překonaly hranici 1 milionu registrovaných uživatelů a od té doby čísla stále rostou, uživatelů je v tuto dobu (25. 4. 2021) už skoro o 6,5 milionu více viz graf na obrázku 3.1. A nárůst se netýká pouze uživatelů, databáze obsahuje téměř 7 miliard uzlů [11]¹. Pro další nebo aktuálnější statistiky můžete navštívit tuto stránku².

¹<https://wiki.openstreetmap.org/wiki/Stats>

²https://www.openstreetmap.org/stats/data_stats.html



Obrázek 3.1: Graf vývoje registrovaných uživatelů na stránkách OpenStreetMap od 1. srpna 2005 do 18. března 2021. Převzato z dokumentace [12] a upraveno.

3.2 Data a datové formáty

Tato podsekcce se zabývá OpenStreetMap datovým modelem a datovými formáty OSM XML a PBF, což jsou nejčastěji využívané formáty pro OpenStreetMap data. Soustřeďuje se na jejich popis, výhody, případné nevýhody a kde takovéto soubory stáhnout.

Datový model OpenStreetMap

Datový model OpenStreetMap se skládá ze tří základních prvků: uzlů, cest a relací. Tyto prvky pak doplňuje značka.

Uzel [11]³ (anglicky node) reprezentuje jedinečný bod na mapě (viditelný i neviditelný) a tvoří tak pomyslnou základní stavební jednotku všech ostatních prvků mapy. Každý uzel je označen číselnými údaji, které reprezentují zeměpisnou šířku (latitude) a zeměpisnou délku (longitude). V případech, kdy se nachází jeden uzel přímo nad druhým, je uzel doplněn ještě o specifikaci úrovně. U každého uzlu je tak zcela jasné, kde se nachází.

Cesta [11]⁴ (anglicky way) se skládá ze seřazeného seznamu 2 až 2000 uzlů a obsahuje většinou 1 a více značek. Cesty na mapě reprezentují linie tedy čáry. Cesty se dělí do tří skupin.

- Otevřená cesta – linie, která má počáteční a koncový bod jiný.
- Uzavřená cesta – linie, která začíná a končí ve stejném bodě např. kruhový objezd.
- Oblast – linie, která začíná a končí ve stejném bodě, ale na rozdíl od uzavřené cesty má výplň. Oblast je např. park nebo les.

³<https://wiki.openstreetmap.org/wiki/Node>

⁴<https://wiki.openstreetmap.org/wiki/Way>

Relace [11]⁵ (anglicky relation) je skupina, která se skládá z uzlů, cest nebo jiných relací (takováto relace je poté označována jako super relace). Člen relace je tzv. účastník (member). Relace se dělí do mnoha skupin pro podrobnější přehled, viz [11]⁶

Značka [11]⁷ (anglicky tag) se skládá z klíče a hodnoty. Klíč určuje téma, kategorii nebo typ prvku a hodnota je pak specifická vlastnost. Používá se u uzlů, cest i relací.

OSM XML

Tento datový formát je založen na jazyku XML a soubory v něm se značí příponou `.osm`. Data v souboru začínají XML prologem, poté následuje kořenový element `<osm>` s atributy specifikujícími generátor a licenci. V elementu `<osm>` se nachází dětský element `<bounds>` s atributy označující hranice oblasti, v tomto elementu se pak dále zanořují elementy `<node>`, `<way>` a `<relation>`, u každého z těchto tří elementů jsou uvedeny tyto atributy:

- identifikační číslo prvku (id)
- uživatel (user), který prvek vytvořil nebo naposledy upravil
- uživatelské identifikační číslo (uid)
- časová značka vytvoření nebo poslední změny (timestamp)
- verze (version)
- sada změn (changeset), značící skupinu změn provedených uživatelem
- viditelnost (visible), která upřesňuje, jestli objekt stále existuje a má se tedy zobrazit na mapě

Ručně vybranou specifickou oblast skládající se maximálně z pěti tisíc uzlů, je možné přímo exportovat ze stránky s mapou⁸ OpenStreetMap v záložce „Export“, pokud je tento limit překročen, tak se oblast nestáhne a uživatel je upozorněn. Větší oblasti se dají stahovat na stejném místě pomocí Overpass API. Celou mapu v tomto formátu si lze stáhnout z oficiální stránky⁹. Další specifické oblasti nebo různé datové výřezy pak nabízejí ke stažení různé další stránky např. tato stránka¹⁰ nabízí výřezy z více než 200 velkých měst.

Element `<node>`

Element `<node>` (ukázka na výpisu 3.1) v tomto datovém formátu reprezentuje prvek uzel. Tento element může být rodičem elementům typu `<tag>`. Kromě atributů již zmíněných v této sekci, ale navíc obsahuje atributy „lat“ a „lon“, které udávají hodnotou zeměpisnou šířku a zeměpisnou výšku.

```
<node id="1689128648" visible="true" version="4" changeset="99689712"
timestamp="2021-02-21T17:00:29Z" user="osmbit" uid="11456907"
lat="49.2277295" lon="16.5958584">
```

⁵<https://wiki.openstreetmap.org/wiki/Relation>

⁶https://wiki.openstreetmap.org/wiki/Types_of_relation

⁷<https://wiki.openstreetmap.org/wiki/Tags>

⁸<https://www.openstreetmap.org/>

⁹<https://planet.openstreetmap.org/>

¹⁰<https://download.bbbike.org/osm/bbbike/>

```
<tag k="amenity" v="bench"/>
<tag k="backrest" v="yes"/>
<tag k="check_date" v="2021-02-21"/>
</node>
```

Výpis 3.1: Příklad uzlu v datovém formátu OSM XML.

Element <way>

Rodičovský element <way> (ukázka na výpisu 3.2) má dětské elementy <nd>, které představují uzly. Atributem „ref“ odkazují na jejich identifikační čísla.

```
<way id="30359936" visible="true" version="4" changeset="99689728"
timestamp="2021-02-21T17:00:45Z" user="osmbit" uid="11456907">
  <nd ref="334877211"/>
  <nd ref="334877214"/>
  <tag k="highway" v="path"/>
  <tag k="surface" v="sand"/>
</way>
```

Výpis 3.2: Příklad cesty v datovém formátu OSM XML.

Element <relation>

Element <relation> (ukázka na výpisu 3.3) je vždy rodičem alespoň jednoho elementu <tag> a <member>, jehož atribut „type“ označuje typ prvku (uzel, cesta, relace) a „ref“ identifikační číslo, poslední atribut „role“ určuje roli objektu v relaci, tato hodnota může být i prázdná [11]¹¹.

```
<relation id="10189024" visible="true" version="2" changeset="76061883"
timestamp="2019-10-22T17:25:36Z" user="vop" uid="18683">
  <member type="way" ref="737655139" role="outline"/>
  <member type="way" ref="44684869" role="part"/>
  <member type="way" ref="737655138" role="part"/>
  <tag k="type" v="building"/>
</relation>
```

Výpis 3.3: Příklad relace v datovém formátu OSM XML.

Element <tag>

Element <tag> má povinné atributy klíč (key) a hodnota (value). Klíče i hodnoty mohou nabývat libovolných hodnot, existují ovšem konvence¹², kterými by se mapující uživatelé měli řídit. V jednom elementu typu uzel, cesta nebo relace nemůže být více značek se stejným klíčem.

U ukázky uzlu 3.1 první značka upřesňuje, že uzel reprezentuje lavičku. Druhá, že jde o lavičku s opěradlem a třetí zaznamenává datum poslední kontroly uzlu.

¹¹<https://wiki.openstreetmap.org/wiki/Relation>

¹²https://wiki.openstreetmap.org/wiki/Editing_Standards_and_Conventions

Ve druhé ukázce 3.2 označuje první značka, že se jedná o cestu pro chodce a druhá upřesňuje, že cesta je písčítá.

A v poslední ukázce 3.3 značka určuje, že uvedená relace reprezentuje budovu.

PBF

PBF [11]¹³ je binární datový formát, jedná se o alternativu k OSM XML. Jeho výhodou je několika násobně menší velikost souboru a rychlost zápisu i čtení než komprimovaný OSM XML formát. Další výhodou je náhodný přístup k jednotlivým blokům dat na nejnižší úrovni. To umožňuje lehce separovat malé části z většího souboru např. nástrojem Osmium. Soubory v tomto formátu jsou značeny koncovkou `.osm.pbf`.

Data z celé planety je možné stáhnout ze stránek OpenStreetMap¹⁴ nebo Geofabrik¹⁵, kde se dají stáhnout i menší oblasti podle kontinentů, států, provincií nebo měst.

3.3 Jak přispívat

OpenStreetMap projekt je založen na získávání dat od dobrovolníku, proto by každý, kdo využívá tyto mapy nebo data, měl přispět.

Pro přispívání do OpenStreetMap dat si stačí vytvořit na oficiálních stránkách účet. Po potvrzení emailové adresy stránka uživatele v krátkém přehledu seznámí s tím, co do mapy patří a co ne, se základními pojmy a pravidly. Poté už lze začít tzv. mapovat, na hlavní stránce s mapou si vybrat oblast, na které chce uživatel něco upravit a kliknout na tlačítko „Upravit“. Upravovat jde v prohlížeči online nástrojem iD¹⁶ nebo počítačovými aplikacemi, jako je například Merkaartor¹⁷ nebo JOSM¹⁸. Zmíněné počítačové aplikace jsou ale doporučeny spíše až pro zkušenější uživatele. V případě volby editoru iD nabízí OpenStreetMap i velmi praktický interaktivní návod, který uživatele seznámí se všemi základy, tzn. jak vkládat do mapy body, cesty, oblasti nebo domy a jak k nim přidávat různé popisky. Pokud si uživatel není jistý, jestli vkládá prvky do mapy správně, může požádat o kontrolu, tu pak provede některý ze zkušených uživatelů. K už existujícímu prvku, u kterého uživatel neví, jak ho správně upravit, může vložit komentář s popisem problému nebo změny, kterou je třeba provést. Případnou změnu poté opět provede někdo zkušenější. Po uložení změn se během několika hodin úpravy promítnou do oficiální mapy. Kromě přispívání mapováním, jde přispět i jinak, jak lze zjistit v dokumentaci¹⁹.

3.4 Alternativy

Jednou z alternativ mohou být data společnosti Esri v systému ArcGIS. Tam jsou kromě placených služeb dostupná i geografická data zdarma.

ArcGIS Hub²⁰ je platforma pro sdílení geografických dat uživateli, organizacemi nebo společnostmi z celého světa. Data mohou být čistě geografická nebo obsahovat informace

¹³https://wiki.openstreetmap.org/wiki/PBF_Format

¹⁴<https://planet.openstreetmap.org>

¹⁵<https://download.geofabrik.de>

¹⁶<https://github.com/openstreetmap/iD>

¹⁷<http://merkaartor.be>

¹⁸<https://josm.openstreetmap.de>

¹⁹https://wiki.openstreetmap.org/wiki/How_to_contribute

²⁰<https://hub.arcgis.com>

z různých průzkumů vztahujících se ke specifickým územím. Je možné je stahovat ve formátech kml, csv nebo shapefile. Výhodou této služby je rozmanitost dat, nevýhodou může být neucelenost.

ArcČR@500²¹ je databáze geografických dat České republiky. Obsahuje základní geografické prvky, jako jsou např. lesy, vodní plochy nebo silniční síť.

Další možnosti jsou data poskytovaná přímo společnostmi nebo státy, například v Česku jsou zdarma dostupná katastrální data. OpenStreetMap nabízí na svých stránkách seznam²² potenciálních zdrojů, který obsahuje odkazy nebo popisy, jak data získat.

²¹<https://www.arcdata.cz/produkty/geograficka-data/arccr-500>

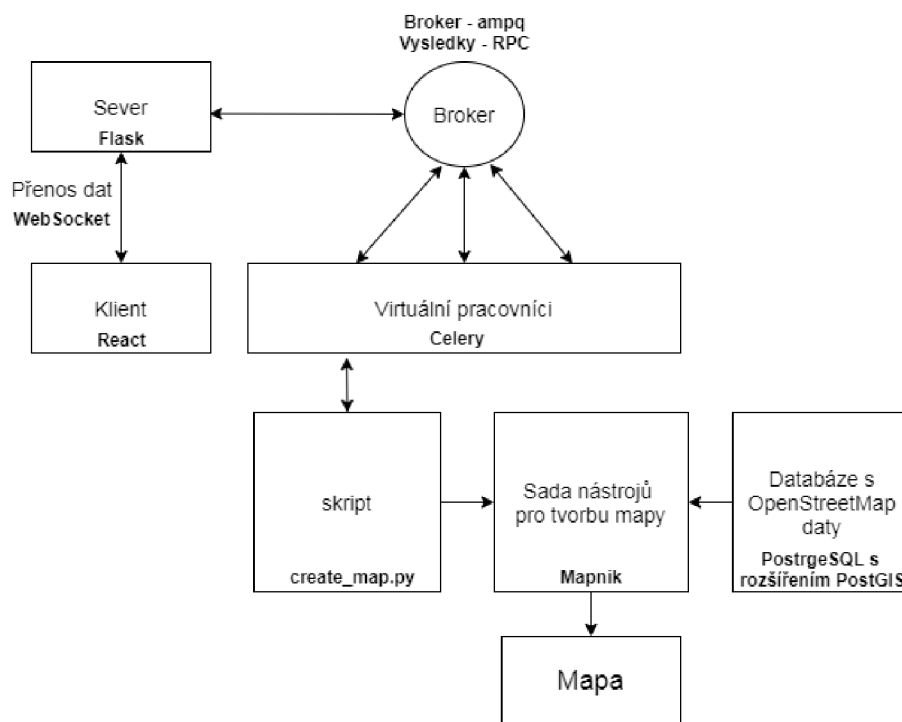
²²https://wiki.openstreetmap.org/wiki/Potential_Datasources

Kapitola 4

Využití technologie

V této kapitole jsou popsány všechny důležité technologie, které byly použity k vytvoření uživatelského rozhraní, komunikace mezi klientem a serverem, webového serveru a procesu tvorby mapy. Kromě popisu technologie je zmiňován i důvod jejího použití.

Celá webová aplikace vznikla propojením všech technologií, uvedených v této kapitole. Toto propojení je znázorněno ve schématu na obrázku 4.1.



Obrázek 4.1: Schéma propojených technologií, které vytváří mapy.

4.1 React

React¹ je moderní knihovna s otevřeným zdrojovým kódem, pro tvorbu uživatelského rozhraní, v programovacím jazyce JavaScript vytvořená společností Facebook. Nejčastěji se

¹<https://reactjs.org>

využívá pro tvorbu jednostránkových webových aplikací a mobilních aplikací. Umožňuje rozdělit kód uživatelského rozhraní do nezávislých částí tzv. komponent, které se dají kdykoliv znovu použít.

Tuto knihovnu jsem si vybral, protože už podle návrhu byla webová aplikace konstruována jako téměř jednostránková (další případné stránky tvoří pouze text) a dost dynamická. Na hlavní stránce vyvinuté aplikace musí být propracovaný systém výběru prvků mapy, který dokáže rychle a spolehlivě reagovat na uživatelské akce. Rovněž bylo využito rozdělení do komponent, protože se některé části často opakují nebo dynamicky objevují a mizí.

4.2 Flask

Flask² je jednoduchý aplikační rámec napsaný v programovacím jazyce Python. Tento aplikační rámec byl vytvořen tak, aby poskytoval jen základní služby, ale byl snadno rozšiřitelný přidávanými balíčky. Tyto balíčky pak tvůrci poskytují přesně to, co potřebuje. [8]

Flask jsem si zvolil, jelikož umožňuje rychlou komunikaci s klienty, bez nutnosti aktualizace stránek. A taky právě kvůli jeho jednoduchosti, protože bylo navrženo, že server bude potřebovat jen funkce pro komunikaci s klientem a asynchronním strojem, který bude vytvářet mapy. Kdyby bylo potřeba, dalších nástrojů nebo funkcí, balíčky lze server jednoduše rozšířit.

4.3 WebSocket

WebSocket je komunikační protokol, který zajišťuje plně duplexní oboustranné spojení mezi klientem a serverem. Má také nižší odezvu, jelikož je spojení stále navázáno, klient i server mohou posílat zprávy, kdykoliv je potřeba a nemusí navazovat stále nová spojení. Pro navázání spojení klient pošle požadavek přes HTTP protokol, že by chtěl používat komunikaci skrze protokol WebSocket, pokud je ze strany serveru také podporován, tak server začne tímto protokolem komunikovat na stejném portu jako bylo HTTP spojení. [17] WebSocket používá například Github nebo Google ve svých Google dokumentech.

Tuto technologii jsem si vybral, protože poskytuje rychlou a bezpečnou komunikaci bez nutnosti celou stránku obnovovat a navíc umožňuje serveru kontaktovat klienta bez nutnosti akce z klientovy strany. K realizaci tohoto druhu komunikace byla na straně klienta zvolena knihovna socket.io-client³ a na straně serveru Flask-SocketIO⁴.

4.4 Celery

Celery⁵ je software s otevřeným zdrojovým kódem pro asynchronní zpracování úkolů, napsaný v programovacím jazyce Python, pro který je také primárně určen, ale je možné jej použít v upravených formách např. v Ruby nebo GO. U webových aplikací se často využívá, aby složité a dlouhotrvající úkoly nezatěžovaly běh webového serveru, z tohoto důvodu jsem ho využil i já. Jeho úkolem je v mnou implementované aplikaci tvorba map podle param-

²<https://flask.palletsprojects.com/en/2.0.x>

³<https://www.npmjs.com/package/socket.io-client>

⁴<https://github.com/miguelgrinberg/Flask-SocketIO>

⁵<https://docs.celeryproject.org/en/stable/index.html>

trů uživatelů. V části implementace (soubor cel.py), jsem převzal kód, který je uvedený na stránce⁶.

Ke komunikaci mezi webovým serverem a Celery slouží tzv. „broker“, ten přiděluje práci virtuálním pracovníkům neboli procesům, kteří je plní. V aplikaci tuto roli zastává základní „broker“ RabbitMQ⁷, kvůli jeho stabilitě a jednoduchosti.

4.5 PostgreSQL

PostgreSQL⁸ (nazýván také jako Postgres) je objektově-relační databázový systém, který je vyvíjen jako otevřený software. Podporuje různé datové typy např. základní (integer, řetězec, boolean), strukturované (datum, pole, rozsah), geometrické (polygon, line, bod). A je rozšiřitelný o celou řadu rozšíření⁹.

PostGIS¹⁰ je jedno z rozšíření do Postgres. Toto rozšíření přidává podporu geografických objektů a umožňuje s nimi provádět celou řadu funkcí jako třeba průnik, sjednocení nebo rozdíl.

Tuto databázi s rozšířením jsem si vybral primárně kvůli, sadě Mapnik. Pro tu se tato kombinace databázového systému Postgres a rozšíření PostGIS využívá často, hlavně kvůli již zmíněné podpoře geografických objektů.

Podpůrné nástroje

Osm2pgsql¹¹ je nástroj, který se používá primárně pro vykládání OpenStreetMap dat do PostgreSQL s rozšířením PostGIS databáze a jejich aktualizaci. Pro spuštění nástroje je nutné mít stažená data a styl podle, kterého se data importují do tabulek. Při importu dochází k vytváření geometrických útvarů jako jsou body, linie, polygony a multipolygony, každý útvar je před vložením zkontrolován, jestli je validní. Podporované formáty dat pro import jsou OSM XML, PBF nebo O5M. Při vkládání se používá soubor s koncovkou .style, který určuje, do které tabulky se která data vloží a která se zahodí. Pokud tabulka neexistuje tak ji vytvoří. Při vytváření aplikace jsem použil a upravil styl z online repozitáře¹²

Jelikož může být problém získat pouze data oblasti, která je potřeba, může se využít nástroj Osmium¹³. Je to nástroj pro práci s OpenStreetMap daty ve formátu PBF, tato data dokáže např. porovnávat, převádět, spojovat nebo vykrajovat menší oblasti. Lze tedy stáhnout větší kus mapy a vzít si, jen kolik je potřeba. Tento nástroj byl použit v raném vývoji pro zmenšování oblastí.

4.6 Mapnik

Mapnik je sada nástrojů, primárně určených pro vytváření map a mapových dlaždic. Je napsaný v C++, ale lze jej skriptovat pomocí vazebných jazyků Node.js, Python, Ruby a Java. Umí používat data z PostGIS databáze, shapefile souboru, starší verze uměly přes

⁶<https://flask.palletsprojects.com/en/1.1.x/patterns/celery>

⁷<https://www.rabbitmq.com>

⁸<https://www.postgresql.org/docs/13/index.html>

⁹<https://www.postgresql.org/download/products/6-postgresql-extensions>

¹⁰<https://postgis.net/docs/manual-3.0>

¹¹<https://osm2pgsql.org/doc/manual.html>

¹²<https://github.com/openstreetmap/osm2pgsql/blob/master/default.style>

¹³<https://osmcode.org/osmium-tool/manual.html>

rozšíření i OSM XML, csv a několika dalších. Výsledné mapy nebo dlaždice mohou být ve formátech png, jpeg, svg nebo pdf. [11]¹⁴ Velkou nevýhodou tohoto nástroje je absence aktuální dokumentace a relevantních informací. Jediný oficiální zdroj informací, který je alespoň trochu aktuální je github wiki¹⁵. Projekty, které Mapnik využívají k vytváření dlaždic jsou např. mapy.cz nebo OpenStreetMap.

Vykreslování map často probíhá podle stylů napsaných v Mapnik XML. Příklady různých uživatelů, si lze prohlédnout na této stránce¹⁶. Styl tvoří kořenový element `<Map>`, atributy tohoto elementu specifikují pozadí mapy a projekci. Nejdůležitější částmi předpisu je styl (v Mapnik XML značen elementem `<Style>`), který určuje, jak se co vykreslí a vrstva (značena elementem `<Layer>`), která určuje co se vykreslí.

```
<Style name="highway_style">
  <Rule>
    <Filter>[highway]="motorway"</Filter>
    <LineSymbolizer stroke="red" stroke-width="2"/>
  </Rule>
  <Rule>
    <Filter>[highway]="primary"</Filter>
    <LineSymbolizer stroke="green" stroke-width="1"/>
  </Rule>
</Style>
```

Výpis 4.1: Příklad stylu v předpisu Mapnik XML.

Styl (příklad stylu ve výpisu 4.1) může obsahovat jedno a více pravidel. V pravidlu, pokud chceme něco vytvořit, by měl být symbolizér, jehož atributy udávají parametry vykreslených prvků. Symbolizér se volí podle toho co se bude vykreslovat:

- LineSymbolizer – vykresluje linie
- PolygonSymbolizer – vykresluje polygony
- PointSymbolizer – vykresluje body
- TextSymbolizer – vykresluje text
- BuildingSymbolizer – vykresluje polygony ve 3D vzhladu, určen na vizualizaci budov
- ShieldSymbolizer – kombinace obrázku a textu
- LinePatternSymbolizer – vykresluje linie se vzorem daným obrázkem
- PolygonPatternSymbolizer – vyplňuje polygon vzorem daným obrázkem
- MarkersSymbolizer – alternativa k PointSymbolizer, poskytuje podrobnější možnosti

Pravidlo může obsahovat i filtr, který upřesňuje, pro které data pravidlo platí (pro případy, kdy dotaz na databázi je pro více prvků, které se vykreslují jinak nebo je potřeba vykreslit prvky pro více úrovní přiblížení). V příkladu je uvedeny dva pravidla se symbolizéry linií a filtry. Podle prvního pravidla se všechny dálnice vykreslí červeně s šířkou 2 pixel a podle druhého se všechny silnice první třídy vykreslí zeleně s šířkou 1 pixel.

¹⁴<https://wiki.openstreetmap.org/wiki/Mapnik>

¹⁵<https://github.com/mapnik/mapnik/wiki>

¹⁶<https://github.com/mapnik/mapnik/wiki/StyleShare>


```
<Layer name="highway_layer" status="on" srs="PROJEKCE">
  <StyleName>highway_style</StyleName>
  <Datasource>
    <Parameter name="table">
      (select way, highway
      from planet_osm_line
      where (highway = "motorway" or highway="primary"))
      as highways
    </Parameter>
    ZDROJ DAT
  </Datasource>
</Layer>
```

Výpis 4.2: Příklad vrstvy v předpisu Mapnik XML

Ve vrstvě (příklad vrstvy ve výpisu 4.2) se definuje dotaz na databázi, ve kterém je upřesněno, která data a odkud se vyberou. Zdroj dat obsahuje údaje k databázi, ve kterou se budou provádět dotazy. Uvedené údaje jsou: typ zdroje dat, heslo, host, port, uživatel, název databáze, specifikace oblasti.

Mapnik jsem si k tvorbě map vybral, kvůli možnosti vytvářet mapy různých rozměrů v rastrových i vektorových formátech a zaměření na vytváření map z OpenStreetMap dat.

Kapitola 5

Návrh

V této kapitole je nejdříve popsána analýza vlastností a funkcí, které by výsledná aplikace měla mít. Poté jsou popisovány jednotlivé části navrženého uživatelského rozhraní a v poslední sekci je navrženo fungování webového serveru. Všechny návrhy uživatelského rozhraní a diagramy, v této kapitole i v kapitole testování 7.1, byly vytvořeny v aplikaci draw.io¹, pokud není uvedeno jinak.

5.1 Analýza

Tato podkapitola se zabývá popisem cílové skupiny uživatelů, která by mohla aplikaci používat, sběrem a analýzou požadavků potenciálních uživatelů na aplikaci, akcemi, které by uživatel mohl v této aplikaci provádět a na závěr popisem ideálních uživatelů.

5.1.1 Cílová skupina a použití

Cílovou skupinu tvoří dva druhy uživatelů. První druh tvoří skupiny, jako jsou např. učitelé, táboroví vedoucí, skauti. Tito uživatelé vytváří více-méně slepé mapy nejčastěji pro vzdělávací nebo volnočasové účely. Druhým druhem jsou cestovatelé, dobrodruzi, sportovci nebo turisté. Tito uživatelé mapu vytvářejí, aby si zobrazili své koníčky nebo cíle. Nemí vyloučeno, že uživatel může být obojího druhu.

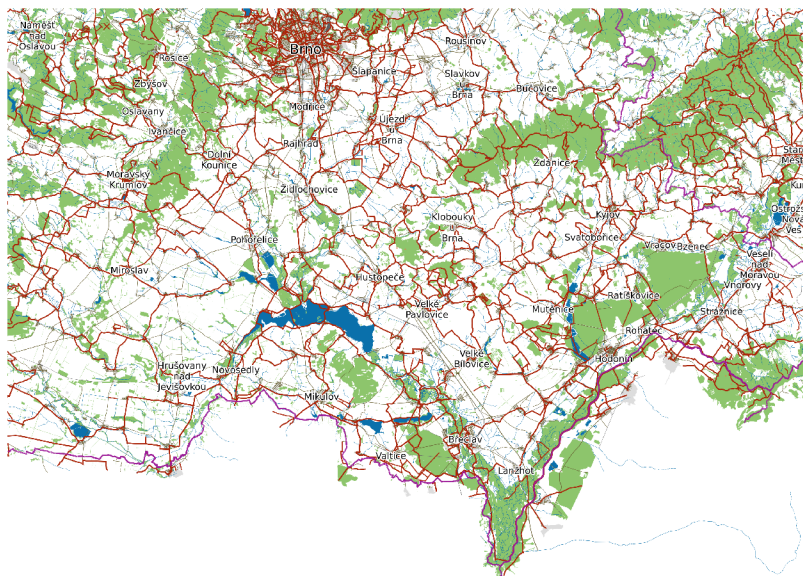
5.1.2 Sběr a analýza požadavků

Sběr informací probíhal formou dialogů a krátkých zpráv s několika potenciálními uživateli, obou druhů z cílové skupiny. Během těchto dialogů a výměn zpráv, vznikl seznam vlastností, které by výsledná webová aplikace měla určitě mít ze stránky vzhledové i funkční. Společně s tímto seznamem vznikly tři návrhy map, které by měly být určité vytvořit.

První mapa je „mapa cyklisty“ viz obrázek 5.1, tato mapa zobrazuje území Jihomoravského kraje a jsou na ní cyklostezky, silnice, řeky, vodní plochy, lesy, zastavěná území a názvy měst. Tato mapa může sloužit jako plánec, který má cyklista u sebe, a do kterého si zakresluje, které trasy už projel nebo plánuje projet.

Druhá mapa má název „mapa pivaře“ viz obrázek 5.2. Mapa zobrazuje základní prvky ze středu města Brna, jako je řeka, budovy, cesty, železnice a zelené plochy. K těmto základním prvkům jsou na mapě umístěny ikony zobrazující hospody a pod nimi jejich názvy. Mapa je

¹<https://app.diagrams.net>



Obrázek 5.1: Navržená mapa – „mapa cyklisty“, vytvořeno sadou nástrojů Mapnik z OpenStreetMap dat.

vhodná pro vytištění ve velkém formátu, aby šel text přečíst a ikony pořádně vidět. „Pivař“ si tak může tuto mapu doma vyvěsit a zakreslovat, kde už všude byl.

Třetí a poslední mapou je téměř klasická slepá mapa zobrazující Českou republiku, rozdělenou do regionů s vyznačenými vrcholky hor nad 1200 m n. m. viz obrázek 5.3. Mapa by tak mohla být vhodná do škol nebo pro cestovatelé, kteří tyto vrcholky rádi zdolávají.

5.1.3 Případy užití

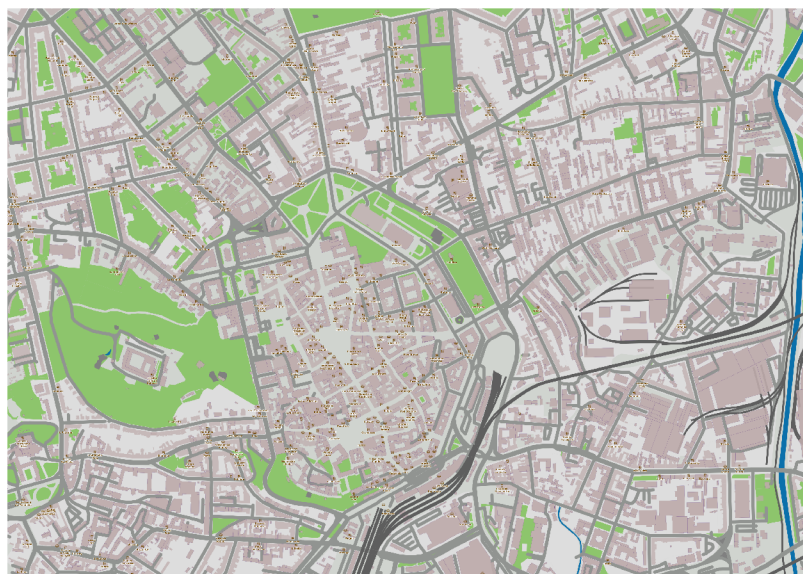
Na základě vytvořeného seznamu a rozhovorů s uživateli vznikl diagram případů užití 5.4, který by měl reprezentovat všechny akce, které může uživatel ve webové aplikaci provádět. Diagram byl v průběhu vývoje a testování upravován a toto je poslední verze.

Uživatel při návštěvě webové aplikace může provádět celou řadu akcí, v jakém pořadí chce. Může vybrat velikost výsledné mapy, datový formát mapy, celé skupiny prvků nebo přímo jednotlivé prvky mapy, které na své mapě chce. Po výběru může zobrazit jejich nastavení a poté je upravovat, tím je myšleno, měnit jejich barvu nebo velikost, záleží podle druhu prvku. Pokud si to rozmyslí, může výběr skupiny nebo jednotlivého prvku zrušit, tím ale přijde o možnost jejich úpravy. Pokud si uživatel chce kousek mapy nechat zobrazit jako náhled, musí pro něj vybrat oblast. Pro vytvoření mapy a následného stažení je nutné mít pro tuto mapu vybranou oblast.

5.1.4 Persona

Z popisu cílové skupiny byly vypracovány dvě osoby, kdy každá persona odpovídá jednomu druhu uživatele.

První personou je Karel. Karel je svobodný, je mu 28 let a bydlí ve středně velké městě, kde pracuje jako účetní. Karel pravidelně o letních prázdninách dělá vedoucího na letním táboře. Každý rok pořádají výlet do jednoho ze zajímavých měst v okolí a všichni vedoucí musí vymyslet činnost, která bude součástí programu. Karel vytváří více-méně slepé mapy,



Obrázek 5.2: Navržená mapa – „mapa pivaře“, vytvořeno sadou nástrojů Mapnik z OpenStreetMap dat.

každý rok trochu jinak a k nim vymýšlí různé úkoly, jako např. zakreslete do mapy, ve které jsou vyobrazeny domy, bez jakéhokoliv popisu, společně s několika dalšími prvky, všechny památky, kolem kterých projdeme. Další mapu, kterou má předpřipravenou pro každý rok, je mapa tábora, do které ručně zakresluje stanoviště, při různých kolektivních hrách. Karel pro tvorbu těchto map potřebuje nástroj, ve kterém mapu snadno a rychle vytvoří, bude si moci vybrat, co právě potřebuje a výsledek bude v dobré kvalitě, ve velké i menší velikosti.

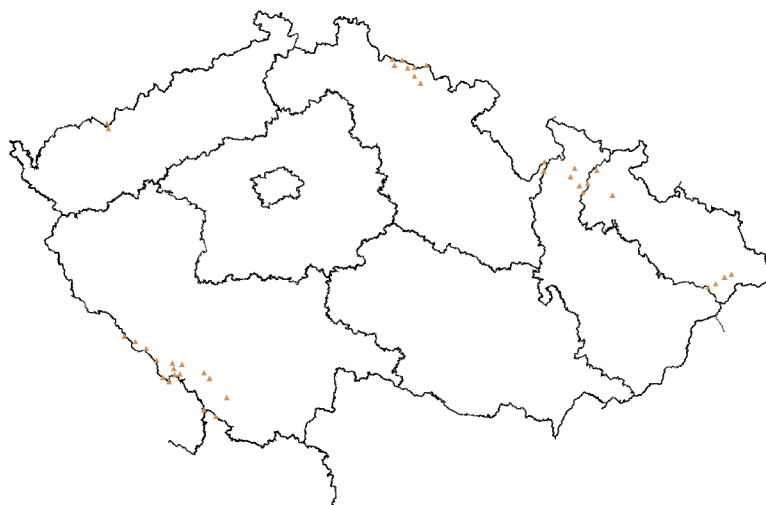
Druhou personou je Václav, ten bydlí s rodinou na předměstí. Václavovi je 50, má vlastní firmu a rád jezdívá společně s rodinou na dovolenou nebo o víkendech do přírody, třeba na hory. Václav si kromě turistické mapy, kterou si koupil pro případ, že by se ztratil, sebou vozí také více-méně slepou mapu, ve které má stezky, památníky, památky, vrcholy hor a restaurace. Do této mapy si zaznamenává, které stezky už prošel, které by chtěl projít znovu, ve které restauraci mu chutnalo, a ve které ne. O tyto poznatky a zážitky se dělí s kamarády z turistického klubu, kteří taky mají své více-méně slepé mapy a z informací, které si poznamenali, si vzájemně doporučují, kam se vydat příště. Václav potřebuje, aby vytvořená mapa byla celkem přesná, ve výborné kvalitě, spíše menší velikosti, aby ji měl vždy po ruce a obsahovala jen to, co doopravdy potřebuje.

5.2 Návrh uživatelského rozhraní

V tomto návrhu uživatelského rozhraní jsem vycházel z názorů uživatelů k existujícím aplikacím, vzniklého seznamu vlastností a diagramu případu užití. Návrh byl během testování upravován. Výsledky některých úprav jsou popsány a zobrazeny v kapitole 7.1.

Horní panel

Horní panel (v obrázku 5.5 označeno číslem 1) se nachází v horní části aplikace a slouží k přechodu na případné další stránky s popisky, obsahuje prostor pro logo a tlačítko pro



Obrázek 5.3: Navržená mapa České republiky s vrcholky hor, vytvořeno sadou nástrojů Mapnik z OpenStreetMap dat

změnu jazyka. Do návrhu byly zapracovány dvě karty, které obsahují popis projektu a kontakt. Texty by měla aplikace být schopná zobrazit v češtině a po stisku tlačítka v angličtině.

Levý panel

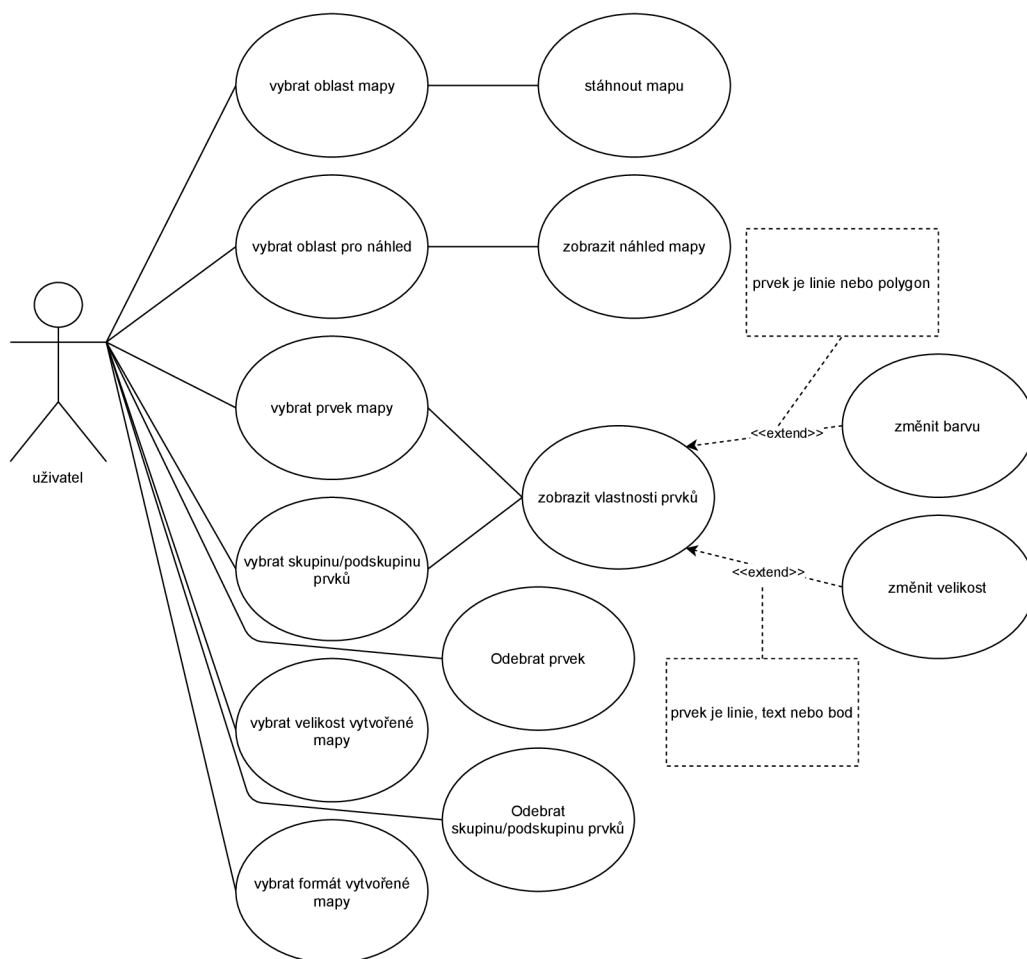
Panel sloužící k výběru prvků nebo skupin prvků mapy (v obrázku 5.5 označeno číslem 2 a detail na obrázku 5.7). Tato část aplikace je pro uživatele asi nejdůležitější a tráví v ní nejvíce času. Nachází se na levé straně aplikace pod horním panelem a je stejně vysoký jako náhled.

Z názorů potencionálních uživatelů vyplynulo, že nejvíce přívětivý způsob výběru prvků měla aplikace map-stylizer viz podsekcce 2.4.1. Vyhovovalo jim zobrazování pomocí zaškrťovacího pole a úpravy parametrů zobrazených prvků na stejném místě. Naopak jim nevyhovovaly některé názvy a použité rozdělení do kategorií. Těmto zjištěním byl návrh uzpůsoben a v pozdější fázi bude předmětem testování. Návrh tohoto panelu lze rozdělit do 3 částí.

První část (na obrázku 5.7 značeno 2.1) tvoří tři tlačítka, která reprezentují různé režimy zobrazení polygonů (lesů, vodních ploch, atd.). Tyto režimy by měly uživateli usnadnit a zrychlit tvorbu.

- „Základní“ – představuje základní režim, na mapě nejsou žádné polygony.
- „Pokročilý“ – režim, při kterém se zobrazí všechny polygony.
- „Vlastní“ – při zaškrtnutí pole se zobrazí samostatné možnosti, jak to je zobrazeno na obrázku 5.7 v oddílu 2.2.

Druhá část (na obrázku 5.7 značeno 2.2) obsahuje skupiny prvků, seskupené podle relevance. Kliknutím na skupinu se zobrazí její podskupiny nebo členové. Členové by měli být zanořeni maximálně dvakrát, aby uživatel neztrácel čas případným hledáním. Pokud si uživatel vybere prvek, který by na mapě chtěl, tak klikne na zaškrťovací pole. Po kliknutí



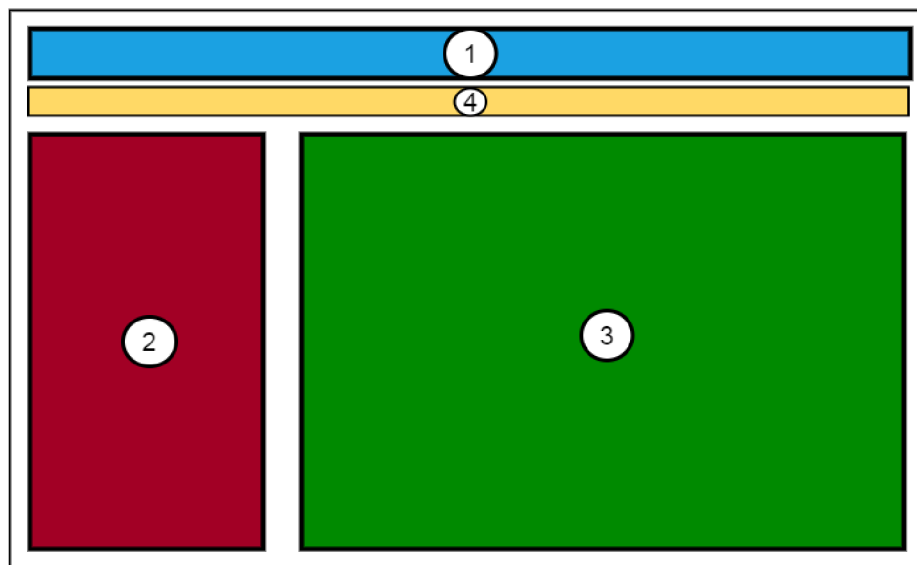
Obrázek 5.4: Diagram užití navrhované webové aplikace.

se u tohoto prvku objeví možnosti úpravy jeho zobrazení, pokud si to uživatel rozmyslí a odklikne zaškrťovací pole, možnosti opět zmizí. Tento výběr se zobrazí pouze u koncových členů skupin.

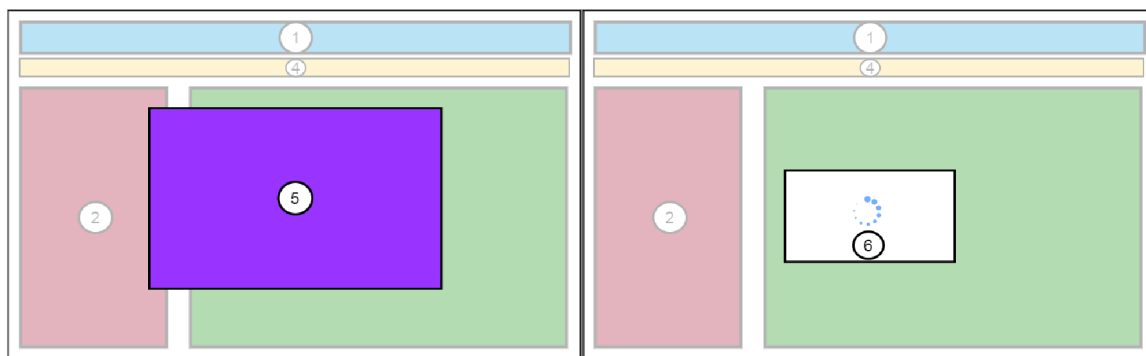
V poslední části (na obrázku 5.7 značeno 2.3) jsou tři tlačítka a dvě rozbalovací nabídky. Tlačítka „Náhled“ a „Stáhnout“ po stisknutí odešlou seznam vybraných prvků, jejich parametry a velikost mapy na server s požadavkem označujícím, že jde o návrh nebo stahování, pokud jde o stahování, odesílá se i vybraný formát mapy. Tlačítko „Výběr oblasti“ zobrazí schovanou část pro výběr oblasti. Rozbalovací nabídky „Formát“ a „Velikost“ mají předdefinované možnosti, které specifikují datový formát a velikost.

Náhled

Náhledová část (v obrázku 5.5 označeno číslem 3) je tvořena rámečkem, který reprezentuje oblast, ve které se zobrazí vytvořená mapa s prvky, po stisku tlačítka „Náhled“. Účelem této části je umožnit uživateli podívat se na mapu předtím, než si jí stáhne. Obrázek, který se zobrazí, bude vždy ve formátu png, převážně kvůli nižší velikosti. Rámeček by měl mít možnost přiblížení a oddalování, aby měl uživatel pohled, který potřebuje.



Obrázek 5.5: Návrh uživatelského rozhraní



Obrázek 5.6: Návrhy uživatelského rozhraní rozšířené o části 5 a 6

Oznámení

Malý panel (v obrázku 5.5 označeno číslem 4) pod horním panelem, ve kterém se zobrazí oznámení, pokud nastane nějaká chyba (např. pokud chce uživatel stáhnout mapu, ale nevybral žádnou oblast) nebo naopak proběhla úspěšná akce.

Výběr oblasti

Tato část (v obrázku 5.6 označeno číslem 5) se objeví před ostatními částmi uprostřed obrazovky, po stisku tlačítka „Vybrat oblast“. Bude obsahovat mapu světa, na které se uživatel může pohybovat a najít si oblast, kterou chce. Poté co ji najde, může si vybrat celou zobrazenou oblast nebo vybrat specifickou část zobrazené oblasti, podobně jako to umí OpenStreetMap při exportování viz 5.8.



Obrázek 5.7: Návrh levého panelu

Načítání

Speciální druh oznámení (v obrázku 5.6 označeno číslem 6), který se objeví ve středu obrazovky pouze po stisknutí tlačítek „Stáhnout“ nebo „Náhled“. Uživateli točícím se kolečkem oznamuje, že probíhá vytváření mapy. Zmizí po zahájení stahování nebo zobrazení náhledu.

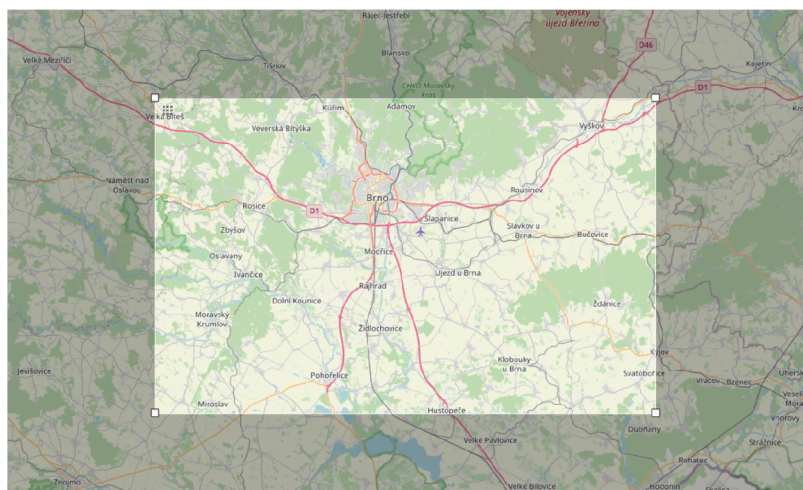
5.3 Návrh pro zpracování na serveru

Od serveru se očekává, že bude plnit prostředníka mezi klienty a asynchronním serverem, který bude pomocí virtuálních pracovníků vytvářet mapu na uživatelské přání.

Po přijetí požadavku k vytvoření náhledu předá pracovníkovi parametry mapy a oznámí mu, že se jedná o návrh. Pracovník zpracuje parametry a vytvoří png soubor s mapou z dat uchovávaných v databázi. Poté oznámí serveru, že je hotov. Server pošle mapu klientovi s informací, že posílá vytvořený návrh.

Se stahováním je proces stejný, akorát pracovník vytváří mapu v uživatelsky zvoleném formátu a velikosti.

Aby server mapy vytvářel rychle, bude potřeba mít databázi umístěnou na SSD disku a zajistit dostatek paměti RAM.



Obrázek 5.8: Ukázka výběru oblasti při exportu z OpenStreetmap.

Kapitola 6

Implementace

Tato kapitola čtenáři stručně zrekapituluje cíle implementace a potom popíše implementaci a fungování uživatelského rozhraní, webového serveru a proces tvorby mapy.

6.1 Cíle implementace

Cílem implementace je vytvořit webovou aplikaci, která umožní uživateli vytvářet více-méně slepé mapy.

Uživatel si bude moci vybrat oblast mapy, kterou chce. Pro tuto oblast si poté vybere prvky, které, pokud se na tomto území vyskytují, na výsledné mapě objeví. Uživatel také může tyto prvky editovat, tím je myšleno např. měnit barvu, velikost a u některých prvků zobrazit jejich text. Bude mu také umožněno zobrazit alespoň část mapy, aby měl přehled, jak mapa vypadá. Když už je uživatel spokojen s výběrem a stylem prvků, vybere si jeden z rastrových nebo vektorových formátů, ve kterém bude mapa vytvořena, a jeden z nabízených rozměrů. Poté kliknutím odešle své nastavení na server, kde se uživatelské volby zpracují a vytvoří se podle nich mapa, která se pošle zpět na jeho počítač.

6.2 Popis zpracování

V této sekci jsou dále popsány: principy fungování některých implementovaných částí, konkrétní funkce, zajímavé nebo složitější části kódu.

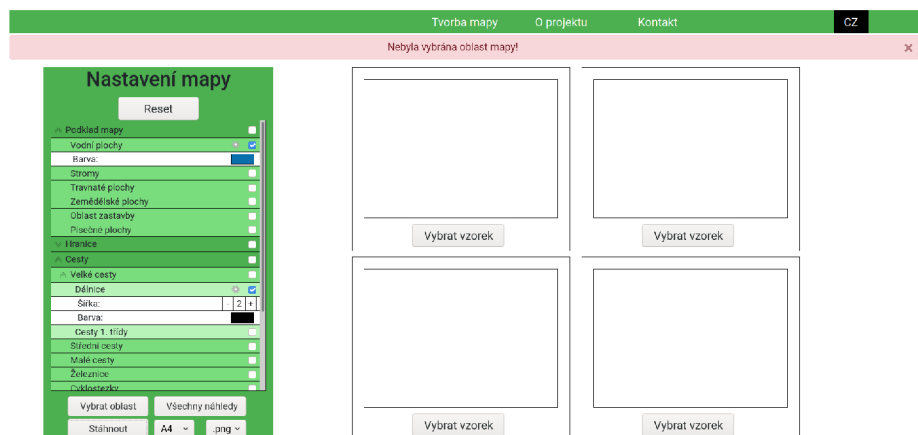
6.2.1 Klientská část aplikace

Tato část se věnuje implementaci klientské části aplikace. Nejprve popíše funkce, které mají na starost komunikaci se serverem, poté fungování komponenty, která získává od uživatele oblasti pro tvorbu mapy a náhledů, dále je uveden popis vzniku a fungování levého panelu a poslední část popisuje proces změny jazyka.

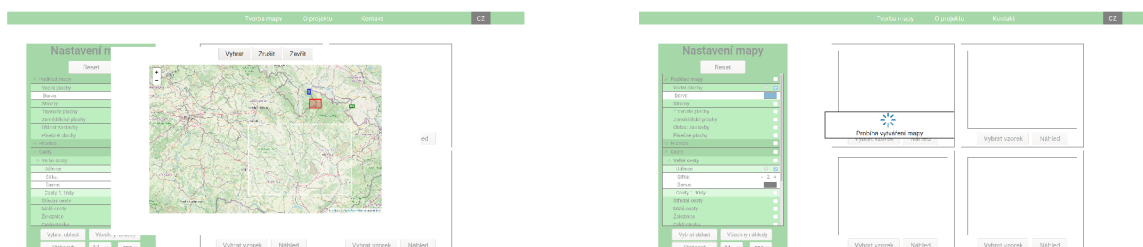
Implementace vizuálních prvků webové aplikace probíhala podle návrhu. Časem však byly upravovány nedokonalosti a design, vliv mělo i testování, jehož výsledky jsou popsány v kapitole 7. Poslední verze uživatelského rozhraní je k vidění na obrázcích 6.1 a 6.2.

Komunikace se serverem

Komunikace mezi klientem a serverem probíhá na straně klienta ve dvou částech programu, převážně pomocí WebSocketu.



Obrázek 6.1: Snímek obrazovky výsledného uživatelského rozhraní.



Obrázek 6.2: Snímky obrazovky výsledného uživatelského rozhraní. Na levém obrázku je prováděn výběr oblasti. Na pravém obrázku je zachyceno čekání na vytvoření mapy.

První částí, kde se funkce pro komunikaci nacházejí je háček **Effect**. Zde se určí, na která jména událostí, poslána skrze WebSocket spojení z webového serveru, má aplikace reagovat a jak.

První takováto událost má název „previewDone“, tato událost nastává vždy, když se dokončí některý z náhledů mapy. Obdržená data události pak obsahují ID náhledu a data náhledu. Pokud nedošlo k žádné chybě, je vytvořen nový objekt typu **Blob**, který obsahuje data náhledu a pro tento objekt se vytvoří URL adresa, která se vloží do příslušného zdroje náhledu, kterému patří.

Druhá událost, na který program reaguje je „downloadDone“. Tato událost oznamuje, že je připraven soubor ke stažení. Po obdržení této události posílá klientská část aplikace, pomocí knihovny **axios**¹, HTTP dotaz typu **GET** na server. Přijatá data jsou poté uložena do souboru.

Třetí a poslední událost má název „register_your_name“. Pokud nastane tato událost klient, tak klient na sever pošle událost „join“, která obsahuje řetězec s jeho ID.

Druhou část komunikace tvoří dvě funkce v těle komponenty. První funkce se jmenuje **createImage()** ta zkontroluje nastavení mapy a pokud jsou v pořádku posílá na server událost s názvem „download“, která obsahuje všechna potřebná data. Druhá funkce pak nese název **createPreviewImage()**, posílá událost „preview“ s daty potřebnými pro vytvoření náhledu.

¹<https://github.com/axios/axios>

Výběr oblasti mapy

Výběr oblasti zprostředkovává komponenta `<Bounds/>`. Tato komponenta pracuje s knihovnou React Leaflet², jejíž komponenty a funkce umožňují zobrazit a pracovat s mapou. Zde byla pro výběr oblasti použita mapa od OpenStreetMap.

Koordináty celé zobrazené mapy se ukládají, do dočasných souborů prohlížeče, po stisku tlačítka „Vybrat celou oblast“. Další možnosti výběru oblasti je stiskem tlačítka „Vybrat specifickou oblast“. Po stisku tohoto tlačítka je pomocí další komponenty z knihovny React Leaflet umístěn na mapu obdélník. Tomuto zobrazovanému obdélníku jsem nastavil různé vlastnosti a naslouchací funkce na události myši. Tyto úpravy umožňují jeho pohyb po mapě a změny jeho velikosti. Uživatel si tak může vybrat přesně takovou oblast, kterou potřebuje.

Levý panel

Levý panel vytváří komponenta `<MapSetting/>`. V této komponentě se funkcí `map()` prochází předaný řetězec ve formátu JSON (ukázka na obrázku 6.3) a podle míry zanoření a hodnot v klíčích se do proměnné vkládá HTML kód. Ten je pak součástí HTML kódu, který se vrací do hlavní komponenty. Zmíněný řetězec ve formátu JSON obsahuje tyto klíče a hodnoty:

- `id` – obsahuje řetězec s unikátním ID prvku mapy
- `value` – obsahuje řetězec, který je uvedený ve webové aplikaci
- `subgroup` – obsahuje pravdivostní hodnotu, která udává jestli se má vykreslit podskupina nebo členové skupiny
- `isChecked` – obsahuje pravdivostní hodnotu, která určuje, jestli danou skupinu, podskupinu nebo prvek uživatel označil (zaškrtnul)
- `nested` – obsahuje pole objektů, objekty mohou být buď další skupiny nebo koncové prvky
- `showColorPicker` – obsahuje pravdivostní hodnotu, která určuje, jestli má být zobrazena paleta pro výběr barvy
- `showSettings` – obsahuje pravdivostní hodnotu, která určuje, jestli mají být zobrazeny vlastnosti prvku
- `settings` – obsahuje pole objektů, které představují specifické nastavení pro prvky mapy např. velikost nebo barvu.

Obsah hodnot tohoto JSON řetězce je měněn podle akcí uživatele. Některé HTML elementy vytvořeného levého panelu mají přiřazeny JavaScriptové události `onClick` nebo `onChange`, které po příslušné akci zavolají jednu z funkcí a ta změní hodnotu v JSON objektu podle jeho ID. Například HTML elementy typu „checkbox“ reagují na událost `onChange`. Tato událost se spustí při každém kliknutí nebo odkliknutí zaškrťovacího pole a zavolá funkci `handleChangeChecked`. Tato funkce prochází přes celý řetězec JSON a hledá odpovídající JSON objekt podle ID. Po nalezení změní hodnotu klíče „checked“ na opačnou, takže z pravda na nepravda nebo naopak.

²<https://react-leaflet.js.org>

```

{"options": [
  {"id": "borders", "value": "Hranice", "subgroup": false, "isChecked": false,
  "nested": [
    {"id": "admin2", "value": "Států", "subgroup": false, "showColorPicker": false, "isChecked": false, "showSettings": false,
    "settings": [{ "size": 2 }, { "color": "#000" }], "nested": [] ]}]
}]

```

Obrázek 6.3: Ukázka řetězce ve formátu JSON, který byl použit v aplikaci.

Změna jazyka

Pro možnost měnit jazyky byly vytvořeny dva soubory: „cz.json“ a „en.json“. Každý z těchto souborů obsahuje jeden objekt typu JSON s klíči a hodnotami, kde hodnoty jsou řetězce s textem, který se má objevit ve webové aplikaci. Klíče jsou v obou souborech totožné, hodnoty významově taky, ale v jednom souboru jsou napsány česky a v druhém anglicky.

V hlavní komponentě se vždy jeden z těchto souborů načte do proměnné a z té se pak na určitých místech přistupuje pomocí klíčů k hodnotám. Po stisku tlačítka v horním menu aplikace se do proměnné načte vždy opačný soubor. Informace o jazyce je rovněž ukládána do dočasného souboru, takže i při příští návštěvě bude mít uživatel aplikaci ve zvoleném jazyce.

6.2.2 Webový server

Tato podsekcce popisuje funkce webového serveru, který byl napsán v jazyce Python s použitím pracovního rámce Flask a knihovny socketIO, která zprostředkovávala komunikaci s klientem přes protokol WebSocket.

funkce `connect()` a `onjoin_room()`

Když se klient připojí k serveru je mu poslána událost z funkce `connect()` se jménem „register_your_room“. Klient po obdržení této události pošle svoje ID, v obsahu události „join“, server na tuto událost zareaguje ve funkci `onjoin_room()`, vytvořením místnosti se jménem uživatelského ID, ve které bude pouze tento uživatel. Do takto vzniklé místnosti, jde poté jednoduše posílat další události bez přímého adresování klienta, stačí znát pouze jeho ID.

funkce `handle_download()` a `handle_preview()`

Funkce `handle_download()` je spouštěna událostí ze strany klienta s názvem „download“, která obsahuje: uživatelské ID, řetězec ve formátu JSON s nastavením všech prvků mapy, koordináty vybrané oblasti, velikost výsledné mapy a typ datového formátu mapy. Pokud jsou přijatá data v pořádku tak se zavolá funkce `create_image()`, která podle předaných parametrů vytvoří mapu.

Funkce `create_image()`, je zpracovávána asynchronní frontou úloh Celery. Při takovémto volání se požadavek na zpracování vkládá nejdříve do fronty a pokud je některý z virtuálních pracovníků volný, tak úlohu z fronty zpracuje.

Po zavolání funkce `create_image()`, se do proměnné `result` uloží objekt odkazující na specifickou úlohu. Tento objekt je poté vložen do seznamu `work_list` společně s ID uživatele a řetězcem „downloadDone“. Funkce `handle_preview()` funguje podobně jako `handle_download()`. Rozdílem je pouze to, že funkce dostává o jeden argument více a tím je ID náhledu a do funkce `create_image()`, předává řetězec „previewDone“.

funkce `background_thread()`

Tato funkce běží zvlášť v samostatném vlákně a kontroluje, jestli už je některá úloha hotová. Kontrola je prováděna nekonečnou smyčkou, ve které jsou iterovány objekty úloh ze seznamu `work_list`. U těchto objektů se kontroluje výstup metody `ready()`, která vrací pravdu, jestli je úloha hotová.

Pokud je hotová, tak pokud šlo o úlohu pro vytvoření náhledu, je klientovi náhled poslán přímo přes WebSocket spojení, událostí se jménem „previewDone“ a zprávou, která obsahuje ID a data náhledu.

Jestli byla dokončena úloha, jejíž výsledkem má být výsledná mapa. Je uživateli poslána událost s názvem „downloadDone“ a zprávou s názvem souboru. Klient pak pošle HTTP požadavek GET, na jedenu z cest, podle formátu souboru. Odsud jim je soubor poslán Flask funkcí `send_file()` nebo `send_from_directory()`.

6.2.3 Vytvoření obrázku mapy

Tvorbu mapy provádí virtuální pracovník asynchronní fronty úloh Celery. Práce začíná ve funkci `create_image()`, v souboru `main.py`. Do této funkce se předávají parametry: uživatelské ID, řetězec ve formátu JSON, který obsahuje nastavení prvků mapy, koordináty oblasti vybrané uživatelem, velikost výsledné mapy, formát výsledné mapy a pokud se jedná o náhled, tak jeho ID.

Na začátku funkce se do proměnných přiřadí nastavení mapových projekcí (toto nastavení bylo převzato ze stránky epsg.io³), název výsledné mapy (spojení uživatelského ID a označení výsledného datového formátu, pokud jde o náhled, tak je přidáno i ID návrhu) s cestou a rozměry mapy.

Poté se inicializuje objekt třídy `Map` z Python modulu `Mapnik`. Tomuto objektu se nastaví rozměry mapy, barva pozadí a výsledné mapové zobrazení. Tímto zobrazením bylo zvoleno Mercatorovo zobrazení, jakožto univerzální zobrazení, které je pro menší oblasti docela přesné. Poté se zavolá funkce `checked()` ze souboru `generate_map.py`, té se předá jako parametr již zmíněný řetězec ve formátu JSON a vytvořený objekt třídy `Map`.

Na začátku tohoto souboru se nachází slovník `data`, ve kterém je ke každému klíči přiřazen seznam. Tento seznam na prvním místě obsahuje typ prvku (`line`, `point`, `polygon`, `text`) a na druhém dotaz na databázi.

Ve funkci `checked()`, se prochází ve smyčce předaný řetězec JSON a porovnávají se ID prvků označených s atributem `checked`, který je pravda, s ID ve slovníku `data`. Pokud jsou obě ID totožná a prvek má být na mapě, tak se podle typu ve slovníku, zavolá buď funkce pro vytvoření linií (`draw_lines()`), polygonů (`draw_polygons()`), bodů (`draw_points_from_xml()`) nebo textu (`draw_text_from_xml()`). Do všech těchto funkcí se předávají parametry: vytvořený objekt třídy `Map`, ID prvku, dotaz na databázi ze slovníku a pole JSON objektů obsahující další objekty s nastavením.

Funkce `draw_lines()` a `draw_polygons()` fungují na stejném principu, který je popsán u `Mapnik XML` stylu v sekci 4.6, jenom se zde využívají funkce z Python modulu `Mapnik` a není tak potřeba `Mapnik XML` předpis. Na začátku těchto dvou funkcí se vždy inicializují objekty pro styl, pravidlo a příslušný symbolizér. Symbolizéru se nastaví atributy vzhledu podle nastavení z objektu JSON. Nastavený symbolizér se pak vloží metodou `append()` do pravidla, pravidlo do stylu a styl do mapy. Poté se inicializuje objekt pro vrstvu, kterému se nastaví mapová projekce, ve které jsou data, a zdroj dat.

³<https://epsg.io>

U funkcí `draw_points_from_xml()` a `draw_text_from_xml()` jsem musel kvůli chybějícím a nefunkčním vázaným funkcím v Python modulu `Mapink`, zvolit jiný přístup, než u funkcí `draw_lines()` a `draw_polygons()`. Do proměnných se ukládá řetězec s Mapnik XML kódem, který se dynamicky nastaví podle nastavení v JSON objektu. Tento jeden dlouhý Mapnik XML řetězec, představující nastavený styl i vrstvu, se pak vloží do objektu mapy metodou `load_map_from_string()`. Ikony, které se při vykreslování bodů využívají, jsem převzal z projektu `OpenStreetMap Carto`⁴.

Poté co se všechny chtěné prvky zpracují, tak se vrátí objekt třídy `Map` zpět do funkce `create_image()`. Zde se ještě nastaví pohled mapy na souřadnice oblasti zadané uživatelem a pokud vše proběhlo, tak jak mělo, tak funkcí `render_to_file()`, které se předá vrácený objekt třídy `Map`, vytvořený název s systémovou cestou a formát výsledného souboru (png, svg, pdf, jpeg). Pokud i toto vytvoření proběhne správně, tak funkce skončí, vrátí „OK“ a v zadané cestě je vytvořená mapa.

⁴<https://github.com/gravitystorm/openstreetmap-carto/tree/master/symbols>

Kapitola 7

Průběh testování a zhodnocení výsledků

Tato kapitola popisuje průběh celého procesu testování. Popisuje, jakým způsobem probíhala jednotlivá kola testování, co bylo cílem, co se v nich testovalo a jaké byly výsledky, případně provedené změny. Závěru této kapitoly pak patří zhodnocení testování a návrh dalšího.

7.1 Testování

Testování probíhalo s několika potencionálními uživateli z cílové skupiny, popsané v podpodsekcí 5.1.1. V každém testovacím kole probíhal kontakt s účastníky testování jednotlivě v rozestupech několika minut až dnů. Popisované změny byly vždy uskutečněny až po skončení celého testovacího kola.

1. testování

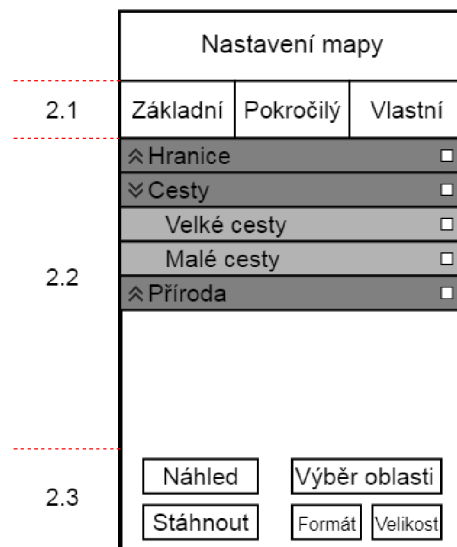
První testování probíhalo formou online hovorů. Uživatelům účastnících se testování byl slovně popsán návrh webového rozhraní a poskytnuty obrázky vytvořeného návrhu. Cílem tohoto testování bylo zjistit první reakce uživatelů a eliminovat prvky uživatelského rozhraní, které by mohly působit neintuitivně.

Návrh uživatelského rozhraní sklidil celkem kladné reakce. Uživatelům, kteří se účastnili i sběru požadavků a líbil se jim výběr prvků v aplikaci map-stylizer, oceňovali stejnou jednoduchost a vyzdvihovali rozdělení možností výběru do smysluplnějších skupin. Nejčastěji zmiňovaným problémem byla nemožnost vizuálně rozpoznat konečnou možnost výběru od skupiny možností, přesněji jestli se po kliknutí v levém panelu zobrazí další možnosti.

Tento nedostatek jsem se pokusil vyřešit přidáním ikon před názvy skupin možností viz obrázek 7.1. Ikona by měla tedy tuto možnost symbolizovat a měnit svůj vzhled, podle toho, jestli je skupina zobrazená nebo není. K tomu problému jsem si ještě poznamenal, že v implementaci by neměla chybět změna vzhledu kurzoru myši.

2. testování

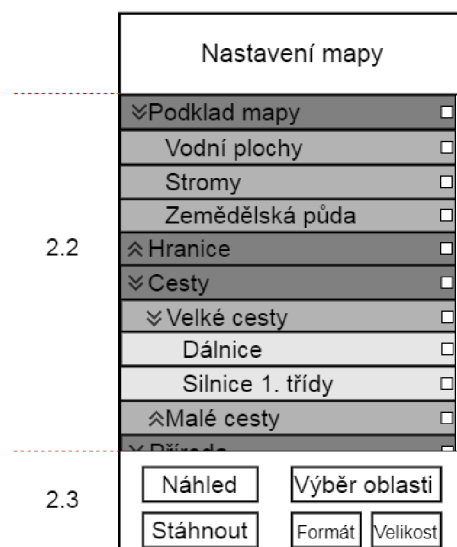
Druhé kolo testování už probíhalo interaktivně, testovalo se implementované uživatelské rozhraní, které bylo umístěno na webovém hostingu. Toto uživatelské rozhraní mělo vzhled návrhu s úpravami z 1. kola testování.



Obrázek 7.1: Změny návrhu levého menu po prvním testování

S uživateli jsem byl v průběhu tohoto testování opět v přímém online spojení, po poslání odkazu na stránku měli za úkol si stránku projít a tzv. „proklikat“. Dalším úkolem bylo provést proces vytvoření mapy, který mi uživatelé slovně popisovali. Pokaždé, když nastaly nějaké nejasnosti, došlo k vysvětlení a diskusi, proč se tak stalo.

Při testování se narazilo na několik chyb v zobrazování, když se některé prvky uživatelského rozhraní posunuly tam, kam neměly. Kromě těchto chyb se přišlo na to, že tlačítka, která měla práci uživateli usnadnit (na obrázku 7.1 označeno 2.1), nejsou téměř vůbec využívána a jsou spíše kontraproduktivní. Většina uživatelů si buď volila mapu bez podkladů, nebo volila vlastní nastavení. Po těchto zjištěních došlo k jejich odstranění a skryté možnosti byly začleněny do normálního zobrazení levého panelu. Na náměty uživatelů byly přidány další možnosti přidání prvků do mapy. Provedené změny lze vidět na obrázku 7.2.



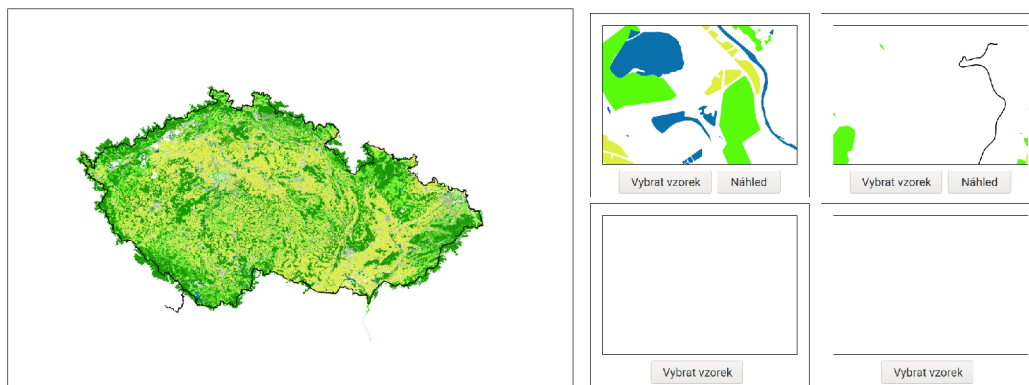
Obrázek 7.2: Změny návrhu levého menu po druhém testování

3. testování

Ve třetím kole už si uživatelé mohli vyzkoušet funkční aplikaci. Tato fáze zkoumala celkovou funkčnost aplikace a případné chyby, které by se mohly objevit. Hostování probíhalo z osobního počítače pomocí služby ngrok¹.

Uživatelům byl vždy na začátku poslán odkaz, pro připojení k aplikaci. Poté, se sdílenou obrazovkou v online hovoru, měli jednoduchý úkol vybrat si zajímavou oblast, kterou naposledy navštívili a vybrat pro zobrazení alespoň 3 prvky mapy, z nichž alespoň dva vzhledově upraví. Výslednou mapu si pak měli stáhnout v jakém formátu a velikosti chtěli.

Výsledky tohoto testování byly celkem uspokojivé a názory většiny uživatelů kladné, podařilo se najít několik prvků, které se kvůli stejné chybě v dotazech na databázi nevykreslily. Také byly znovu přidány další možnosti přidání prvků do mapy, z podnětů uživatelů. Největší změnou, která z tohoto testování vzešla, bylo přepracování konceptu náhledového okna mapy, které se v něm zobrazovaly, byly pro velké oblasti větší, než se předpokládalo (desítky MB). Proto byl jeden velký náhled nahrazen až čtyřmi menšími, do kterých si oblasti uživatel vybírá zvlášť (provedenou změnu, lze vidět na obrázku 7.3). Při těchto výběrech je nastaven maximální limit velikosti oblasti a při vytváření náhledu je jeho velikost nastavena na stejnou velikost jako náhledového okna.



Obrázek 7.3: Náhled před změnou (první obrázek z levé strany) a po změně (druhý obrázek z levé strany)

7.2 Zhodnocení testování

Testování, které proběhlo, prokázalo, že aplikace je v základu funkční. Podařilo se najít některé chyby, které při vytváření vznikly a opravit je.

Vývoj i testování probíhaly, kvůli velikosti dat, pouze s daty z území České republiky a s hostováním na průměrně výkonném osobním počítači, kterému byla přizpůsobena částečně i implementace. Proto si myslím, že další práce a testování by se měly zabývat rozšířením a přizpůsobením na větší oblasti, hostované na opravdovém serveru.

Dále z testování vyplynulo, že výsledné mapy (hlavně ve vektorových formátech) můžou mít větší velikost než se očekávalo. I z tohoto důvodu by se na výkonnějším serveru mělo testovat s většími oblastmi a navrhnout na základě výsledků postup úpravy např. pokud bude

¹<https://ngrok.com>

oblast moc velká, tak se mapa rozdělí na části a ty se pošlou uživateli v komprimovaném archívu.

Kapitola 8

Závěr

Cílem této bakalářské práce bylo vytvořit webovou aplikaci, která by uživatelům umožnila vytvářet a stahovat více-méně slepé mapy, v různých datových formátech a velikostech, jednoduše a bez nutnosti stahování aplikací a objemných zdrojových dat. Takovouto webovou aplikaci se podařilo úspěšně vytvořit.

Podle požadavků a preferencí potenciálních uživatelů byl vytvořen návrh vzhledu a částečně i fungování vyvíjené aplikace. Návrh byl prvním subjektem testování, druhým bylo uživatelské rozhraní a posledním fungování celé aplikace. Po každém testovacím kole proběhlo vyhodnocení a úpravy. Výsledkem je webová aplikace, která umožní uživateli vybrat si oblast, jednotlivé prvky mapy nebo skupiny prvků mapy, upravit vzhled zobrazených prvků, výsledný datový formát mapy a její velikost.

Největší zradou, která v průběhu implementace nastala, byly chybějící vazby funkcí nástroje Mapnik na programovací jazyk Python, ale i tuto nepříjemnost se podařilo vyřešit. Pokud by v dalších verzích nástroje Mapnik došlo k opravě a doplnění těchto funkcí, lze jednoduše tuto část upravit.

Jako další rozšíření a vylepšení této práce bych navrhoval začlenění do informačního systému. Takovéto začlenění by uživatelům umožnilo po registraci ukládat a poté znovu načítat své mapové styly, oblasti nebo už i hotové mapy. Ukládání vlastních stylů bylo několikrát i zmíněno při testování ze strany potenciálních uživatelů. Dalším drobným rozšířením by mohla být možnost, alespoň do rastrových obrázků, přidávat vlastní poznámky, formou digitální tužky nebo pastelky.

Literatura

- [1] BECONYTE, G. a VILIUVIENE, R. The concept and importance of style in cartography. *Geodezija ir Kartografija*. Taylor & Francis. 2009, sv. 35, č. 3, s. 82–91.
- [2] BENNETT, J. *OpenStreetmap: Be your own Cartographer*. 1. vyd. Packt Publishing, 2010. ISBN 978-1-847197-50-4.
- [3] DALIBOR JANAK. *What are vector tiles and why you should care* [online]. 7. únor 2019 [cit. 2021-05-03]. Dostupné z: <https://www.maptiler.com/news/2019/02/what-are-vector-tiles-and-why-you-should-care/>.
- [4] DANIEL R. STREBE. *Mercator projection* [online]. 16. prosinec 2011 [cit. 2021-05-03]. Dostupné z: <https://commons.wikimedia.org/w/index.php?curid=17700069>.
- [5] ESRI. *Map Projections* [online]. [cit. 2021-05-03]. Dostupné z: <https://www.arcgis.com/apps/MapJournal/index.html?appid=31484c80dba54a058369dfb8e9ced549>.
- [6] GISMENTORS. *Mercatorovo zobrazení* [online]. 27. listopad 2020 [cit. 2021-05-03]. Dostupné z: <https://training.gismentors.eu/open-source-gis/soursystemy/mercatorovo-zobrazeni.html>.
- [7] GISMENTORS. *Souřadnicové systémy* [online]. 27. listopad 2020 [cit. 2021-05-03]. Dostupné z: <https://training.gismentors.eu/open-source-gis/soursystemy/index.html>.
- [8] GRINBERG, M. *Flask Web Development*. 1. vyd. O'Reilly Media, Inc., 2014. ISBN 978-1-449-37262-0.
- [9] ABSOLUTE TINKERER, U. *Map-stylizer* [online]. 30. červen 2020 [cit. 2021-04-27]. Dostupné z: <https://github.com/Absolute-Tinkerer/map-stylizer/blob/master/README.md>.
- [10] BREJC, I. *Maperative* [online]. [cit. 2021-04-27]. Dostupné z: <http://maperitive.net/>.
- [11] OPENSTREETMAP CONTRIBUTORS. *OpenStreetMap wiki* [online]. [cit. 2021-04-27]. Dostupné z: https://wiki.openstreetmap.org/wiki/Main_Page.
- [12] UŽIVATEL BLACKADDER. *OpenStreetMap wiki – Stats* [online]. Rev. 18. březen 2021 [cit. 2021-04-27]. Dostupné z: https://wiki.openstreetmap.org/wiki/File:Osmbdstats1_users.png.

- [13] PAVLÍČEK, F. *Analýzy generování rastrových a vektorových mapových dlaždic*. 2019. Diplomová práce. Univerzita Palackého v Olomouci. Přírodovědecká fakulta. Katedra geoinformatiky.
- [14] QGIS DEVELOPMENT TEAM. *Geographic Information System README* [online]. 16. březen 2021 [cit. 2021-04-28]. Dostupné z:
<https://github.com/qgis/QGIS/blob/master/README.md>.
- [15] QGIS DEVELOPMENT TEAM. *Geographic Information System User Manual* [online]. Rev. 18. květen 2021 [cit. 2021-04-28]. Dostupné z:
https://docs.qgis.org/3.16/en/docs/user_manual/.
- [16] STEFANAKIS, E. Web mercator and raster tile maps: two cornerstones of online map service providers. *GEOMATICA*. 2017, sv. 71, č. 2, s. 100–109.
- [17] VANESSA WANG, P. M. *The Definitive Guide to HTML5 WebSocket*. 1. vyd. Apress, 2013. ISBN 978-1-4302-4740-1.

Příloha A

Obsah přiloženého paměťového média

Na přiložené SD kartě se nachází tyto adresáře:

- dokumentace – obsahuje písemnou zprávu ve formátu PDF
- latex – obsahuje zdrojový tvar písemné zprávy
- aplikace – obsahuje soubory webové aplikace
- prilohy – obsahuje plakát a demonstrační video
- navod – obsahuje návod na sestavení aplikace a seznam verzí nástrojů, knihoven