

Univerzita Palackého v Olomouci
Přírodovědecká fakulta
Katedra geoinformatiky

**WEBOVÁ APLIKACE PRO VIZUALIZACI
NAVŠTÍVENÝCH MÍST**

Bakalářská práce

Dominik VÍT

Vedoucí práce Ing. Jan Piňos

Olomouc 2020
Geoinformatika a geografie

ANOTACE

Bakalářská práce se zabývá vývojem webové aplikace pro vizualizaci navštívených míst a je rozdělena na dvě části. V teoretické části se autor zabývá rozborem stávajících řešení na poli aplikací pro vizualizaci navštívených míst. Jsou popsány druhy možných technologií pro vývoj vlastního řešení webové aplikace a rozebrány rozdíly oproti konkurenčním řešením. V praktické části se autor zabývá samotným vývojem vlastního řešení jak s ohledem na zadání, tak na vlastní uvážení. Dle nabytých poznatků pak autor na konci práce uvádí doporučené technologie pro vývoj takové aplikace pro produkční prostředí a diskutuje další možnosti rozvoje.

KLÍČOVÁ SLOVA

Webová aplikace; vizualizace míst; historie polohy; uživatelé; OpenLayers

Počet stran práce: 50

Počet příloh: 2 (z toho 1 volná a 1 elektronická)

ANOTATION

This bachelor thesis focuses on developing a web application for visualization of visited places and is divided to two parts. In the theoretical part, the author deals with the analysis of existing solutions for the visualization of visited places. Next, available technologies for the development of an own web application solution are described and the differences to competing solutions are discussed. In the practical part, the author deals with the development of their own solutions, according to the assignment as well as the author's discretion. Based on the acquired knowledge, the author at the end of the work lists the recommended technologies for the development of such an application in a production environment and discusses other development options.

KEYWORDS

Web app, location visualization, location history, users, OpenLayers

Number of pages: 50

Number of appendixes: 2

Prohlašuji, že

- bakalářskou práci včetně příloh, jsem vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu.

- jsem si vědom, že na moji bakalářskou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,

- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou/diplomovou práci užívat (§ 35 odst. 3),

- souhlasím, aby jeden výtisk bakalářské práce byl uložen v Knihovně UP k prezenčnímu nahlédnutí,

- souhlasím, že údaje o mé bakalářské práci budou zveřejněny ve Studijním informačním systému UP,

- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užít výsledky a výstupy mé bakalářské práce v rozsahu § 12 odst. 4 autorského zákona,

- použít výsledky a výstupy mé bakalářské práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Olomouci dne _____

podpis autora: _____
plné jméno autora: Dominik Vít

Děkuji vedoucímu práce Ing. Janu Piňosovi za podněty, připomínky a vstřícný přístup při vypracování práce. Dále děkuji konzultantu Vítu Horáčkovi za cenná doporučení při tvorbě webové aplikace.

UNIVERZITA PALACKÉHO V OLOMOUCI

Přírodovědecká fakulta

Akademický rok: 2018/2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Dominik VÍT**
Osobní číslo: **R17435**
Studijní program: **B1301 Geografie**
Studijní obor: **Geoinformatika a geografie**
Téma práce: **Webová aplikace pro vizualizaci navštívených míst**
Zadávající katedra: **Katedra geoinformatiky**

Zásady pro vypracování

Cílem práce je vytvořit webovou aplikaci, která obsahuje prvky mapové sociální sítě. Aplikace by měla pro jednoho uživatele zaznamenávat polohu navštívených míst v mapě, k tomuto záznamu přidat časový údaj a pomocí toho tyto informace zobrazovat v různých časových úsecích, a nakonec umožnit spojení takto do mapy zanesených informací pro dva a více uživatelů. V teoretické části student provede průzkum trhu a podrobně popíše neznámější, již existující, webové a mobilní aplikace, které zaznamenávají polohu navštívených míst, a mají prvky sociální sítě (Google Timeline, Polarsteps a jiné). V praktické části bude vytvořen jednoduchý prototyp navržené webové aplikace s prvky mapové sociální sítě. Tento prototyp bude obsahovat mapové, databázové a programové řešení. Tato řešení mohou být provedena v nejjednodušší možné míře, nicméně musí být funkční. Na konci práce řešitel uvede technologie (mapové, databázové, programové), které by bylo možné použít pro vytvoření navržené mapové sociální sítě v produkčním prostředí, porovná různé alternativy řešení a navrhne technologie, jejichž využití vidí jako nevhodnější.

Celou práci, tj. text včetně všech příloh, posteru, výstupů, zdrojových i vytvořených dat, map, programových kódů a databází, student odevzdá v digitální podobě na paměťovém nosiči připevněném k deskám práce s popisem (jméno, název práce, Katedra geoinformatiky UP, rok). Text práce s přílohami odevzdá ve dvou svázaných výtiscích na sekretariát katedry ve stanoveném termínu. O práci student vytvoří webovou stránku v souladu s pravidly dostupnými na stránkách katedry. Práce bude zpracována podle obecných zásad (Voženílek, 2002) a závazné šablony pro kvalifikační práce na KGI. Povinnou přílohou práce je poster formátu A2.

Rozsah pracovní zprávy: **max. 50 stran**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

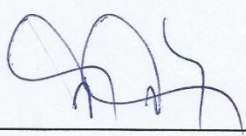
LONGLEY, P. A., GOODCHILD, M. F., MAGUIRE, D. J., RHIND, D. W. (2005): Geographic information systems and science. John Wiley & Sons.
FU, P., SUN, J. (2010): Web GIS: principles and applications. Esri Press.
VOSSSEN, G., HAGEMANN, S. (2010): Unleashing Web 2.0: From concepts to creativity. Elsevier.
VOŽENÍLEK, V. (2002): Diplomové práce z geoinformatiky. Vydavatelství Univerzity Palackého, Olomouc, UP, 31 s.

Vedoucí bakalářské práce: **Ing. Jan Piňos**
Katedra geoinformatiky

Datum zadání bakalářské práce: 6. května 2019
Termín odevzdání bakalářské práce: 6. května 2020

L.S.

doc. RNDr. Martin Kubala, Ph.D.
děkan



prof. RNDr. Vít Voženílek, CSc.
vedoucí katedry

V Olomouci dne 2. září 2019

OBSAH

| | |
|---|-----------|
| SEZNAM POUŽITÝCH ZKRATEK | 10 |
| ÚVOD | 11 |
| 1 CÍLE PRÁCE..... | 12 |
| 2 METODY A POSTUPY ZPRACOVÁNÍ..... | 13 |
| 2.1 Použité metody | 13 |
| 2.1.1 OpenLayers..... | 13 |
| 2.1.2 HTML a CSS | 14 |
| 2.1.3 Javascript..... | 14 |
| 2.1.4 MySQL..... | 15 |
| 2.1.5 JSON a GeoJSON..... | 15 |
| 2.1.6 Sublime Text..... | 15 |
| 2.2 Postup zpracování..... | 16 |
| 3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY | 18 |
| 3.1 Aplikace pro vizualizaci míst nebo pohybu..... | 19 |
| 3.1.1 Google Maps Timeline | 19 |
| 3.1.2 Foursquare Swarm..... | 21 |
| 3.1.3 Facebook Local | 22 |
| 3.1.4 Polarsteps | 23 |
| 4 VÝVOJ APLIKACE MAPLAY | 24 |
| 4.1 Přípravné práce na aplikaci..... | 24 |
| 4.1.1 Založení MySQL databáze..... | 25 |
| 4.1.2 Přidání mapového okna do aplikace..... | 27 |
| 4.2 Přidávání bodů a kolekcí..... | 28 |
| 4.2.1 Přidávání bodů kliknutím do mapy..... | 30 |
| 4.3 Vizualizace navštívených míst v mapě | 34 |
| 4.3.1 Vyhledávání uživatelů a zobrazení jejich kolekcí..... | 35 |
| 4.3.2 Základní zobrazení bodů | 37 |
| 4.3.3 Funkce rerenderMap | 37 |
| 4.3.4 Bounding box..... | 37 |
| 4.3.5 Funkce updateForm – časová osa | 38 |
| 4.3.6 Funkce drawLine | 39 |
| 4.3.7 Smazání a výběr kolekcí | 39 |
| 4.3.8 Funkce getNewCollectionList | 40 |
| 4.3.9 Funkce Tour, flyTo a krokování | 40 |
| 4.4 Uživatelské účty..... | 41 |
| 4.4.1 Registrace a přihlašování..... | 42 |
| 4.4.2 PHP sessions..... | 42 |
| 4.4.3 Stránka acc-profile.php..... | 43 |
| 4.5 Editace bodů | 43 |
| 5 VÝSLEDKY | 45 |
| 5.1 Výsledky vývoje aplikace | 46 |
| 5.2 Návrh pro produkční řešení..... | 46 |
| 5.2.1 Mapové produkty | 46 |

| | | |
|----------|---|-----------|
| 5.2.2 | Databáze..... | 47 |
| 5.2.3 | Programová řešení..... | 48 |
| 6 | DISKUZE | 49 |
| 7 | ZÁVĚR | 51 |
| | POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE | |
| | PŘÍLOHY | |

SEZNAM POUŽITÝCH ZKRATEK

| Zkratka | Význam |
|----------------|---|
| AJAX | Asynchronous JavaScript and XML |
| API | Application programming interface |
| CRUD | Create, Read, Update, Delete |
| CSS | Cascading Style Sheets |
| EPSG | European Petroleum Survey Group |
| FOSS | Free and OpenSource Software |
| GIS | geografický informační systém |
| GPS | Global Positioning System |
| GPX | GPS Exchange Format |
| HTML | HyperText Markup Language |
| JS | JavaScript |
| JSON | JavaScript Object Notation |
| OL | OpenLayers |
| PHP | Hypertext Preprocessor |
| SQL | Structured Query Language |
| URL | Uniform Resource Locator |
| UI | User Interface (uživatelské rozhraní) |
| UX | User Experience (uživatelská zkušenost) |
| WFS | Web Feature Service |
| WMS | Web Map Service |

ÚVOD

Odjakživa se lidé v různé míře zajímají o to, jaká místa navštívili ať už ve svém životě celkově nebo třeba loňské léto. Je velmi příjemné podívat se opět na navštívená místa a prožít zážitky s nimi spojená. Dříve k tomu mohly sloužit poznámky v mapách či mapy různých oblastí samotné. V době moderních technologií nám k tomu slouží v masové míře především mobilní telefony, které mnohokrát nezištně data o naší poloze ukládají.

Práce se ale nezabývá tím, k čemu všemu musí dojít, aby se takové údaje o poloze uložily. Jejím posláním je vyvinout webovou aplikaci pro vizualizaci navštívených míst. K tomu autor použil dostupné prostředky z oblasti otevřeného softwaru. Místa uživatel může nahrát do aplikace různými způsoby. Tam je pak shromažďovat do skupin, tzv. kolekcí. V případě, že si chce zobrazit data od jiných uživatelů, tak může využít funkcionality vyhledávání podle uživatelského jména a zobrazit veřejně viditelné kolekce vyhledaných uživatelů. Ty pak připojit k vizualizaci do své mapy. Ta probíhá chronologicky podle zadaného času u jednotlivých míst. Sestavená aplikace tedy jednoduchým způsobem ukazuje uživateli jeho historii polohy pomocí jednoduché ale efektní vizualizace.

Aplikace je čistě webová, neexistuje k ní desktopová či mobilní alternativa. Součástí práce je rešerše, ve které je popsán stav současné problematiky tzv. geosociálních sítí a jejich nejznámějších zástupců. Autor porovnává stávající řešení na trhu a popisuje jejich různé zaměření. V teoretické části je dále popsán postup a technologie pro vývoj vlastní geosociální aplikace. Na konci práce pak popisuje prostředky, které by byly pro vývoj nové aplikace pro vizualizaci míst ideální pro produkční prostředí.

1 CÍLE PRÁCE

Cílem práce je vytvořit webovou aplikaci, která obsahuje prvky mapové sociální sítě. Aplikace umožňuje pro jednoho uživatele nahrát polohu navštívených míst v mapě, k tomuto záznamu přidat časový údaj a název bodu a pomocí vizualizací tyto informace zobrazovat podle různého seskupení bodů či všechny body zároveň, a nakonec umožnit spojení takto do mapy zanesených informací pro dva a více uživatelů. V teoretické části autor provede průzkum trhu a podrobně popíše nejznámější, již existující, webové a mobilní aplikace, které zaznamenávají a vizualizují polohu navštívených míst, a mají prvky sociální sítě (Google Timeline, Polarsteps a jiné). V praktické části bude vytvořen jednoduchý prototyp navržené webové aplikace s prvky mapové sociální sítě. Tento prototyp bude obsahovat mapové, databázové a programové řešení. Tato řešení mohou být provedena v nejjednodušší možné míře, nicméně musí být funkční. Na konci práce autor uvádí technologie (mapové, databázové, programové), které by bylo možné použít pro vytvoření navržené mapové sociální sítě v produkčním prostředí, porovnává různé alternativy řešení a navrhuje technologie, jejichž využití vidí jako nejvhodnější.

2 METODY A POSTUPY ZPRACOVÁNÍ

Vypracování této práce začalo průzkumem zkoumané problematiky ve stávajících komerčních řešeních na trhu. V rámci této první části rešerše došlo k vymezení následujících procesů z hlediska odlišení vůči stávajícím produktům. Druhá část rešerše se zabývala nekomerčními produkty, které alespoň částečně obsahovaly podobné technické řešení jako vyvíjená aplikace. Následovala velmi podstatná část, a to osvojení si programovacích a databázových jazyků, díky kterým byla tato aplikace vytvořena. Autor si také na začátku práce vyzkoušel různé programové prostředí určené k vývoji takových aplikací a vybral pro něho ta nejvhodnější. Všechny výše popsané náležitosti jsou rozebrány detailněji v podkapitolách.

2.1 Použité metody

V této části jsou popsány všechny použité metody, díky kterým vznikla konečná podoba vyvinuté aplikace, a to jak z mapového, serverového, programovacího tak i databázového řešení. Pro mapové řešení byla zvolena JavaScript knihovna OpenLayers ve verzi 6.3.1. pro naprogramování prostředí na straně klienta a výsledný vzhled aplikace bylo použito jazyka HTML, CSS, JavaScript, jQuery a AJAX. Pro dynamické vykreslování obsahu stránek a stěžejních částí aplikace bylo použito jazyka PHP. Testování aplikace probíhalo nejdříve na lokálním serveru prostřednictvím balíčku XAMPP. Pro uložení dat o uživateli a dat, jež uživatelé vytvořili a používali bylo využito databáze MySQL v prostředí phpMyAdmin. Drtivá většina zdrojového kódu aplikace byla napsána v programu Sublime Text ve verzi 3.

2.1.1 OpenLayers

OpenLayers je volně dostupná mapová knihovna v jazyce JavaScript umožňující tvorbu interaktivních dynamických webových map a mapových produktů, která využívá mapovou databázi OpenStreetMap pro výchozí zobrazení podkladové mapy. Dále je možné zobrazovat mapová data ze zdrojů jako například Bing, MapBox, Esri či jiné dostupné XYZ dlaždicové zdroje. Také je možné připojit různé WMS či WFS služby. Do OpenLayers lze nahrát vektorová data z formátů GeoJSON, TopoJSON, GML, GPX nebo KML. (OpenLayers.org, 2019). Řešení také podporuje asynchronní načítání dat. Knihovna OpenLayers je vyvíjena open-source komunitou a distribuována bezplatně pod dvoustupňovou BSD licenci.

Historii OpenLayers představovalo několik milníků. Knihovna vznikla v roce 2006 pod záštitou společnosti MetaCarta jako alternativa Google Maps. O rok později se přesouvá pod hlavičku Open Source Geospatial Foundation (OSGeo). Následující léta však ukázala, že je projekt problémový a ve druhé verzi se vývoj zastavil. Po vzniku The OpenLayers Dev Team došlo ke vzkříšení ve verzi 3, které následovalo ještě mnoho dalších pokusů o získání uživatelů. Dlouhé prodlevy, několikeré plné spuštění knihovny a nekompatibilita

s OL v.2 mělo za následek hromadný odliv vývojářů k alternativním platformám. (Nétek,2015)

Aktuální verze OpenLayers je 6.3.1 s níž také byla vytvořena webová aplikace. Velkou výhodou této knihovny je široká uživatelská komunita a také mnoho příkladu užití na stránkách dokumentace OpenLayers. V neposlední řadě je to také podpora mnoha různorodých rozšíření třetích stran. Autor nicméně v průběhu tvorby aplikace doznal, že kvůli již popsané nekompatibilitě mezi staršími verzemi či složitosti základních prvků (přepínač podkladových map, vyhledávání geolokací atp.) oproti konkurenčním řešení, především Leafletu, dochází k mnoha komplikacím, které měly za důsledek pomalejší programování aplikace. Důvodem pro výběr této knihovny bylo především různorodá nabídka rozšiřujících funkcí, které autor považoval za velmi užitečné do finální aplikace. Nicméně například kvůli dynamickému získávání dat z databáze došlo k tomu, že se nakonec taková rozšiřující funkce použila jen jedna. Veškeré uvedené příklady na stránkách OpenLayers používají statická data nahraná přímo ve zdrojovém kódu (např. souřadnice v jedné proměnné, nikoliv nahrávané z externího zdroje). Všechny ostatní musely být vytvořeny na míru aplikace či zcela přepracovány. Víz v kapitole 5.2 Návrh produkčního řešení.

2.1.2 HTML a CSS

HyperText Markup Language je značkovací jazyk pro tvorbu webových stránek. Je vyvíjen konsorciem W3C a jeho současná stabilní verze je 5.2. Slouží k vytváření základní kostry webových stránek. V dnešní době neslouží k formátování vzhledu internetových stránek. K tomu slouží Cascade Style Sheets, neboli kaskádové styly. Jedná se většinou o samostatné soubory, ve kterých je definováno, jak budou prvky v HTML vypadat a jak budou zobrazeny.

2.1.3 Javascript

Javascript je skriptovací jazyk na straně klienta. Skript se spouští před nebo po načtení samotné webové stránky, předtím se uloží do paměti prohlížeče. Používá se k asynchronní komunikaci se serverem. Slouží tedy k tomu, aby mohl uživatel interagovat s webovou aplikací bez neustálého načítání samotné aplikace v prohlížeči. Je důležitou součástí tzv. bohatých webových aplikací.

jQuery a AJAX

jQuery je JavaScript knihovna, která usnadňuje práci s JavaScriptem. Její výhodou je množství předpřipravených funkcí, rychlost a odlehčenou oproti jiným knihovnám. (jquery-navod.cz, 2020) Je použitelná v drtivé většině dnes používaných webových prohlížečů. JQuery poskytuje metody pro funkce AJAX a zjednodušuje s nimi práci. AJAX je v překladu „asynchronní JavaScript a XML“. AJAX metody slouží pro vytváření rychlých a dynamicky načítaných obsahů webových stránek či aplikací. AJAX umožňuje

stránce asynchronní získávání dat za pomoci malého objemu posílaných informací. V praxi to tedy znamená, že pokud uživatel různými způsoby interaguje na webové stránce s cílem měnit obsah, tak není třeba pokaždé obnovovat celou webovou stránku. To významně zjednodušuje a zpříjemňuje práci mimo jiné s aplikací, jež se zabývá tato bakalářská práce. V dnešní době se jedná o nepostradatelný prostředek při vývoji webových aplikací.

2.1.4 MySQL

MySQL je jedno z nejrozšířenějších řešení pro relační databáze. Jedná se o tzv. systém řízení báze dat (angl. Database management systém – DBMS). V současnosti vyvíjený společností Oracle. Je k dispozici jako svobodný software pod licencí GPL, je multiplatformní a komunikace probíhá v jazyce SQL. Práce s databází probíhala v klientovi phpMyAdmin, který je dostupný v rámci balíčku XAMPP.

2.1.5 JSON a GeoJSON

JSON je formát, který je široce používán pro ukládání a přenos strukturovaných dat v rámci internetové interakce. Používá se především k přenosu dat mezi serverem potažmo databází a webovou aplikací (w3schools, 2020). Formát GeoJSON logicky vychází z formátu JSON. Jak už z názvu vyplývá, jedná se o „otevřený standardní formát navržený pro reprezentaci jednoduchých prostorových geografických dat společně s jejich atributy. Slouží k (de)kódování rozličných geografických dat. Používá WGS84 a desetinné stupně jako jednotky“ (ESRI, 2020)

```
[{"lat": "34.71678454", "lon": "-6.33593723", "datum": "0000-00-00", "name": ""},
{"lat": "33.91829459", "lon": "-0.44726536", "datum": "0000-00-00", "name": ""},
{"lat": "53.25483997", "lon": "7.8436847", "datum": "2020-01-01", "name": "DE - A"},
{"lat": "53.53287493", "lon": "10.1083784", "datum": "2020-01-03", "name": "DE - B"},
{"lat": "54.66844023", "lon": "25.32464164", "datum": "2020-03-30", "name": "lot3"},
{"lat": "47.24108", "lon": "11.41137699", "datum": "2020-03-31", "name": "druhy"},
{"lat": "47.03182235", "lon": "13.6416016", "datum": "2020-04-01", "name": "treti"},
{"lat": "47.3677276", "lon": "15.17968753", "datum": "2020-04-03", "name": "ctvrty"},
{"lat": "48.08749924", "lon": "13.26806644", "datum": "2020-04-20", "name": "sd"},
{"lat": "54.94405068", "lon": "23.83263359", "datum": "2020-04-21", "name": "lot2"}]
```

Obr. 2.1 Odpověď skriptu pointquery.php ve formátu JSON.

2.1.6 Sublime Text

Sublime Text je textový editor pro vytváření a editaci textových souborů, a to především zdrojových kódů. Je šířen v režimu shareware, což v jeho případě znamená, že program jako takový sice není zdarma, nicméně jeho zkušební bezplatná verze není nijak časově omezena a jde jednoduše stáhnout a nainstalovat z oficiálních stránek poskytovatele softwaru. V čase psaní práce byl program ve verzi 3. Tento program nabízí mnoho různých rozšíření, které vylepšují práci s kódem. Tyto rozšíření jsou ve většině případů zdarma.

2.2 Postup zpracování

Postup práce byl navržen a dodržován po prozkoumání zadaného tématu a rešerši odborné literatury. Tento postup autor konzultoval s vedoucím této práce a dle potřeb upravován tak, aby vyhovoval autorovi ke zdárnému dokončení práce.

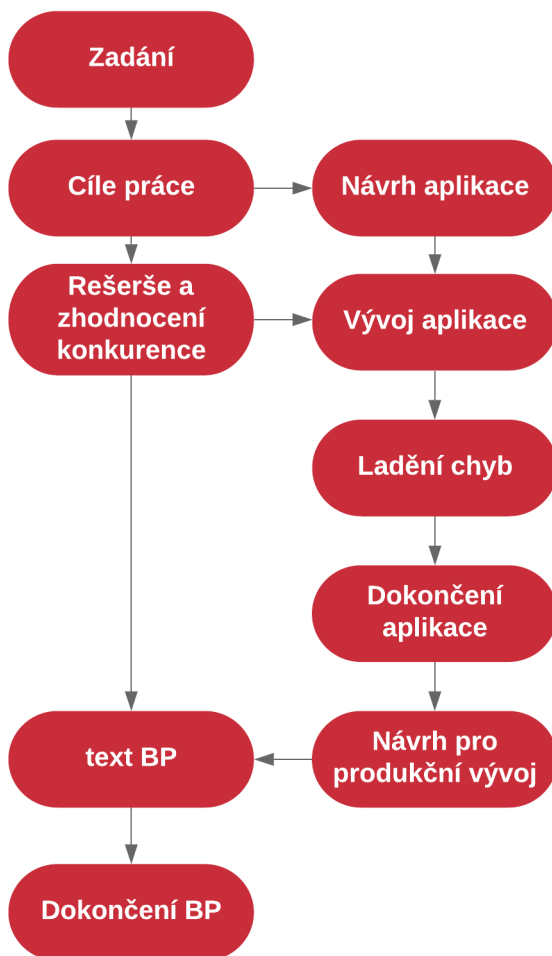
Jako první autor provedl rešerši zadané odborné literatury a rozšířil ji o další zdroje. Vzhledem k počáteční autorově nezkušenosti z hlediska vytváření tak komplexní webové aplikace bylo logicky zcela nutné zorientovat se v možnostech tvorby takových aplikací a osvojit si tyto metody. Poté následoval výběr nejvhodnějších z nich. Pravděpodobně nejdůležitější volba se týkala výběru vhodné JavaScript knihovny pro vytvoření mapové aplikace, která je zcela klíčová pro celou aplikaci, jak už vyplývá z výše uvedeného. Autor se také v co nejvíc možné míře snaží naleznout podobná řešení na trhu, a to jak v komerční sféře, tak i naopak na poli podobně malých projektů z akademického prostředí. Díky této fázi došlo k vymezení odlišností a nastínění, jak bude prototyp vyvíjené aplikace vypadat. A to jak z hlediska vzhledu, tak především z funkčního pohledu. Autor si je vědom, že prototyp s disponibilními prostředky nemá šanci konkurovat či porovnávat se s globálními službami jako například Polarsteps či Swarm. Principem této práce a vyvíjeného prototypu je ale právě pokusit se o tvorbu takové aplikace s dostupnými zdroji z oblasti free and open source v akademickém prostředí. Tomuto se autor věnuje v kapitole již řešené problematiky. Následoval samotný vývoj prototypu aplikace.

Vývoj aplikace probíhal v několika fázích. Po návrhu prvotního uživatelského prostředí autor zvolil pro usnadnění počátečního vývoje aplikace volně dostupnou šablonu s předpřipravenými kaskádovitými styly. Tu si následně upravoval dle svých soukromých potřeb, kde reflektoval různé změny, které vyvstaly při tvorbě. Vývoj aplikace byl rozdělen přibližně do navzájem na sebe navazujících bodů:

- Návrh tabulek v databázích
- Přidávání bodů uživatelem
- Zobrazení uložených bodů z databáze v mapě
- Možnost přidání informací o bodech (název, datum, publicita)
- Postupná implementace vizualizačních funkcí
- Možnost jednoduché editace bodů (tzv. CRUD)
- Vytvoření systému uživatelských účtů a jeho implementace
- Vytvoření stránky s údaji o uživateli
- Dokončovací práce na aplikaci

Po vytvoření prototypu aplikace a jeho schválení vedoucím práce se konalo zhodnocení tohoto vývoje a poslední stěžejní část této práce. Tím je návrh takových řešení, jež by byla dle autorových poznatků ideální pro produkční prostředí k vývoji právě takové webové aplikace pro vizualizaci navštívených míst s prvky sociální sítě. A to z pohledu databází, programovacích metod a mapových produktů. K závěru práce autor diskutuje výsledky

své práce a navrhuje, jak by bylo vhodné vyvinutou aplikaci dále rozvíjet, či čeho by se měl někdo jiný vyvarovat v případě, že by se někdo jiný chtěl věnovat něčemu podobnému. Závěr práce shrnuje výsledky práce.



Obr. 2.2 Schéma postupu práce na BP.

3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Tato kapitola se zabývá podrobnějším průzkumem stávajících řešení v oblasti aplikací, jež uživatelé umožňují různými způsoby ukázat jeho historii geografické polohy získané skrz elektronické zařízení. Kapitola se věnuje nejznámějším produktům z oblasti komerčních řešení, které jsou velmi široce známy a oblíbeny světovou veřejností.

Historie záznamu a vizualizace polohy je zřejmě stejně dlouhá jako lidská historie sama. Vznik prvních map byl svým způsobem kromě prvotního účelu poznání svého okolí i záznamem, kde se daný člověk pohyboval, co viděl nebo kam až daleko se dostal při svých cestách. S příchodem nových technologií se zodpovědnost záznamu přenesla z člověka na elektronická zařízení, v dnešní době typicky GPS lokátory v mobilních telefonech nebo třeba dopravních prostředcích.

Pohyb obyvatelstva se dá pojímat z různých pohledů a je zkoumán obory jako například demografií, filozofií nebo sociologií. Pohyb může být dělen na pohyb přirozený, sociálně ekonomický a mechanický. První jmenovaný je důsledkem přirozeného rozmnožování obyvatelstva a jeho umíráním, naproti tomu pohyb sociálně ekonomický je vyvoláván změnami předních demografických charakteristik obyvatelstva, jako například změna povolání (Toušek a kol., 2008).

Tato práce se zabývá jednou z mnoha možností vizualizace pohybu mechanického, resp. míst, které během takového pohybu daná osoba navštívila. Nezabývá se tedy jakými způsoby může docházet k záznamu takového pohybu, pro kontext práce je nicméně důležité alespoň okrajově zmínit jaké to jsou.

Mezi nejjednodušší možnosti, jak sledovat něčí pohyb je vlastní pozorování nebo pomocí dotazníku. Z pohledu dnešních technologií je nicméně důležité zmínit pojmy jako lokačně založené služby (anglicky location based services – LBS) a tzv. geosociální sítě (location based social network – LBSN). Podle Jarcovjaka (2013) je „hlavní myšlenkou geolokačních služeb je odpovědět uživateli na otázky, kde se nachází, co se nachází kolem něj a kde jsou jeho přátelé. Napomáhají tedy v lokalizaci, hledání, identifikaci a poznávání okolí.“ Virrantaus a kol. (2001) LBS například definoval jako „geolokační informační služby, přístupné přes mobilní zařízení, které jsou schopny za pomoci mobilní sítě využít pozici mobilního zařízení.“

Geosociální sítě úzce souvisí s dnes již standardní součástí životů většiny z nás, a to se sociálními sítěmi. Jarcovják (2013) píše, že principem takové sítě je například možnost vkládat příspěvky, jež se vztahují pouze k určenému místu na Zemi nebo je možné si na mapě nechat zobrazit, kde se právě nacházejí přátelé připojení k dané síti.

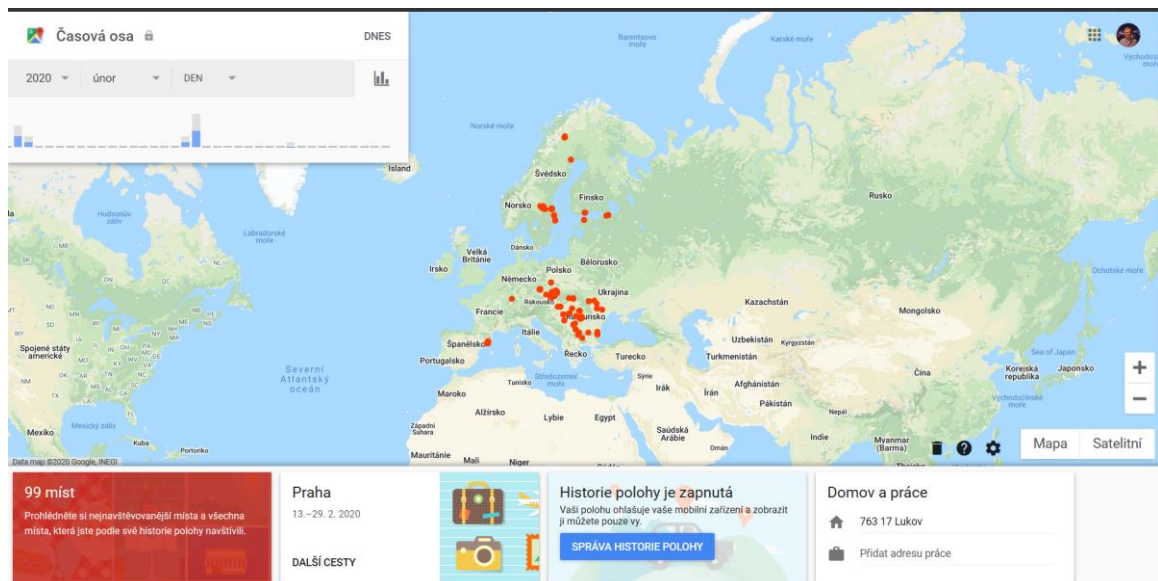
Možnostmi kartografické vizualizace pohybu se například zabývala Vondráková (2007) ve své bakalářské práci. Vizualizacemi zobrazovala pohyb obyvatel města Olomouce ve vybraných lokalitách. Následující podkapitoly se ale zaměřují na výčet popis nejznámějších a nejvíce používaných konkrétních programových řešení na poli aplikací pro vizualizaci navštívených míst.

3.1 Aplikace pro vizualizaci míst nebo pohybu

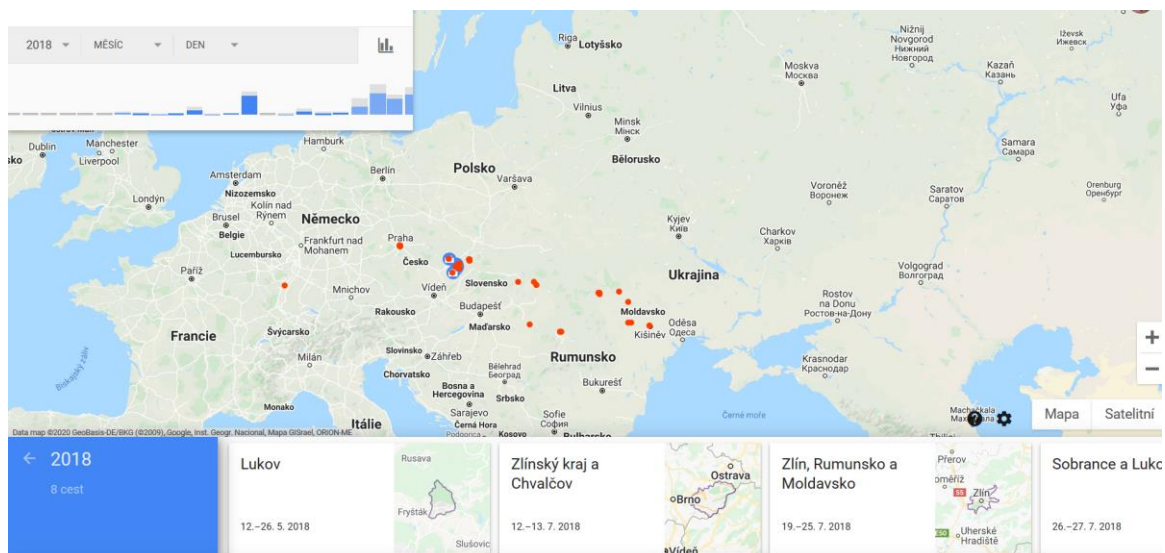
3.1.1 Google Maps Timeline

Asi nejznámějším produktem, který uživateli ukáže nějakým graficky srozumitelných způsobem jeho polohovou historii je Google Maps Timeline. Google maps jsou nejrozšířenější mapovou platformou na světě a nabízí uživatelům rozličné množství funkcionalit. Samotná rozšířenost této aplikace po světě je dána hned několika důvody. Hlavní předností aplikace Google Maps je především její univerzálnost, jednoduchost a přehlednost. Dalším neméně podstatným faktorem, proč je platforma tak rozšířená, je určitě samotná společnost Google, jež stojí za jejím vývojem. Ta samá společnost také automaticky předinstalovává tuto aplikaci na mobilní operační systém Android, který je také velmi rozšířen na mobilních telefonech (Statcounter.com, 2020)

Google Maps Timeline (česky též Google Mapy – časová osa) velmi přehledně zobrazuje uživateli jeho polohovou historii. Tato historie je v drtivé většině případů získávána z mobilních telefonů, pokud jejich uživatele mají povolenou funkci záznamu polohy a její ukládání do aplikace Google Maps.

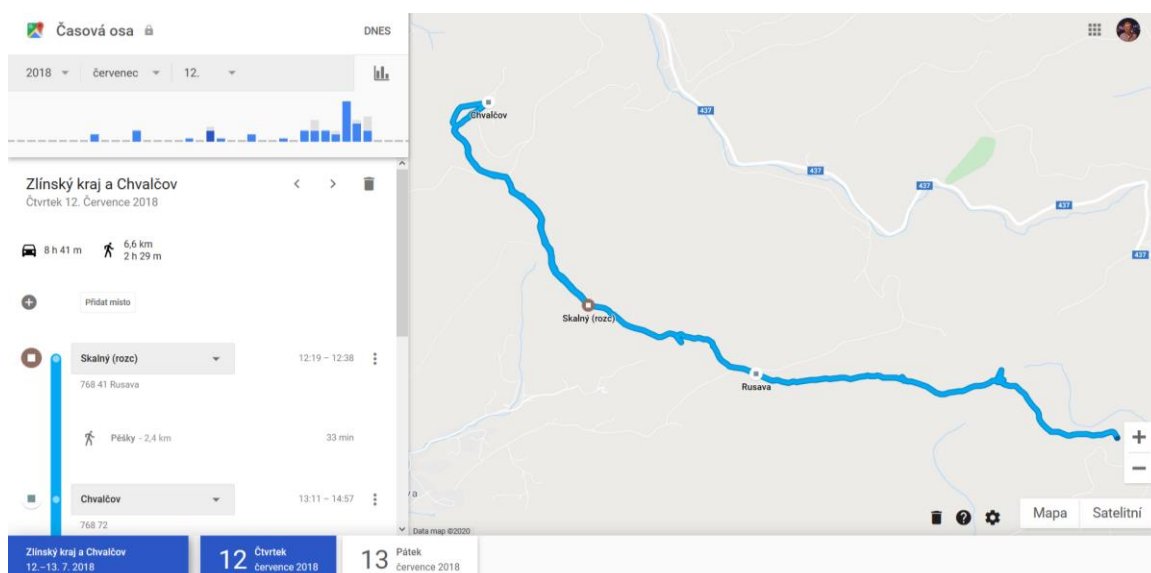


Obr. 3.1 Hlavní stránka Google Maps Timeline.



Obr. 3.2 Zobrazení GM Timeline pro rok 2018.

Podrobnější zobrazení poté, co uživatel definoval nějaké časové rozpětí se ukáže například jako na obrázku 3.2 pro celý rok 2018. Jsou zobrazeny tři hlavní části aplikace. První je mapové okno se samotným mapovým podkladem a navštívenými body. Druhou částí je nabídka určení časového rozpětí, za které se má polohová historie zobrazit. Třetí a poslední je dolní lišta, kde se v časové posloupnosti zobrazují detaily jednotlivých míst či jejich seskupování, slouží k možnému rozkliknutí na nabízené skupiny míst. Po kliknutí dojde k zobrazení jednotlivé cesty v daný časový úsek jako na obrázku 3.2.



Obr. 3.2 Zobrazená GM Timeline pro daný časový úsek.

Aplikace se mírně promění a nalevo se zobrazí podrobnosti u trvání cesty, způsobu dopravy a podobně.

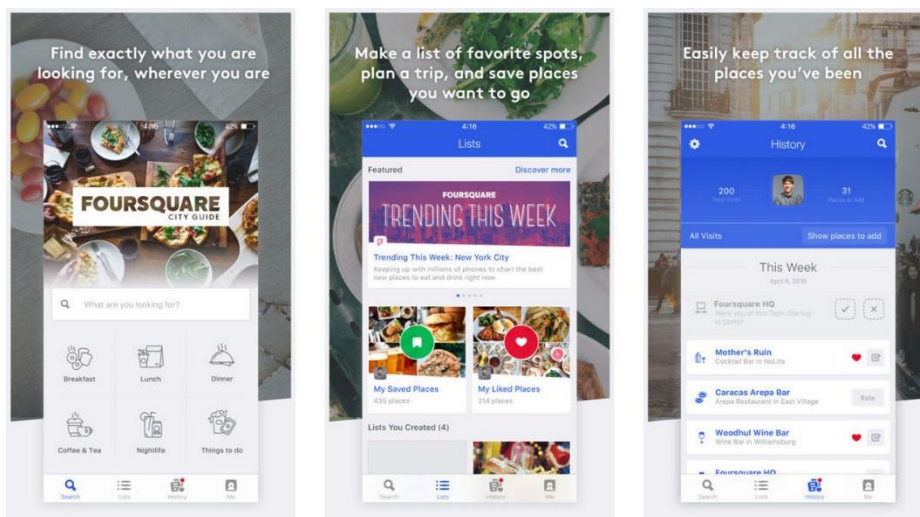
3.1.2 Foursquare Swarm

Foursquare je firma, která v roce 2009 přišla se stejnojmennou aplikací na trh a velmi rychle si ji oblíbili uživatelé po celém světě. Aplikace tehdy pracovala na principu sdílení polohy uživatele mezi jeho přáteli pomocí potvrzení polohy (check-in) na různých místech zájmu pomocí mobilních zařízení. Podmínkou bylo a stále je samozřejmě zjišťování polohy pomocí globálních navigačních a polohových systémů (GNSS).

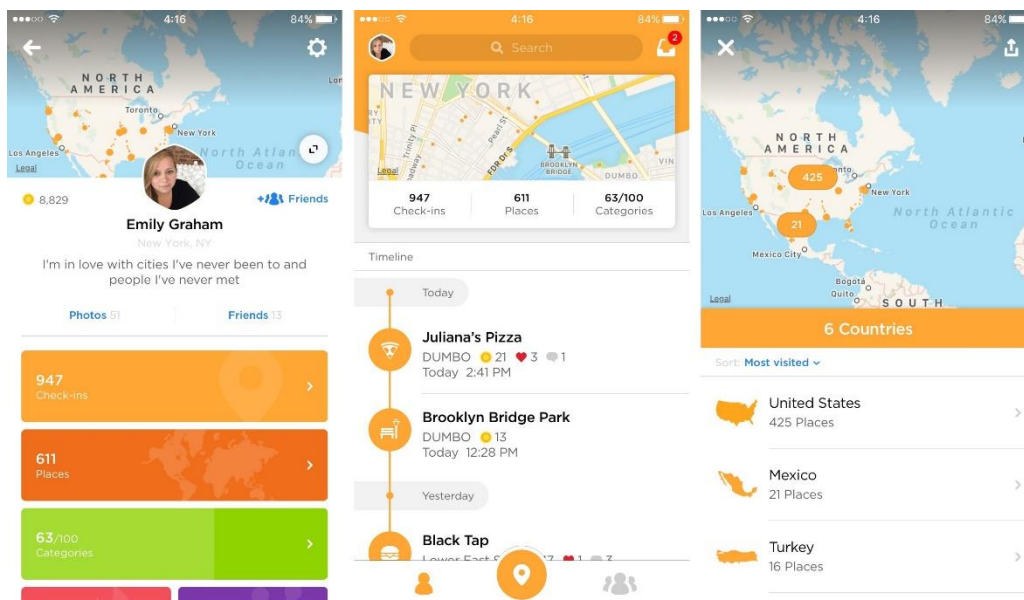
Postupem času díky opravdu značnému celosvětovému rozšíření získala aplikace mnoho funkcí, díky kterým se aplikace Foursquare stal komplikovanějším, než si jeho tvůrci pravděpodobně na začátku přáli. Jak aplikace rostla, přenesl se její gamifikační význam více do oblasti hodnocení navštívených míst, přidávání recenzí, fotografií, popisů o nich anebo naopak jako průvodce do neznámých míst, kde si uživatel například zjistil, v jakém hotelu přespí. To však škodilo jak přehlednosti aplikace, tak i samotným uživatelům, mezi kterými nastal rozkol na ty, kteří aplikaci používají jako geoprostorovou hru a ty, kteří chtějí objevovat nepoznaná místa a hodnotit je. V roce 2014 tedy vznikly dvě aplikace přesně pro každou z těchto skupin lidí: Foursquare City Guide a Foursquare Swarm. Aplikace Swarm je ryzím zástupcem výše zmíněných LBSN aplikací fungujících obdobně.

Jádrem této aplikace jsou výše zmíněné check-ins. Uživatelé stále mohou sdílet navštívená místa s přáteli, a soutěžit s nimi v počtu navštívených míst či v četnosti takových návštěv na jednom místě. Nováková (2015) ve své práci popisuje onen charakter virtuální hry následovně: „Uživatelé ve Swarm sbírají počet navštívených míst, jako tomu bylo v případě Foursquare. V první řadě uživatel po registraci dostane přidělených pouze deset ikon neboli samolepek, které může přiřazovat ke svým „check-ins“. Postupně, jak uživateli přibývá navštívených míst, přibývá i počet a variant těchto ikon, např. když uživatel navštíví několikrát vinotéku, objeví se mu samolepka sklenice vína apod. Druhou formou soutěžení je již zmíněná četnost návštěv konkrétního místa, ale není tomu již tak, jako v případě původní verze Foursquare, kdy mezi sebou soutěžili všichni registrovaní uživatelé. V případě Swarm mezi sebou soutěží jen lidé, jež se mají navzájem v přátelích. Statistiky se objevují podle toho, kdo z přátel navštívil které konkrétní místo kolikrát, anebo také podle charakteru navštíveného místa. Přátelé mají možnost soutěžit mezi sebou v tom, kdo nejvíce navštěvuje vegetariánské restaurace, vzdělávací zařízení nebo večerní kluby.“

Z pohledu vizualizace nabízí obě aplikace několik mapových náhledů o navštívených místech, viz obrázky 3.3 a 3.4:



Obr. 3.3 Prezentační ukázky aplikace Foursquare City Guide (zdroj: iclarified.com).



Obr. 3.4 Ukázky aplikace Foursquare Swarm (zdroj: geoawesomeness.com).

V aplikaci nalezneme časovou osu (Timeline), kterou si uživatel může projíždět, zobrazit která místa navštívil a ukázat toto místo přímo v mapě. Na třetím obrázku je ukázka rozdělení navštívených míst a jejich clustery na mapě.

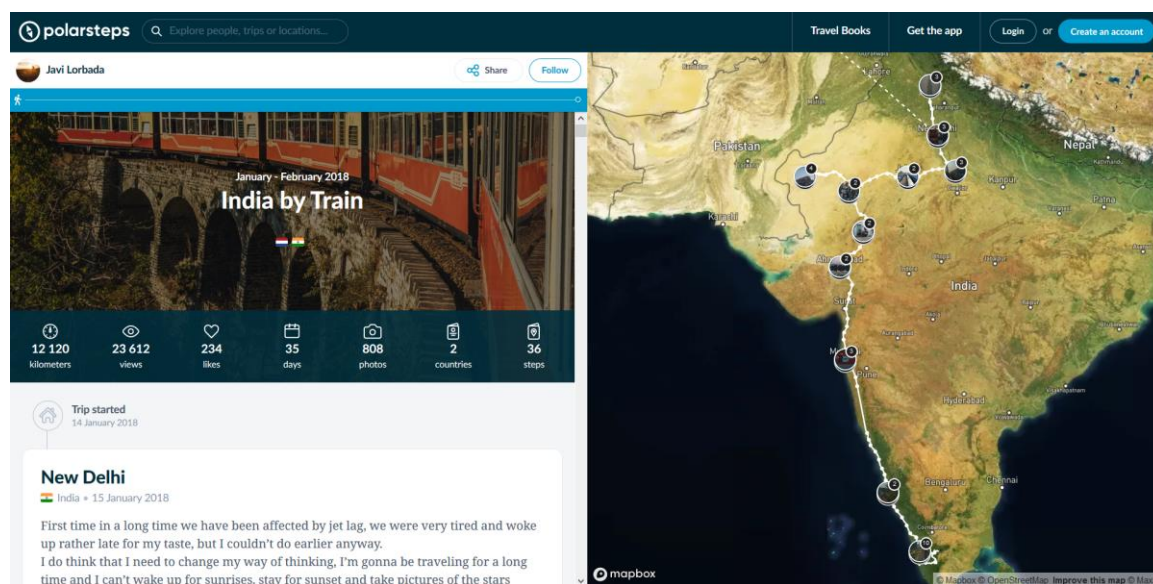
3.1.3 Facebook Local

Vlastní aplikaci v podobném stylu nabízí i společnost Facebook. Facebook Local je mobilní aplikace uvedená na trh v roce 2017, která je výsledkem snahy společnosti usnadnit uživatelům získávání informací o oblasti, kde se právě nacházejí. Ti mohou podle polohy zkoumat recenze podniků ve své blízkosti, zjišťovat kolik přátel dané podniky už navštívilo či přehlednější zobrazení nabídky různorodých kulturních akcí. Nejde ale čistě o geosociální aplikaci, Facebook pouze agreguje určité typy svých dat navázané na geografickou polohu do separátní aplikace pro snazší užití.

3.1.4 Polarsteps

Zřejmě nejvíce podobným řešením na komerčním trhu s vyvíjenou aplikací v této práci je Polarsteps. Jedná se o produkt rozvíjen převážně nizozemskými vývojáři, jehož jádrem je opět záznam polohy skrze mobilní aplikaci. Jejím účelem ale není sledovat každodenní pohyb osoby či zjišťování informací „denní potřeby“ (recenze obchodů, vytíženost dopravy apod.), jak je tomu v předešlých případech, nýbrž se zaměřuje na ukládání zážitků během cestování. Ideální příklad užití se může jevit například cestování stopem po východní Evropě nebo měsíční cyklodovolená podél Dunaje atp. Je zde možnost volby intenzity záznamu polohy, míry detailu, což je třeba výhodné pro šetření kapacity baterie v mobilním zařízení na dlouhé cestě mimo civilizaci.

Cestovatelé pak zaznamenanou trasu mohou buď v průběhu cesty nebo dodatečně editovat a k zajímavým místům přidávat popisy zážitků, doporučení a videa či fotoalba. To vše je pak k nahlédnutí na webu Polarsteps, kde se uživatelé zobrazí prostředí jako na obrázku 3.5. Nalevo se nachází další z variací časové osy, která zde plní spíše funkci cestovatelského blogu-deníku, která je propojena s mapou. Aplikace také funguje bez nutnosti internetového připojení. Je zde také možnost vytvoření fyzické cestovatelské knihy za cenu začínající na 30 amerických dolarech.



Obr.3.5 Ukázka výletu v aplikaci Polarsteps (zdroj: Polarsteps.com).

4 VÝVOJ APLIKACE MAPLAY

Hlavním výstupem této bakalářské práce je návrh a vývoj prototypu aplikace pro vizualizaci navštívených míst. Autorovým záměrem bylo naprogramovat takovou aplikaci, která kromě splnění zadání například vyzkouší možnosti JS mapové knihovny OpenLayers. Pro ztotožnění se s aplikací byl vymyšlen název „Maplay“, se kterým autor v průběhu popisu práce může pracovat. Jak už název napovídá, vznikl ze dvou anglických slov a to „map“ a „play“. Přičemž byla snaha, aby název co nejvíce reflektoval účel aplikace, a to že si uživatel pro vizualizaci aplikaci přehrává.

4.1 Přípravné práce na aplikaci

Mezi prvními návrhy uživatelského prostředí aplikace dominovala autorova podmínka, aby všechny části aplikace byly zobrazeny uživateli v jednom jediném okně. V průběhu času se však ukázalo, že by taková forma uživatelského prostředí byla příliš komplikovaná na to, aby plynule fungoval dynamický princip aplikace. Autor tím má na mysli například to, že vzhledem k jeho počáteční nezkušenosti s vývojem takové komplexní aplikace jak na straně klienta, tak serveru došlo k rozdělení dvou stěžejních částí aplikace na dvě samostatné stránky, a to na stránku mp-points.php s přidáváním uživatelských bodů a mp-visual.php pro jejich vizualizaci. V neposlední řadě také figurovala obecná snaha o celkovou lepší uživatelskou přívětivost aplikace a user experience (UX). Přehlednost byla tímto rozhodnutím zajištěna do uspokojivě možné míry, oproti variantě aplikace, která by obsahovala všechny funkcionality najednou v jednom zobrazeném okně v prohlížeči. To by mohlo uživatele mást.

Pro celkový vzhled celé aplikace bylo použito předpřipravené šablony (<https://templated.co/intensify>), kterou si autor následně upravoval dle svých potřeb. Prototyp aplikace je rozdělen na pět stránek:

- Home
- Add points
- Visualize
- Edit points
- Profile

Úvodní stránka (index.php)



Obr. 4.1 Úvodní stránka aplikace – index.php.

Na obrázku 4.1 je zobrazen snímek úvodní stránky. Ta je taky jediná, kterou uvidí nepřihlášený uživatel. Na této stránce je přichodzí uživatel informován, o čem tato aplikace je a jsou na ní odkazy na přihlášení již zaregistrovaných uživatelů či registraci uživatele nového. Z pohledu vývoje je tato stránka nejjednodušší. Není na ní žádný dynamicky tvořený obsah a uživatel zde interaguje pouze přes jednoduché odkazy, které jsou nalinkovány k posunutí stránky dolů k části o popisu aplikace nebo na stránku s registrací. Kromě toho je na její horní straně lišta, která je jednotná pro celou aplikaci. Na ní je umístěn název aplikace, vpravo tlačítka pro registraci, přihlášení či odhlášení uživatele. Vlevo je umístěno tlačítko rozbalovacího menu – rozcestníku na další části aplikace ve stylu tzv. hamburger menu.

4.1.1 Založení MySQL databáze

Stěžejní částí aplikace je komunikace serveru s databází, kde jsou uložena data o uživatelských účtech a samotná data uživatelů. Aplikace potřebuje k plné funkčnosti celkem 3 tabulky s daty:

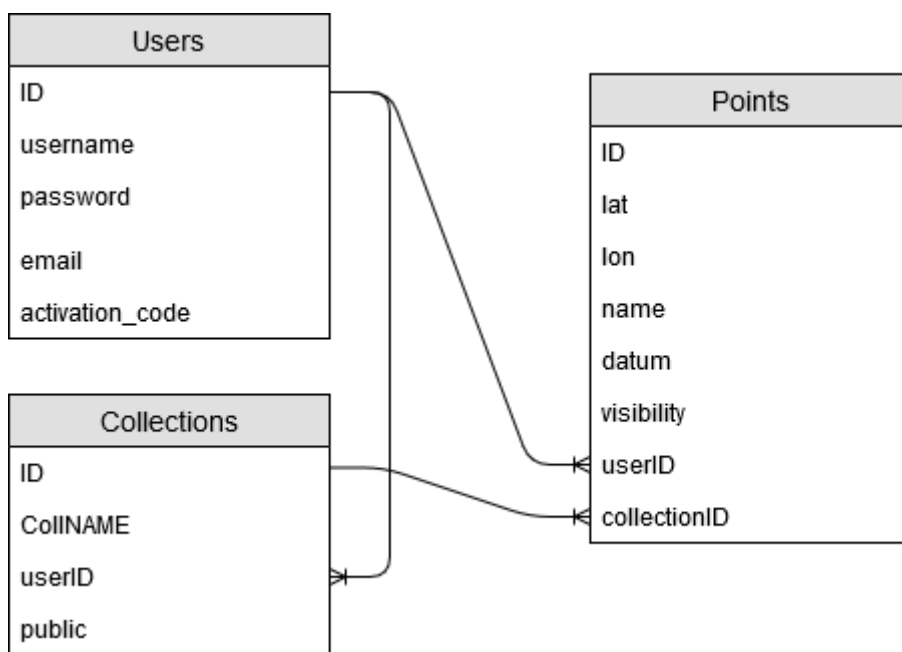
- Accounts
- Collections
- Points

V tabulce accounts jsou uloženy data registrovaných uživatelů. Její struktura je na obrázku 4.2:

| # | Název | Typ | Porovnávání | Vlastnosti | Nulový | Výchozí | Komentáře | Další |
|---|-----------------|--------------|-----------------|------------|--------|---------|-----------|----------------|
| 1 | id 🗝️ | int(11) | | | Ne | Žádná | | AUTO_INCREMENT |
| 2 | username | varchar(50) | utf8_general_ci | | Ne | Žádná | | |
| 3 | password | varchar(255) | utf8_general_ci | | Ne | Žádná | | |
| 4 | email | varchar(100) | utf8_general_ci | | Ne | Žádná | | |
| 5 | activation_code | varchar(50) | utf8_general_ci | | Ano | | | |

Obr. 4.2a Struktura tabulky accounts.

Pole `activation_code` se v prototypu aplikace reálně nevyužívá a je pouze předpřipraveno pro potenciální využití k ověřování uživatele přes e-mailovou adresu. V prototypu se toto pole vyplňuje automaticky jako „activated“, jakmile je uživatel úspěšně zaregistrován. Další pole korespondují se základními požadovanými údaji při registraci. Uživatel si volí své vlastní uživatelské jméno, heslo, které je v databázi šifrované, a zadává svůj e-mail. Identifikační číslo uživatele je primárním klíčem spojeno s cizími klíči `user_id` v tabulkách `collections` a `points`, kde plní identifikační funkci.



Obr. 4.2b Schéma databáze.

V tabulce `collections` jsou uloženy data o přidaných kolekcích. Kolekce plní funkci seskupení dvou a více bodů do jedné, pojmenované entity, pod kterou lze všechny body hromadně zobrazit, třídit je nebo například sdílet s ostatními uživateli. Struktura tabulky je na obrázku 4.3.

| # | Název | Typ | Porovnávání | Vlastnosti | Nulový | Výchozí | Komentáře | Další |
|---|---|--------------|--------------------|------------|--------|---------|-----------|----------------|
| 1 | id  | int(11) | | | Ne | Žádná | | AUTO_INCREMENT |
| 2 | collNAME | varchar(100) | utf8mb4_general_ci | | Ne | Žádná | | |
| 3 | userID | int(11) | | | Ne | Žádná | | |
| 4 | public | int(1) | | | Ne | Žádná | | |

Obr. 4.3 Struktura tabulky collections.

Vazba bodů na kolekci není povinná. Uživatel může přidat body bez toho, aniž by byly zařazeny do kolekce a může je mít volně uložené. Podstatnou funkcí aplikace je sdílení těchto kolekcí s ostatními uživateli. K tomu slouží pole public, které bude mít hodnotu 1 pakliže uživatel chce, aby byla daná kolekce bodů dohledatelná ostatními uživateli. Více v kapitole 4.4 o sdílení mezi uživateli.

Tabulka points je naprosto klíčová. Jak už název vypovídá, jsou v ní uloženy data o bodech – lokacích či navštívených místech. Její struktura je následující:

| # | Název | Typ | Porovnávání | Vlastnosti | Nulový | Výchozí | Další |
|---|---------------------|-------------|-----------------|------------|--------|---------|----------------|
| 1 | id | int(11) | | | Ne | Žádná | AUTO_INCREMENT |
| 2 | lat | double | | | Ne | Žádná | |
| 3 | lon | double | | | Ne | Žádná | |
| 4 | name | varchar(60) | utf8_general_ci | | Ne | Žádná | |
| 5 | datum | date | | | Ne | Žádná | |
| 6 | visibility | int(1) | | | Ne | Žádná | |
| 7 | userID | int(11) | | | Ano | NULL | |
| 8 | collectionID | int(11) | | | Ano | NULL | |

Obr. 4.4 Struktura tabulky points.

Souřadnice bodu jsou uloženy ve dvou polích – lat a lon. Pole name, datum a visibility jsou uživatelsky definovatelná.

4.1.2 Přidání mapového okna do aplikace

V této podkapitole je popsán obecný způsob, jakým je do aplikace zaneseno mapové okno. Knihovna OpenLayers nabízí na svých webových stránkách celkem jednoduchý popis, co dělat, pokud někdo chce vložit mapové okno OpenLayers do vlastního webu.

```

var view = new ol.View({
  center: ol.proj.fromLonLat([17.50, 49.25]),
  zoom: 4
});

var map = new ol.Map({
  target: 'map',
  layers: [
    new ol.layer.Group({
      // A layer must have a title to appear in the layerswitcher
      'title': 'Base maps',
      layers: [
        new ol.layer.Tile({
          // A layer must have a title to appear in the layerswitcher
          title: 'ESRI aerial',
          // Again set this layer as a base layer
          type: 'base',
          visible: false,
          source: new ol.source.XYZ({
            attributions: ['Powered by Esri',
                          'Source: Esri'],
            attributionsCollapsible: false,
            url: 'https://services.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}',
            maxZoom: 25
          })
        }),
        new ol.layer.Tile({
          // A layer must have a title to appear in the layerswitcher
          title: 'OSM',
          // Again set this layer as a base layer
          type: 'base',
          visible: true,
          source: new ol.source.OSM()
        })
      ]
    })
  ],
  view: view
});

```

Obr. 4.5 Programový kód pro vložení OpenLayers mapy na stránku.

Na obrázku 4.5 je ukázka takového vložení mapy na stránce mp-points.php. Od mapového okna na mp-visual.php se liší především v tom, že obsahuje možnost volby ze dvou podkladových map. A to výchozí OpenStreetMap (dále jen OSM) mapy a ortofoto mapy poskytovaných společností Esri.

Pro přidání OpenLayers do stránky je nejprve zapotřebí stáhnout JavaScript balíček ol.js a vložit ho na začátku dokumentu do hlavičky přes `<script src="ol.js"></script>`. Dále je nutné vložit parametr div s mapou `<div id="map"></div>`. Hlavní částí je pak vložení samotného skriptu mapy. V aplikaci Maplay je tento skript vložen přímo do těla souboru. Nejprve je definována proměnná view, která udává základní zobrazení mapy po načtení přes třídu ol.View. Následuje nastavení, kterou je definice proměnné map třídou ol.Map, která je základním stavebním prvkem mapové aplikace. Jak bylo zmíněno výše, na stránce přidávání bodů jsou pomocí tříd ol.layer.Group a ol.layer.Tile definovány vrstvy podkladových map. Je vhodné poznamenat, že OSM podklady jsou již vloženy a OL knihovna s nimi dokáže pracovat pouze s nastavením třídy ol.source.OSM(). To je rozdíl oproti vložení vrstvy ortofoto mapy, kde je využita podtřída XYZ. Její zdroj je popsán přímou URL adresou na server společnosti Esri, která tato data poskytuje.

4.2 Přidávání bodů a kolekcí

V této kapitole je podrobně popsána jedna ze dvou hlavních částí aplikace Maplay, stránka mp-points.php. Poté, co uživatel provede registraci, potažmo přihlášení, má

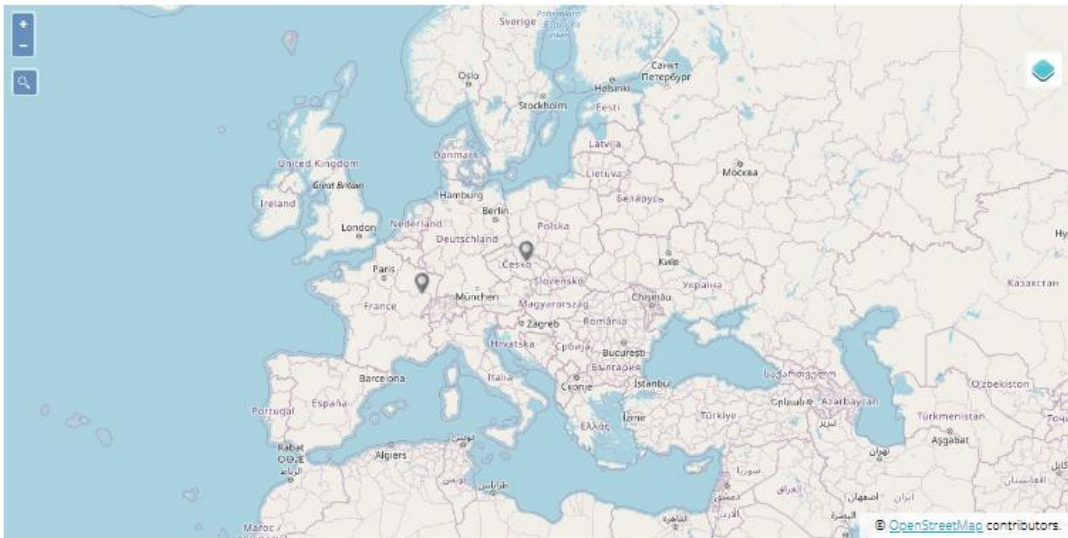
možnost přidat svá data o navštívených místech dvěma jednoduchými způsoby. Dále je popsána možnost editace těchto bodů během přidávání, možnost sbalení všech bodů do kolekce a nastavení viditelnosti – publicity.

Takřka veškerý programový kód je obsažen v jedné jQuery funkci ready, která se spustí, až jsou všechny zdroje potřebné k zobrazení stránky načteny. Mimo ni se nachází pouze header a footer stránky, klientská část možností hromadného nahrávání – upload bodů v souboru GeoJSON a pod mapovým oknem tabulka, množící se podle počtu přidanych bodů. Funkce se spustí, jakmile jsou načteny všechna data.

Select a point

or upload a geoJSON file (only with point features)

Procházet... Soubor nevybrán. UPLOAD



Do you want these points to be a collection? Yes No

body

Public

| Latitude | Longitude | Place name | Date of visit | Public | |
|-------------|-------------|------------|---------------|-------------------------------------|---|
| 49.82983571 | 16.44581250 | bod A | 19. 05. 2020 | <input checked="" type="checkbox"/> | DELETE |
| 47.80467180 | 6.09179561 | bod B | 27. 04. 2020 | <input type="checkbox"/> | DELETE |

SUBMIT
CLEAR

Obr. 4.6 Náhled stránky mp-points.php.

Hromadné nahrávání bodů skrz GeoJSON

Do mapy lze v aplikaci hromadně vložit body ve formátu GeoJSON. Pro tuto funkcionalitu se využívá funkce FileReader a metoda document.getElementById().

Uživatel zvolí soubor v adresáři. Ze souboru se načtou data o bodech do proměnné ve formátu JSON string, což vyvolá načtení metody `JSON.parse`. Ta pak výsledek uloží do proměnné `result` ve struktuře, se kterou se dá následně pracovat. Následuje `for each` funkce, která pro každý bod v proměnné vytvoří marker v mapě přes funkci `addMarker`.

Přepínání podkladových vrstev v mapě a vyhledávání míst (geocoding)

Pro přepínání vrstev bylo použito externího rozšíření pro OpenLayers ze stránky <https://github.com/walkermatt/ol-layerswitcher>. Jedná se o externí JS rozšíření, které si může uživatel nastavit dle svých potřeb. Pro případ této práce je použita pouze základní funkcionality díky třídě `ol.control.Layerswitcher` pro přepínání dvou vrstev v rámci třídy `ol.layer.Group`. Přepínač je v mapovém okně v pravém horním rohu a otevře se automaticky po najetí myši. Umístění je zajištěno pomocí metody `map.addControl(layerswitcher)`.

Velmi užitečnou drobností v aplikaci je možnost vyhledávání lokací a jejich zobrazení v mapě. To využívá rozšíření `ol-geocoder`, dostupné z <https://github.com/jonataswalker/ol-geocoder>. Stejně tak jako předchozí externí rozšíření, musí být v hlavičce odkázáno na zdroj na serveru, kde se tato extenze nachází. Programový kód použitý v hlavním skriptu je na obrázku 4.7.

```
var geocoder = new Geocoder('nominatim', {
  provider: 'osm',
  lang: 'en',
  placeholder: 'Search for ...',
  limit: 4,
  debug: false,
  autoComplete: true,
  keepOpen: true
});
map.addControl(geocoder);

//listen when an address is chosen
geocoder.on('addresschosen', function (evt) {
  console.info(evt);
  window.setTimeout(function () {
    popup.show(evt.coordinate, evt.address.formatted);
  }, 3000);
});
```

Obr. 4.7 Kód funkce `ol-geocoder`.

Proměnná *markers*

Malá, byť podstatná část kódu, je definování proměnné `markers`. Ta, jak název napovídá, nastavuje, jak budou vypadat body v mapě po grafické stránce. Knihovna OpenLayers ve výchozím stavu ukazuje body pouze velmi nepatrným malým kruhem, je tudíž žádoucí, aby byly body v mapě vidět zřetelně během práce s aplikací. Stylizaci markeru se docílí tak, že se využije obrazového podkladu a určitých rozměrů tohoto obrázku, který bude představovat marker.

4.2.1 Přidávání bodů kliknutím do mapy

Následující kapitola popisuje, jakým způsobem probíhá přidávání bodů uživatelem přes klikání myši do mapového okna. Autor předpokládá, že takový způsob by byl nejvíce využívaný uživateli, a tudíž je mu věnována větší pozornost.

Nejprve jsou definovány tři prázdné pole. Pole `markersArray` drží všechny instance třídy `marker`. Pole `place_names` sdružuje informace o názvech míst a pole `formLines` drží řádky formuláře, jehož fungování je popsáno v podkapitole o funkci `updateForm`. Vše je provedeno metodou `map.on()` s parametrem `click`. Pokud uživatel klikne kamkoliv do mapy, funkce to zachytí a přidá marker do mapy funkcí `addMarker` (viz dále). Pokud však klikne na již existující marker, funkce to rozpozná a vrátí k zobrazení popup (informační okénko) s názvem místa (pokud byl vyplněn) a s polohou náležející k markeru. Tohle slouží k rozpoznání jednotlivých bodů v mapě, aby se uživatel mohl ujistit, která místa již má přidat.

Funkce `addMarker`

Funkce má dva parametry, a to `evt` a `coord`. Na začátku funkce je přidána proměnná `mouseCoords`, ve které jsou vždy aktuální souřadnice po kliknutí myši v mapě. Pokud uživatel nahrává body přes hromadný upload, je `evt` parametr `null`, tudíž se souřadnice musí nahrát jiným způsobem přes metodu `ol.proj.transform()`. Následuje proměnná, která každému markeru přidá atribut `name` (název), který drží index v poli `place_names` a poté samotné grafické nastavení zobrazení markeru přes třídu `ol.style.Style()`.

Podstatnou částí funkce je vůbec rozlišení zeměpisné šířky a délky. Toho je docíleno přes proměnné `templateLon` a `templateLat`, které slouží jako parametry výsledných proměnných `lat` a `lon`. S těmi se pak dále pracuje například při vyplňování polí v tabulce pod mapovým oknem.

```
var templateLon = '{x}';
var templateLat = '{y}';
var lon = ol.coordinate.format(mouseCoords, templateLon, 8);
var lat = ol.coordinate.format(mouseCoords, templateLat, 8);
```

Obr. 4.8 Nastavení struktury souřadnic.

Souřadnice jsou ve formátu stupňů (`xx.xxxx`) s přesností na osm desetinných míst. Poté se vytvoří objekt `formLine` která drží informace o řádku a slouží k propojení s tabulkou pod mapou. Do tohoto objektu je pak pomocí metody `unshift` na první pozici (proto `unshift`) přidán nový řádek do proměnné `formLines` a dojde k jejímu obnovení díky funkci `updateForm`

Funkce `updateForm`

Funkce `updateForm` slouží k upravování formuláře/tabulky pro daný počet bodů v mapě. Tento formulář slouží k zobrazení všech bodů, které uživatel přidal, a k editaci jejich názvů, datumu návštěvy a viditelnosti pro ostatní uživatele. Je také možné tlačítkem `delete` jakýkoliv bod smazat jak z formuláře, tak z mapy. Ve funkci je vytvořena pomocná proměnná `form`, kde se formuje obsah formuláře.

Formulář se dynamicky rozšiřuje, jakmile je přidán další bod v mapě. Toho je docíleno jQuery funkcí, jež funguje jako `for each` smyčka. Ta použije každý záznam v poli `lines` (to je definované na začátku funkce) a umožní manipulovat s každým řádkem ve formuláři. Dále je vytvořen string (řetězec) pro každý řádek formuláře a přidán do pomocné

proměnné. Každému bodu je také přiděleno identifikační uid pro spárování s popup oknem (informační bublinou) či obecně pro jakékoli napojení na další akce s datovou strukturou např. spárování řádku tabulky s daty o bodech. Ve smyčce je definována struktura řádku formuláře, jednotlivá pole a zdroje obsahu. Kromě zeměpisné šířky, délky, názvu a datumu je zde také if else podmínka u pole public, pokud je zaškrtnuto pole viditelnosti bodu. Jako poslední je zde umístěno tlačítko ke smazání jak řádku, tak především náležícího bodu. Následuje funkce, která vloží vytvořený formulář do těla tabulky/formuláře v html. Ke konci je vytvořena funkce, která pro každé pole datum omezuje přidání dat budoucích.

Úpravy formLines pole

V této části jsou funkce navázané na input pole uživatele, které naslouchají každou interakci/změnu s inputem, aby upravily data v proměnné. To zajistí, že v případě, že uživatel klikne na upload, tak budou data uložena v proměnných vždy synchronizována s tím, co uživatel vidí, píše nebo zadává.

Zvýraznění markeru

Při přejíždění kurzorem myši ve formuláři s body pod mapovým oknem, se automaticky mění barva markeru – bodu pro lepší orientaci. Toho je docíleno funkcí na obrázku 4.9.

```
// Při najetí na řádek ve formuláři se zvýrazní daný bod v mapě
$("html").on("mouseenter", "tbody tr", function(e) {
  // Sežene id řádku stejně jako v předchozí funkci
  var id = $(this).data("id");
  var feature = markersArray[id];
  var markerStyle = new ol.style.Style({
    image: new ol.style.Icon({
      anchor: [0.5, 1],
      src: 'images/marker2.png'
    })
  })
  feature.setStyle(markerStyle)
})
```

Obr. 4.9 Funkce pro zvýraznění markeru.

Zjednodušeně řečeno je tohoto efektu docíleno tak, že se při najetí myši rozpozná id řádku a podle spárovaného id bodu ze změni jeho marker na marker2.png. Což je marker stejného stylu, jen v červeno-žluté variantě pro lepší orientaci. Po této funkci následuje ta stejná, jen s parametrem „mouseleave“. Ta hlídá opačný proces, který při opuštění kurzoru navrátí původní styl markeru.

Odeslání bodů do databáze

Poté, co uživatel nahraje různými způsoby svá navštívená místa a upraví jejich metadata, jako názvy či datum návštěvy, má možnost tato data odeslat do databáze v MySQL. Proces odesílání má dvě části, první část je v samotné JavaScript části stránky a druhá v ukládacím souboru save.php.

První část se stará o to, jakým způsobem se data pošlou souboru save.php. To se děje pomocí metody AJAX a POST požadavku. Kvůli kumulativnímu způsobu množení řádku ve formuláři, je tohle lepší způsob než čistě přes metodu form submit. Použití metody AJAX umožňuje především to, že se stránka nemusí při odeslání formuláře celá znovu obnovit. Pokud totiž uživatel někde udělal chybu, může ji snadno opravit a data odeslat znovu. Nebo se také provede kontrola validity dat a v případě chyby o tom skript informuje uživatele a ten svá data může opravit. Jinak by se mu celý formulář vymazal a veškerý postup by musel zopakovat, což není pohodlné. Data jsou převedena metodou JSON.stringify() a odeslána metodou AJAX ukládacímu skriptu, předtím je však nutná tato data formovat do nějaké struktury. Uživatel může ale nemusí své přidané body seskupit do kolekce a také viditelnost pro ostatní uživatele, což je potřeba rozlišit právě v kódu na obrázku 4.10.

```
var collection_data = []
var isCollection = $("#collection-radio:checked").val();
if (isCollection == "Yes") {
    var collection_name = $("#collection-name").val()
    var collection_visibility = $("#collection-visibility").prop('checked');
    formLines.map(function(val, i) {
        val["public"] = collection_visibility;
    })
    collection_data.unshift({"name": collection_name, "public": collection_visibility});
}
```

Obr. 4.10

Poté je vytvořena jednoduchá proměnná data, která se pak ve finální AJAX funkci zakóduje do formátu JSON: `var data = {"collection": collection_data, "points": formLines}`. Odesílání metodou POST má standartní strukturu: je definován cílový soubor (save.php), data o proměnných, které se mají odeslat (`data: { 'data': JSON.stringify(data)}`) a v neposlední řadě jakým způsobem se má nakládat s odpovědí při úspěchu či neúspěchu procesu odesílání. PHP skript pošle odpověď, která je také ve formátu JSON a má dva klíče: status a message. Status je buď true nebo false, podle toho, jestli se data uložila správně či nikoli. V message je informace odpovídající úspěšnosti skriptu save.php.

Save.php

Save.php je z podstaty věci v této části aplikace nejdůležitější. Na začátku tohoto samostatného skriptu se začne relace (session), přes které se přenáší data nejen o uživateli v téměř celé aplikaci Maplay. Více o sessions v podkapitole 4.4.2. Následuje podmínka `include("db-config.php")` a přihlašovací proces, které zajišťují správné přihlášení k databázi právě zmíněným souborem. Pokud se tohle celé nepodaří, skript vrátí chybu. Jelikož data z formuláře do tohoto skriptu přicházejí ve formátu JSON, je nutné je dekodovat. Dále, jelikož jsou data uložena v objektu, tak musíme zápis změnit na pole. Poté je zde if-else podmínka sloužící k rozpoznání, jestli uživatel chce body mít seskupené v kolekci nebo ne. Pokud ano, jsou data poslány do tabulky collections a vytvořeno id kolekce, které je pak přiřazeno bodům v tabulce points v poli collection_ID. Pokud uživatel nechce body ukládat pod kolekci, přistupuje se přímo k ukládání

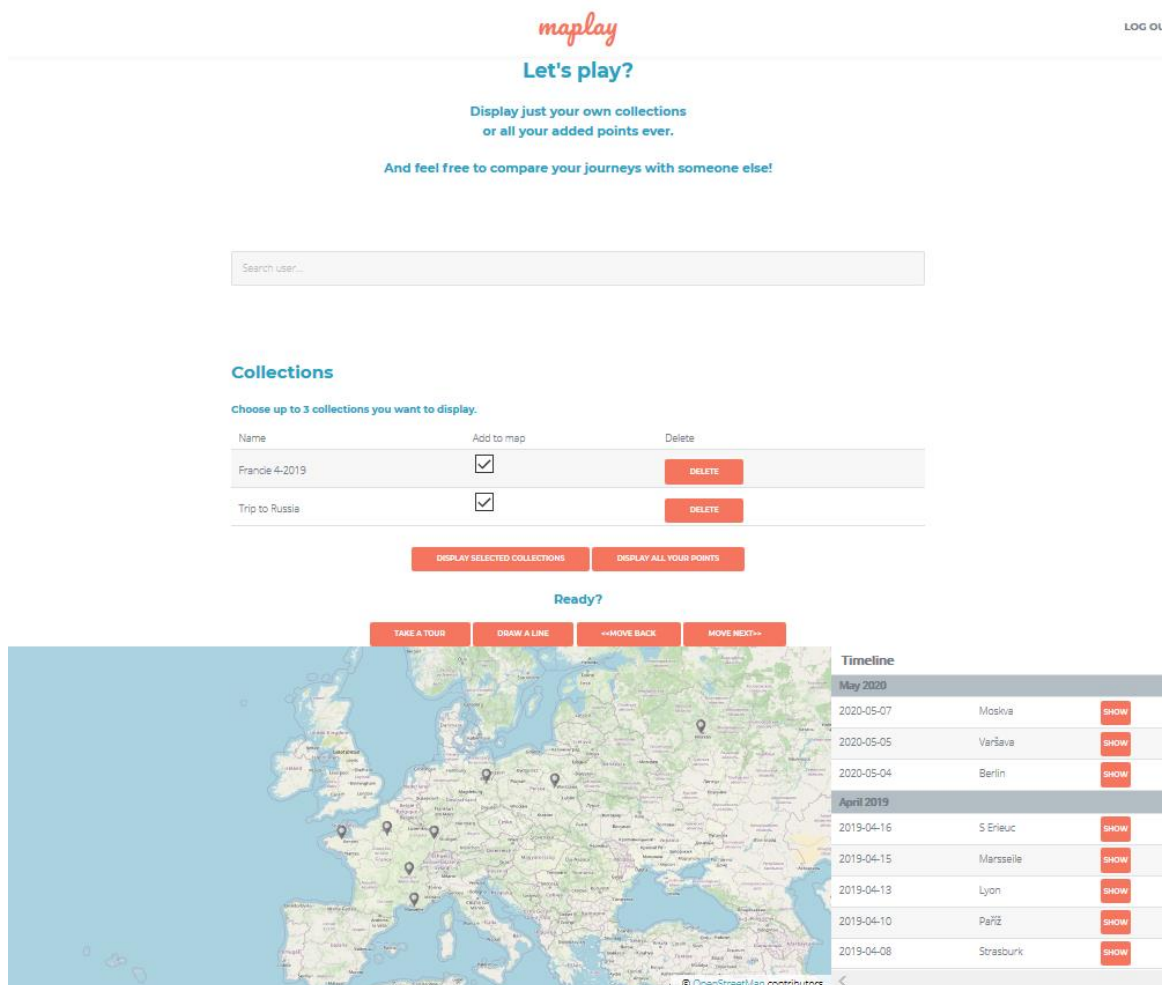
samotných bodů. Pro každý řádek formuláře proběhne for each smyčka a podobně jako u kolekcí je nutno přeměnit data z objektu do pole. Skript pokračuje definicí, která proměnná náleží k jakým datům a ukládá je klasickým SQL příkazem INSERT INTO points (...) VALUES (...). Je zde ještě try-catch kontrola, pokud by došlo v průběhu skriptu k chybě. Pokud by k ní došlo, smyčka se ukončí a vypíše chybovou hlášku. Pokud nikoli, skript odešle odpověď zpátky ke klientovi, že vše proběhlo v pořádku a body byly úspěšně uloženy.

Konec kódu mp-points.php

Po ukončení vloženého JS skriptu je v html vytvořený formulář na straně frontendu, tlačítka na odeslání či vymazání polí a patička stránky.

4.3 Vizualizace navštívených míst v mapě

Tato kapitola popisuje to nejdůležitější z této bakalářské práce. Tím je vizualizace bodů v mapě v prototypu aplikace Maplay. Nejde jen o jednoduché statické zobrazení bodů nad mapou, ale také o dynamické zobrazení pomocí efektu přelétávání (funkce Tour) a vykreslování spojovací čáry mezi navštívenými body. To vše podle toho, jaký uživatel je zrovna přihlášen a jaké kolekce si chce zobrazit v mapě. Kromě mapové části je v této kapitole popsán způsob tvorby časové osy: přehledové tabulky v pravé části stránky, ve které uživatel vidí své body přidané v mapě chronologicky podle definovaného data návštěvy. Stránka má na serveru název mp-visual.php a kromě výše zmíněného umožňuje propojit záznamy dvou až třech jiných uživatelů, kteří poskytují svá data veřejně. Sdílení záznamu se autor více věnuje v kapitole 4.4 o uživatelských účtech. Celkový vzhled stránky s několika zobrazenými body bez vizualizace je na obrázku 4.11:



Obr. 4.11 Stránka mp-visual.php.

4.3.1 Vyhledávání uživatelů a zobrazení jejich kolekcí

Po načtení stránky se uživateli zobrazí krátká textová informace o tom, co na ní může udělat. Kód po prvcích jako header, vložení JS knihoven atp. obsahuje menší ze dvou skriptů, který obstarává dynamické vyhledávání uživatelů. Toho je docíleno opět metodou AJAX. Architektura tohoto skriptu byla částečně převzata ze stránky <https://www.tutorialrepublic.com/php-tutorial/php-mysql-ajax-live-search.php>. Bylo ji však nutné aplikovat na dynamicky měnící se data, systém uživatelských účtů a další specifika aplikace Maplay.

Nejdříve je zde naprogramováno pole na vyhledávání ostatních uživatelů v databázi. Prototyp umožňuje zobrazení všech – pokud uživatel zná jejich uživatelská jména. Není tudíž nijak ošetřena možnost „skrytí“ vlastního profilu před jinými uživateli, viz diskuze. Skript používá metodu GET ke spuštění souboru search-user.php. Vyhledávání funguje asynchronně, při psaní username se nabídka výsledků automaticky mění dle vyplňovaného textu. Následně je nabídka výsledku udělána tak, že se zde zobrazí tlačítko

show user collections, tedy zobrazit kolekce uživatele. Po kliknutí na něj se spustí další AJAX funkce, která opět metodou GET vyvolá skript getUserCollectionById.php a podle identifikačního čísla vyhledaného uživatele vrátí veřejně zobrazitelné kolekce do tabulky pod tímto vyhledáváním. V této tabulce jsou kromě názvů kolekcí i zaškrťovací pole, které slouží k výběru kolekcí celkově do mapy uživatelem. Počet vybraných kolekcí je omezen na tři, nezávisle na tom, jestli se jedná o kolekce vlastní či jiných uživatelů. Je možné zaškrtnout jednu kolekci například uživatele A, vyhledat uživatele B, vybrat jeho kolekci a doplnit ji výběrem kolekce své vlastní. Poté stačí výběr potvrdit tlačítkem display selected collections a zobrazit body v mapě obsažené ve všech těchto kolekcích.

```
//VYHLEDAVANI UZIVATELU
$(document).ready(function(){
  $('#user-search input[type="text"]').on("keyup input", function(){
    /* Get input value on change */
    var inputVal = $(this).val();
    var resultDropdown = $(this).siblings(".result");
    if(inputVal.length){
      $.get("search-user.php", {term: inputVal}).done(function(data){
        // Display the returned data in browser
        console.log(data)
        resultDropdown.html(data);
      });
    } else{
      resultDropdown.empty();
    }
  });
});
```

Obr. 4.12 ukázka 1/3 skriptu k vyhledávání uživatelů.

Pod prvky týkající se vyhledávání je tabulka zobrazující přihlášenému uživateli všechny jeho kolekce. A to logicky i ty, které neurčil jako viditelné pro ostatní. Pro tuto část je určena html třída collection-list s vloženým PHP skriptem a jednoduchým SQL dotazem na kolekce z id uživatele.

| Name | Add to map | Delete |
|---------------|--------------------------|--------|
| rakousko | <input type="checkbox"/> | DELETE |
| sever nemecka | <input type="checkbox"/> | DELETE |
| dd | <input type="checkbox"/> | DELETE |

DISPLAY SELECTED COLLECTIONS

DISPLAY ALL YOUR POINTS

Obr. 4.13 Volba kolekcí.

4.3.2 Základní zobrazení bodů

Druhý, podstatně rozsáhlejší skript, je základním kamenem stránky mp-visual.php. Veškeré funkcionality jsou obsaženy v jedné jQuery funkci, z podobného důvodu, jako je tomu na stránce mp-points.php, která řeší přidávání bodů do databáze uživatelem. Na začátku této jQuery funkce je definováno několik globálních polí, které jsou v různých částech skriptu používány. První základní funkcionalitou, kterou může uživatel využít při prvotním otevření stránky mp-visual.php je zobrazit si všechny jim přidávané body, které má uloženy v databázi. Na stránce se po kliknutí na Display all your points odešle požadavek na stažení a zobrazení všech bodů do mapy z id právě přihlášeného uživatele. To se děje pomocí AJAX funkce metodou GET, která spustí skript pointquery2.php. Jeho struktura je na obrázku 4.14.

```
<?php
session_start();

include("db-config.php");

$conn = new mysqli($servername, $username, $password, $dbname);

$query = "
SELECT `lat`, `lon`, `datum`, `name` FROM `points`
WHERE `lat` <> 0
AND `userID` = '" . $_SESSION['id'] . "'
ORDER BY DATE(`datum`) ASC
";
$query = mysqli_query($conn,$query);

if ( ! $query ) {
    echo mysql_error();
    die;
}

$data = array();

while($row = mysqli_fetch_assoc($query)){
    $data[] = $row;
}

echo json_encode($data);

mysqli_close($conn);
?>
```

Obr. 4.14 Kód skriptu pointquery2.php.

4.3.3 Funkce rerenderMap

Tato funkce, jak už název napovídá, byla vytvořena k tomu, aby zjednodušila zobrazování bodů v mapě. Pokud je funkce spuštěna s parametrem points a parcoll_A, načte pole s názvem polePrvku, které primárně slouží k přenosu dat o poloze bodů, a další pomocné pole. Jádrem funkce je for each smyčka v poli points, která pokaždé vybere potřebná data, jako zeměpisnou šířku a délku a vloží je do pole polePrvku. Následuje vytvoření markerů pro každý prvek v tomto poli. Kromě souřadnic funkce pracuje i s názvy míst, které posílá do pole place_names a na konci vyvolává funkci updateForm, která slouží v plnění časové osy. Funkce updateForm na stránce mp-visual.php je popsána v kapitole 4.3.5.

4.3.4 Bounding box

Malou, byť důležitou funkcí aplikace při vizualizaci je nepochybně bounding box, který zajišťuje, aby se mapa zobrazovala podle toho, kde se body nachází. Není žádoucí, aby měla mapa stále výchozí zobrazení, jelikož uživatel musí vidět všechny své body bez

ohledu na to, kde se nachází a mapa si s tím musí umět poradit. K tomu je využita třída `ol.extent.boundingExtent` (polePrvku) a její související nastavení. Ve výchozím stavu je prostorový rozsah mapy přesný, což má za následek to, že okrajové body nejsou vidět. Tento problém je vyřešen pomocí oddálení zoomu mapy o jeden stupeň.

```
function boundingBox() {  
  // Bounding box spatial extent  
  var ext = ol.extent.boundingExtent(polePrvku);  
  ext = ol.proj.transformExtent(ext, ol.proj.get('EPSG:4326'), ol.proj.get('EPSG:3857'));  
  map.getView().fit(ext, map.getSize());  
  map.getView().setZoom(map.getView().getZoom() - 1);  
}
```

Obr. 4.15 Funkce boundingBox.

4.3.5 Funkce updateForm – časová osa

Funkce `updateForm` tvoří druhou podstatnou část vizualizace, a to časovou osu. Tato časová osa není vytvořena klasicky v horizontálním směru, jak je obecným zvykem. Autor se pokoušel využít více rozšíření třetích stran, avšak hlavní komplikací bylo vždy využívání dynamických dat v aplikaci. Přistoupilo se tedy k variantě, vytvořit obdobu časové osy vlastním způsobem. Časovou osu na pozadí tvoří především funkce `updateForm`. Ta upravuje počty řádků osy (pozn.: autor používá označení `osa`, ve své podstatě zde žádná osa není, správně by se měla tato část stránky nazývat například okno s údaji o zobrazených bodech v mapě dle času – viz obrázek 4.16, pro jednoduchost se však autor drží označení časová osa nebo Timeline) podle daného počtu markerů. Jediným parametrem je proměnná `lines`. Je vytvořeno několik pomocných proměnných pro formování obsahu Timeline. Jádrem je podobně jako ve funkci `renderMap` for each smyčka, která z proměnné `date` každého bodu vytváří jednotlivé řádky osy. Jedná se o podobný způsob, jakým je vytváření tabulky formuláře na stránce `mp-points.php`.

| Timeline | | |
|------------|-----------|------|
| May 2020 | | |
| 2020-05-07 | Moskva | SHOW |
| 2020-05-05 | Varšava | SHOW |
| 2020-05-04 | Berlin | SHOW |
| April 2019 | | |
| 2019-04-16 | S Eriec | SHOW |
| 2019-04-15 | Marsseile | SHOW |
| 2019-04-13 | Lyon | SHOW |

Obr. 4.16 Ukázka časové osy.

Řádek kromě zobrazení datumu a názvu navštíveného místa nabízí i tlačítko show. Po kliknutí na něj se zobrazení mapy přiblíží na daný bod. Proces zjistí id řádku, spojí si to s daným bodem a pomocí funkce `flyTo(ol.proj.fromLonLat(polePrvku[id]), function() {})` uživatel zjistí, kde se v mapě daný bod nachází.

Pro rychlejší orientaci než přes tlačítko show, může uživatel podobně jako ve formuláři na stránce `mp-points.php`, přejíždět kurzorem po této časové ose. Podle pohybu nad řádky se mu zvýrazňují příslušné body v mapě změnou barvy markeru.

4.3.6 Funkce `drawLine`

Možnost vykreslení spojovací čáry mezi jednotlivými body je další možností vizualizace navštívených míst v aplikaci Maplay. Tuto funkcionalitu obstarává funkce `drawLine`. Funkce dostane pole těch bodů, ze kterých má vytvořit čáru. Pomocí smyčky a funkce `setTimeout`, která umožňuje čáru vykreslovat postupně se do mapy postupně přidá čára spojující dané body obsažené v parametru `lines`. S využitím OL vrstvy `LineString` vytvoří OL feature (prvek), která se přidá do mapy jako vrstva.

4.3.7 Smazání a výběr kolekcí

Jak už bylo zmíněno v kapitole 4.3.1 uživatel má možnost zobrazit maximálně tři kolekce. Kromě tohoto omezení má možnost svou vlastní kolekci s příslušnými body smazat z databáze. Tato podkapitola se věnuje popisu zobrazení vlastních kolekcí a jejich smazání. Toho je docíleno dvěma jQuery funkcemi. V prvním z nich (pořadí v kódu) je metoda AJAX, která pomocí metody GET spouští skript `deleteCollection.php`. V tomto

skriptu se po připojení k databázi vybere příslušná kolekce, která se má smazat pomocí příkazu SELECT. Po získání id do proměnné se přistupuje k příkazu DELETE z příslušné tabulky a skript skončí.

V kódu následuje ošetření už zmíněného limitu pro výběr maximálně tří kolekcí, bez ohledu na to, jestli se jedná o kolekce vlastní či cizí. Opět se jedná o jQuery funkci, tentokrát s podmínkou if-else. Pokud délka pole selectedCollections dosáhne čísla 3, skript klientovi vrátí upozornění, že jich není možné přidávat další kolekce a zabrání další exekuci funkce.

Další část skriptu obstarává samotné zobrazení vybraných kolekcí pomocí tlačítka display selected. Opět zde hraje důležitou roli AJAX, který spustí skript getCollectionByID.php. Odpověď z tohoto skriptu jsou kolekce, které jsou zpracovány funkcí rerederMap.

4.3.8 Funkce getNewCollectionList

Pomocná funkce v případě, když chce uživatel změnu výběru kolekcí, například od jiného uživatele. Funkce opět dotazuje server a databázi, aby dostala aktualizovaný seznam kolekcí, který bude odpovídat zadanému kritériu. Následuje opětovné renderování tabulky kolekcí hledaného uživatele.

4.3.9 Funkce Tour, flyTo a krokování

Tato kapitola popisuje fungování hlavní přednosti vizualizace v prototypu aplikace Maplay. Tím je možnost dynamickým přibližováním nad body v mapě zobrazit všechna navštívená místa, tedy ta, která byla vybrána do mapy. Náhled mapy se postupně pohybuje nad jednotlivými body, nad každým provede plynulé přiblížení a oddálení a pokračuje s vizualizací na další bod. Tato vizualizace probíhá chronologicky v pořadí podle data definovaného u každého bodu.

Jádrum tohoto efektu je funkce flyTo. Ta byla převzata z oficiálních stránek knihovny OpenLayers, část příklady (examples): <https://openlayers.org/en/latest/examples/animation.html>. Tato možnost vizualizace byla v počátku tvorby aplikace jedním z důvodů, proč si autor vybral pro mapovou knihovnu zrovna OpenLayers. Více o tom v kapitole 5.2. V průběhu tvorby se nicméně ukázalo, že bude nutné tyto funkce upravit pro dynamicky načítaná data.

Funkce flyTo využívá třídy ol.view.animate. Pomocí proměnných je na začátku nastaveno například doba trvání jednoho kroku animace. V této funkci se tedy řídí pouze proces jednorázového přiblížení k danému bodu

FlyTo je pak podstatou fungování samotné funkce Tour. Ta, jak už bylo popsáno výše, umožňuje efektně vizualizovat uživateli jeho či cizí kolekce bodů. Funkce tohle provádí vše najednou, pro svoje potřeby si všímá pouze data u bodů. Nerozlišuje kolekce v tom smyslu, že by animace probíhala paralelně či by se po animaci jedné kolekce přistoupilo k druhé. Je to nicméně jedním z předmětů v diskuzi na konci této práce. Ve funkci je na počátku definováno prázdné pole locations, které je naplněno obsahem proměnné

polePrvku. Přes iterace a funkce next pak proces animace projde postupně všechny body v tomto poli. Je zde nastavena časová prodleva 750 milisekund.

```
function tour() {
  var locations = [];
  polePrvku.map((val) => {
    locations.unshift(val);
  });
  var index = -1;
  function next(more) {
    if (more) {
      ++index;
      if (index < locations.length) {
        var delay = index === 0 ? 0 : 750;
        setTimeout(function() {
          flyTo(ol.proj.fromLonLat(locations[index]), next);
        }, delay);
      } else {
        alert('Tour complete');
      }
    } else {
      alert('Tour cancelled');
    }
  }
  next(true);
}
```

Obr. 4.17 Funkce Tour.

Dalším využitím funkce flyTo je možnost krokování mezi jednotlivými body. K tomu slouží dvě tlačítka: move back a move next. Uživatel si tak může postupně projít například svůj uložený výlet krok po kroku, tak jak navštívená místa následují po sobě a případně se vracet zpět. Tyto funkce pracují na stejném principu, ovšem s tím rozdílem, že se mění definovaný flyToIndex a to buď přičítáním nebo odčítáním jeho délky po jednom.

4.4 Uživatelské účty

Jak vyplývá ze zadání práce, prototyp aplikace obsahuje prvky mapové sociální sítě. Aplikace tedy funguje na bázi uživatelských účtů. Bez registrace a následného přihlášení ji nelze používat a anonymnímu návštěvníkovi se zobrazí pouze úvodní stránka aplikace. Jak je v textu již mnohokrát zmíněno, vyvinutá aplikace je prototyp, tudíž i prvky „sociálnosti“ tomu odpovídají. Tato podkapitola popisuje základ takové funkcionality v podobě registrace, přihlašování uživatelů. Další část tohoto základu byla popsána v podkapitole 4.3.1 o vyhledávání jiných uživatelů a připojení jejich veřejně viditelných kolekcí do vlastního mapového okna k vizualizaci. Další takové prvky, které by byly přínosné v dalším rozvoji prototypu aplikace či návrhu pro produkční prostředí se autor věnuje v kapitole 5.2 či v diskuzi na konci práce.

System uživatelských účtů není ničím originálním, na internetu existuje nespočet tutoriálů, jak tohle implementovat do svého webového projektu. Autor aplikace využil návod na stránce codeshack.io dostupné z: <https://codeshack.io/secure-login-system-php-mysql/>. System byl nejdříve vytvořen autorem zvláště přesně podle návodu, po dokončení aplikace byl do ní implementován a upravován dle finálních potřeb.

4.4.1 Registrace a přihlašování

Uživatel se může po příchodu do aplikace a krátkém zorientování v ní zaregistrovat. K tom slouží jednoduchá registrační formulář na obrázku 4.18.

The image shows a web registration form for 'maplay'. At the top left is a hamburger menu icon, and at the top right is the 'maplay' logo. The main heading is 'Register to Maplay'. Below this are four input fields: 'Username', 'Password', 'Password again', and 'Email'. At the bottom of the form is a red button with the text 'REGISTER' in white capital letters.

Obr. 4.18. Registrační formulář.

Aplikace chce po uživateli, aby si zvolil libovolné uživatelské jméno a heslo, které pro prevenci překlepu musí zopakovat. Jako poslední údaj je potřeba poskytnout e-mailovou adresu v platném formátu. Po odeslání formuláře je spuštěn skript v adresáři `phplogin` s názvem `register.php`. Tento skript se nejdříve pomocí jednotného skriptu pro celou aplikaci `dp-config.php` připojí k databázi, ověří platnost vstupů z formuláře (pokud se například vyskytují neplatné znaky v poli uživatelského jména, skript vrátí chybu) a přistupuje k samotnému odeslání dat do databáze do tabulky `accounts`. Pomocí příkazů si nejdříve například ověří, jestli uživatel se zadaným `username` již neexistuje. Pokud je vše v pořádku, pomocí SQL příkazu `INSERT` vloží data do tabulky. Hesla se šifrují metodou `password_hash`.

Skript je předpřipraven pro případné ověřování uživatele e-mailovou zprávou. Autor pro účely práce nezakládal server pro odesílání těchto ověřovacích zpráv, jelikož se jedná o prototyp. V tabulce `accounts` se do pole `activation_code` automaticky po registraci vyplní textem „activated“. Po úspěšné registraci může uživatel přistoupit k přihlášení pomocí uživatelského jména a hesla, které si zvolil v předchozím kroku.

4.4.2 PHP sessions

Téměř veškeré části stránky aplikace začínají příkazem `session_start()`. Sessions neboli česky relace, jak uvádí [pehapko.cz \(2020\)](#) zjednodušeně „umožňují sdílet údaje mezi jednotlivými http requesty a identifikovat uživatele. (...)Při vytváření sessions je uživateli vygenerován unikátní identifikátor sloužící k jeho rozpoznání a umožňující přistupovat ke konkrétnímu datovému souboru. Pro sdílení mezi jednotlivými stránkami tento identifikátor uložíme do cookie, kterou prohlížeč na server odesílá při každém načítání stránky. Data samotná jsou ve výchozím stavu uložena v serializované podobě v souboru uloženém na serveru a pojmenovaném podle uvedeného identifikátoru.“

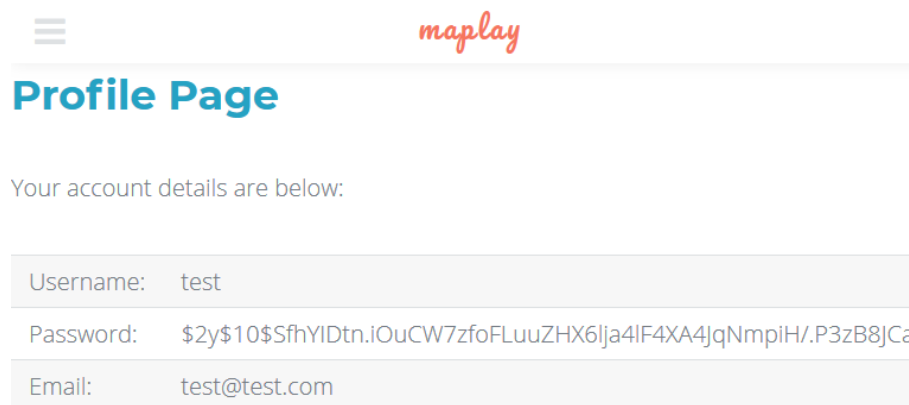
Aplikace používá relace nejen k identifikaci uživatele jako takové, ale slouží v ní i k přenosu mnoha identifikátorů kolekci či bodů k zobrazení mezi skripty, SQL dotazům nebo rozdílnému zobrazování tlačítek log in a log out.

```
<?php
session_start();
if (isset($_SESSION['loggedin']) && $_SESSION['loggedin'] == true) {
} else {
    header('Location: index.php');
    exit;
}
?>
```

Obr. 4.19 Začátek relace/session v kódu.

4.4.3 Stránka acc-profile.php

Stránka acc-profile.php plní v prototypu aplikace spíše informativní funkci. Uživatel na ní nalezne údaje o svém účtu. Těmi jsou uživatelské jméno, zašifrovaná podoba hesla a e-mailová adresa. Poslání této části aplikace je nicméně perspektivní v tom, že by zde měl mít uživatel možnost editovat své údaje o účtu. O tomto se také autor více zmiňuje v diskuzi.









Obr. 4.20 Stránka acc-profile.php.

4.5 Editace bodů

Podkapitola se věnuje poslední zatím nepopsané části aplikace. Tou je stránka mp-read.php, kterou uživatel v menu nalezne s označením Edit, funguje jako částečný tzv. CRUD systém. CRUD je anglická zkratka pro slova create, read, update a delete. Částečný z toho důvodu, že část pro vytváření dat se odehrává v jiné části (mp-points.php). Uživatelovi se na této stránce zobrazí přehledná tabulka se všemi jím přidanými body z databáze. U každého řádku jsou tlačítka pro editaci či smazání bodu.

[Back to the edit page](#)

Edit Points

| # | lat | lon | name | date | visibility | |
|-----|-------------|-------------|------------|------------|------------|---|
| 114 | 43.27720532 | 5.37506104 | Marsseille | 2019-04-15 | 1 |   |
| 115 | 45.7406934 | 4.81750488 | Lyon | 2019-04-13 | 1 |   |
| 117 | 48.51114602 | -2.74658203 | S Eriec | 2019-04-16 | 1 |   |

Obr. 4.21 Stránka mp-read.php.

Podstránka update.php zobrazí jednoduchý editační formulář pro editaci atributů daného bodu. Uživatel má zde možnost změnit souřadnice, avšak pouze přes číselné pole a nikoli v mapě, což je další část diskuze v rámci dalšího rozvoje aplikace. Po kliknutí na delete.php se stránka uživatele zeptá, zdali si je svou volbou opravdu jist a provede smazání daného bodu.

```

<?php
include 'functions.php';
$pdo = pdo_connect_mysql();
$msg = '';
// Check if the point id exists, for example update.php?id=1 will get the point with the id of 1
if (isset($_GET['id'])) {
    if (!empty($_POST)) {
        // This part is similar to the create.php, but instead we update a record and not insert
        $id = isset($_POST['id']) ? $_POST['id'] : NULL;
        $lat = isset($_POST['lat']) ? $_POST['lat'] : '';
        $lon = isset($_POST['lon']) ? $_POST['lon'] : '';
        $name = isset($_POST['name']) ? $_POST['name'] : '';
        $datum = isset($_POST['datum']) ? $_POST['datum'] : date('Y-m-d');
        // Update the record
        $stmt = $pdo->prepare('UPDATE points SET id = ?, lat = ?, lon = ?, name = ?, datum = ? WHERE id = ?');
        $stmt->execute([$id, $lat, $lon, $name, $datum, $_GET['id']]);
        $msg = 'Updated Successfully!';
    }
    // Get the point from the points table
    $stmt = $pdo->prepare('SELECT * FROM points WHERE id = ?');
    $stmt->execute([$_GET['id']]);
    $point = $stmt->fetch(PDO::FETCH_ASSOC);
    if (!$point) {
        exit('Point doesn\'t exist with that ID!');
    }
} else {
    exit('No ID specified!');
}
?>
```

Obr. 4.22 První polovina kódu stránky update.php.

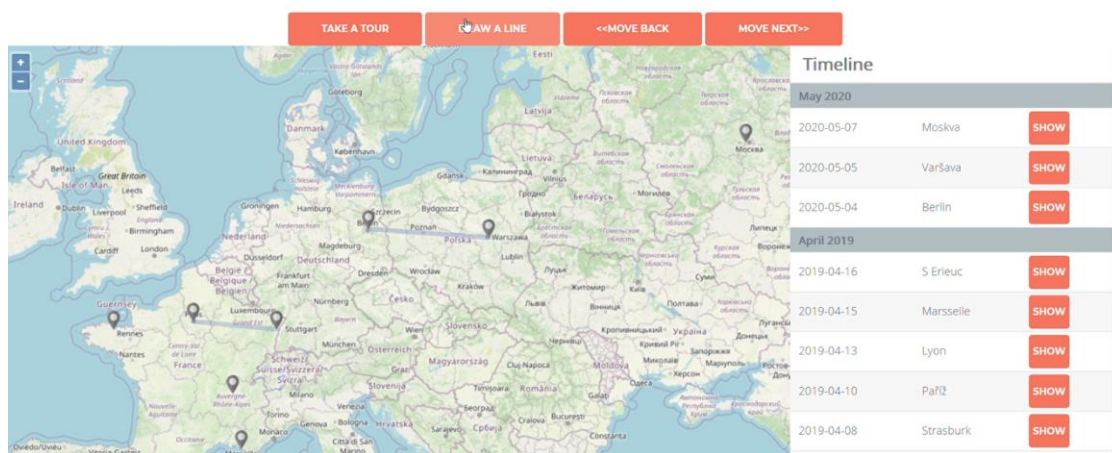
I pro tento částečný CRUD systém autor využil tutoriálu z webu dostupný na: <https://codeshack.io/crud-application-php-pdo-mysql/>, byl vytvořen odděleně od aplikace a následně začleněn.

V rámci dokončovacích prací došlo na ladění chyb, které byly objeveny v rámci finálního autorova testování. Hlavní úskalí, jimž autor v průběhu vývoje čelil, jsou popsány v diskuzi na konci práce.

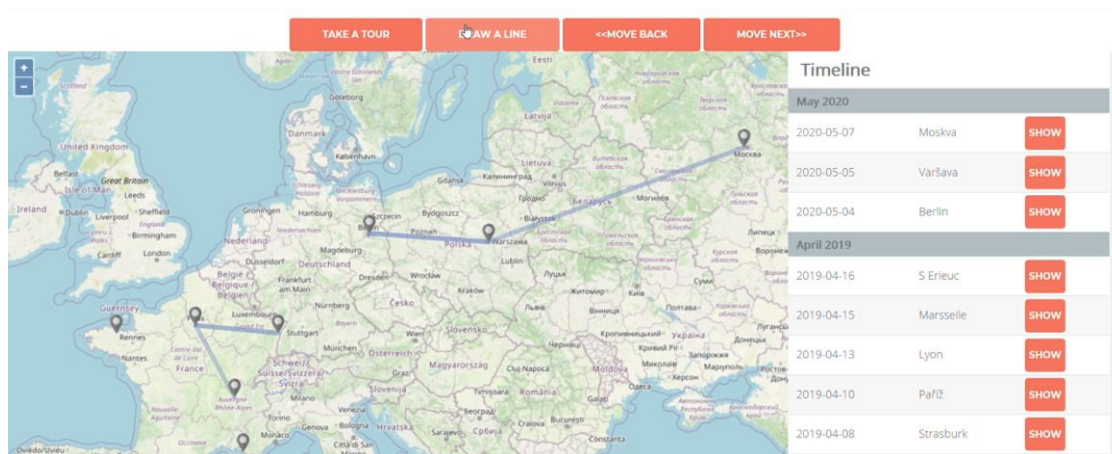
5 VÝSLEDKY

Tato kapitola popisuje výsledky, kterých bylo provedením této práce dosaženo. Výsledky jsou rozděleny do dvou částí. První část se zabývá popisem výsledků vývoje samotné aplikace, jak se podařilo naplnit vytyčené cíle. V druhé části bude věnován prostor pro návrhy možných řešení v komerčním prostředí, návrh možného vývoje do budoucna a podobně. Kromě toho byla na začátku práce provedena rešerše dostupné literatury, která se v různých pohledech týkala zadané práce. Dalším neméně důležitým krokem bylo prozkoumat a zhodnotit stávající podobná řešení na trhu, které vizualizaci polohy člověka v čase pojmají různými způsoby.

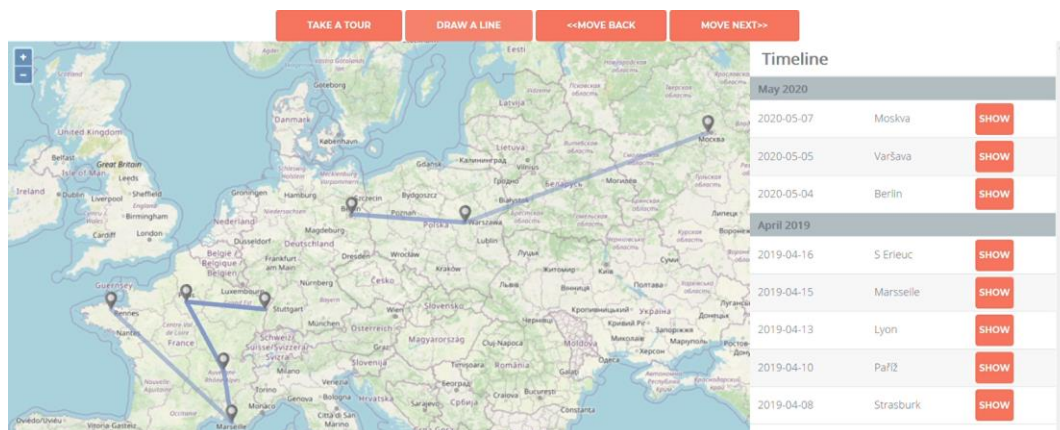
Na následujících obrázcích (5.1 až 5.3) je zachycena možnost vizualizace ve vyvinuté aplikaci Maplay funkcí vykreslení čáry u dvou kolekcí s několika body.



Obr. 5.1 První část po spuštění.



Obr. 5.2 Druhá část, kratší kolekce bodů je již vykreslena.



Obr. 5.3 Konečný výsledek vykreslování čáry mezi body u dvou kolekcí v mapě.

5.1 Výsledky vývoje aplikace

Hlavní částí této bakalářské práce bylo vytvoření funkčního prototypu webové aplikace. Cílem této aplikace bylo jednoduchou formou umožnit uživateli zaregistrovat si vlastní uživatelský profil, přihlásit se, nahrát data o poloze, editovat je, uložit do databáze a seskupovat do kolekcí. Následně si tato místa v mapě zobrazit, moci je vizualizovat a připojit záznamy od jiných uživatelů.

Zdárným dokončením aplikace, která dostala název Maplay, se podařilo tyto cíle naplnit. Autor se vypořádal s mnoha problémy při vývoji a k tomu si dokázal osvojit nové dovednosti potřebné k tvorbě webové aplikace. Také je snahou autora poskytnout programový kód volně ostatním volně k použití, aplikace je dostupná v repozitáři na stránce github.com/dominik-vit/maplay. Kromě tohoto zdroje je možné aplikaci používat na webovém serveru Katedry Geoinformatiky Univerzity Palackého v Olomouci dostupné na <http://gis.upol.cz/maplay/>.

5.2 Návrh pro produkční řešení

Tuto kapitolu autor věnuje subjektivnímu návrhu programových, mapových a databázových řešení, které by bylo vhodné použít v produkčním prostředí k vývoji mapové aplikace pro vizualizaci navštívených míst. Vzhledem k charakteru práce, který byl úzce zaměřen vybrané technologie vybrané na začátku, autor porovnává své širší zkušenosti s návrhy v kontextu vyvinuté aplikace. Součástí práce nebylo pokusit se vyvinout aplikaci podobného stylu na více platformách.

5.2.1 Mapové produkty

Pravděpodobně nejzajímavější poznatky vyplývají ze zkušenosti použití mapové knihovny OpenLayers.js. Tu si autor vybral ze dvou důvodů. Prvním je jejich podstata ve smyslu opensource řešení. Šlo o to pokusit se prozkoumat její stávající možnosti pro vývoj

takové aplikace. A také trochu snaha zkusit něco jiného, než převažující knihovnu Leaflet.js. Druhým důvodem byla možnost animací, kterou chtěl autor využít a nenalezl ji v takové podobě u žádného jiného řešení.

Jak se však během tvorby ukázalo, knihovna OpenLayers byla tím, co způsobovalo mnohokrát spíše komplikace než usnadnění tvorby. Knihovna je sice velmi rozsáhlá, existuje již mnoho let a byla jednou z prvních takových řešeních své doby. Na internetových zdrojích sice existuje nespočet řešení na mnoho problému, ty ale například naráží na nekompatibilitu mezi staršími a novějšími verzemi OpenLayers. Další komplikací především pro začátečníky může být na pochopení složitější dokumentace z pohledu názorných ukázek či vysvětlení návaznosti mezi jednotlivými kroky. Programátor je pak často odkázán hledat pomoc na zdrojích třetí strany, typicky stackoverflow.com. Po zavítání na stránky OpenLayers.com uživatel z nabízených příkladů užití nabude dojmu mnoha předpřipravených funkcí, které lze pro vlastní projekt jednoduše využít. Jak se právě autor sám s funkcemi animace přesvědčil, stálo ve výsledku více úsilí jejich úprava do funkční podoby než zvolit nejrozšířenější knihovnu Leaflet a použít jinou, byť široce v praxi ověřenou funkcionalitu z ní. Největší překážkou bylo používání dynamických dat z databází a uživatelských účtů. Pokud je čtenář znalý problematiky, určitě potvrdí, že drtivá většina projektů podobného rázu jsou buď proprietárními produkty jako například Google Maps Timeline, anebo otevřenými produkty, používající převážně knihovny Leaflet nebo velmi perspektivní knihovnu Mapbox.

Autor má za to, že reálný vývoj knihovny OpenLayers již prakticky v současnosti neprobíhá natolik, myšleno v kontextu velikosti uživatelské základny a progresivity nových funkcí, aby dokázal programátory zaujmout. The OpenLayers Dev Team je samozřejmě stále funkční entitou, nicméně s porovnáním s knihovnami Leaflet či Mapbox je na poli webových mapových řešení spíše okrajovým hráčem.

Pro pokračování ve vývoji aplikace Maplay v produkčním prostředí či pro vývoj zcela nové aplikace autor doporučuje pro začínající programátory využít knihovnu Leaflet. Jejimi přednostmi je jednak neustále se rozšiřující progresivní uživatelská základna nebo jednoduchost a přívětivost programového kódu. Pro složitější projekty či pokročilé programátory se nabízí knihovna Mapbox GL JS.

5.2.2 Databáze

Z pohledu databázových řešení je situace ohledně návrhu pro produkční řešení o něco jednodušší. Především z důvodu několikanásobně širšího trhu ve smyslu poptávky a nabídky. Autor pro vývoj prototypu aplikace použil databázový produkt MySQL, který používal skrze program phpMyAdmin. Vzhledem k dominanci MySQL na trhu, a tudíž obrovské uživatelské základně nenastal žádný problém v průběhu tvorby aplikace a autor ji doporučuje pro využití v produkčním prostředí. Ukládání všech dat probíhá do sloupců a řádků v tabulkách, a to jak pro prostorová data (navštívená místa) tak i pro data o

uživatelích. Právě jednoduchost a možnost mít všechna data v jedné databázi jsou přednostmi MySQL. Další z nich je například jednoduchá správa a komunikace serveru prostřednictvím skriptovací jazyk PHP. To je výhoda především pro začínající programátory.

V případě, že by vývoj aplikace zašel trochu dál v rámci rozšíření uživatelského zážitku, je nasnadě zvážit databázová řešení více adaptovaná na práci s prostorovými daty. Jsou jimi například PostGIS nebo SpatialLite. PostGIS je externí rozšíření databázového systému PostgreSQL, které podporuje geografické prvky, datové typy s funkcemi usnadňující práci s nimi. PostGIS má skvělou uživatelskou podporu. Je vyhledáván pro schopnost zpracovávat a analyzovat geografická data (Nétek, 2018).

Rozšířením uživatelského zážitku má autor na mysli doplnění vizualizace o funkce, které nejsou vhodné pro zpracování v prohlížeči (třeba na mobilních telefonech). Takovými funkcemi mohou být například různé prostorové analýzy dostupnosti, výpočet vhodné trasy mezi jednotlivými body podle zadaných požadavků (chůze vs. jízda autem) nebo převod dat do jiných formátů či jejich sdílení.

5.2.3 Programová řešení

Výběr programovacích jazyků je navázán na vybrané mapové řešení. Autor doporučuje použití JavaScriptu, jehož vybrané knihovny jsou základními kameny v aplikaci Maplay. Dále se osvědčilo použití PHP, které v aplikaci zajišťuje především dynamické získávání obsahu a autor ho doporučuje i pro produkční prostředí. Autor využil předstylované šablony, stojí nicméně za zvážení vzhled aplikace vyrobit zcela od základů, tak, aby vyhovoval moderním postupům webového designu a reflektoval potřeby aplikace.

6 DISKUZE

Práce je rozdělená do dvou částí, v praktické se autor zabýval vývojem samotné aplikace a navrhuje řešení pro produkční prostředí. V teoretické popisuje například již stávající řešení, porovnává je a vyzdvihuje rozdíly s vlastním řešením. Při řešení práce vyvstalo mnoho dílčích problémů, které bylo potřeba v postupném vývoji řešit. Autor musí podotknout, že většina komplikací byla různě velkou měrou způsobena jeho postupně se zmenšující nezkušeností v programování webových aplikací. V diskuzi je zmíněno pár z nich, které mají své opodstatnění. Také je popsáno pár bodů pro možný alternativní vývoj aplikace Maplay.

Knihovna OpenLayers by byla pravděpodobně tím, co by autor při teoretickém příštím vývoji webové aplikace s mapou vynechal a nahradil. Více byla tato problematika zmíněna v kapitole o návrhu pro produkční řešení, což je jedna s dílčími cíli této bakalářské práce. Za zmínku stojí nutnost základních funkcí jako třeba zobrazení markerů v mapě nebo přepínač podkladových map dohledávat u řešení třetích stran. Poté zkusit kolik z těchto rozšíření je funkčních (problém s kompatibilitou) nebo použitelných na data získávána dynamicky z databáze. Byť je množství takových pluginů relativně velké a autoři slibují jejich ověřenou funkčnost, málokdy šla implementace tak hladce jak se autor domníval. Z technického hlediska lze pak zvážit například použití předpřipravených frameworků pro vývoj webových aplikací, což by ovšem mohlo ze začátku práci narazit na autorovu nezkušenost.

V rámci samotného prototypu aplikace Maplay lze zmínit hned několik bodů, které by byly řešitelné relativně časově nenáročně, pokud by se jednalo o dlouhodobější vývoj. Takovými body jsou jednoznačně rozšíření funkcionalit při samotné vizualizaci, jako například různé druhy animací, větší grafické propojení s časovou osou, možnost zobrazení (okna vizualizace, nejen mapy) na celou obrazovku, rozlišení kolekcí v mapě podle uživatele, doplnění funkcí a vylepšení podoby markerů (fotografie míst, podrobnější popis) nebo třeba rozdělení animace při funkci Tour podle uživatelů do samostatných oken. U přidávání bodů by další vývoj obsahoval rozšíření o nahrávání dat z jiných souborů než GeoJSON, jako například GPX nebo KML (Keyhole Markup Language). Autor si je také vědom limitů při nahrávání datově objemných souborů. Testování limitů nicméně ukázalo nerelevanci výsledků z důvodu odlišnosti mezi prohlížeči, počítačových sestavách, a především malého počtu testů. Testování však nebylo cílem této práce a autor ho provedl pouze v omezené míře z vlastního zájmu. Pro další vývoj je také tedy nutné zaměřit se na to, jak velký objem dat by uživatele ještě neomezoval při práci s aplikací a případně nastavit limit. Co se týče funkcionalit uživatelských účtů, tak stojí za zmínku například možnost editace uživatelského profilu, přidání dalších charakteristik nebo třeba profilových obrázků. To všechno by potenciálně mohlo vést k rozšíření oněch prvků mapové sociální sítě. Nabízí se určitě zavedení možnosti propojit jednotlivé profily (obdoba přátelství na síti Facebook), seskupit je podle původu (rodina, kolegové z práce apod.) nebo sdílení nahraných dat externě pro jiné existující sociální sítě. V otázce nastavování viditelnosti dat pro jiné uživatele bylo diskutováno, zdali si přímo sám

uživatel nenastaví, že jeho celý profil bude „neviditelný“ z důvodu důkladnější ochrany osobních údajů. Nyní je možné dohledat jakéhokoli uživatele v databázi, pokud uživatel zná jeho jméno. Také editace vlastních dat je část, která byla na úkor vizualizace zanedbána. Ve vypracované podobě lze například editovat umístění bodů pouze změnou souřadnic v textovém poli, což je zcela nevhodné a je na místě tohle změnit na editaci přímo v mapě.

Jedna z věcí, která byla diskutována vzhledem k praktičnosti UI/UX a technickým možnostem bylo například omezení výběru kolekcí k zobrazení v mapě na tři. V případě dalšího vývoje by tohle omezení mohlo být rozhodně větší nebo vůbec žádné. Autor se však rozhodl, že ve stávající formě zobrazení by více kolekcí zároveň mohlo spíše škodit (splývání navzájem) než uživateli přinést něco zajímavého.

Celkový potenciál aplikace není ani z části zanedbatelný, a právě spíše narážel na limity, které jsou v této práci vícekrát zmiňovány. Další rozvoj by však vyžadoval buď řádově větší znalosti autora ve věcech programování a softwarového inženýrství nebo zapojení více osob do rozvoje. Už jen z podstaty vývoje takových aplikací není možné dokonale obsáhnout jak frontend tak backend. Autor se vývojem prototypu snažil nastínit, jak by si zadanou aplikaci představoval dosažitelnými prostředky a jak už bylo zmíněno, dává veškerý programový kód k dispozici jiným k jejich vlastnímu zdokonalování či navázání na vývoj aplikace.

7 ZÁVĚR

Hlavním cílem práce bylo vytvořit prototyp webové aplikace s prvky mapové sociální sítě. Aplikace disponuje možností uložit polohu navštívených míst v mapě, k tomuto záznamu uživatel přiřadí časový údaj, pomocí nichž si uživatel může svá data zobrazit v čase a název. Prototyp také umožňuje spojení takto do mapy zanesených informací pro dva až tři uživatele. Obsahuje mapové, databázové a programové řešení, která jsou plně funkční. Autor si osvojil základní principy při vývoji takového prototypu aplikace a získal cenné zkušenosti pro další osobní i profesní rozvoj.

Vytvořená aplikace slouží jako autorova představa, jak by taková aplikace v možnostech začínajícího programátora měla vypadat. Veškerý programový kód je dostupný volně ke stažení ze stránek bakalářské práce (<http://www.geoinformatics.upol.cz/dprace/bakalarske/vit20>) nebo v repozitáři na stránce github.com/dominik-vit/maplay. Bude autorovou ctí, pokud tento prototyp může alespoň částečně pomoci někomu jiného v rozvoji svých vlastních projektů či jako inspirace.

Cílem teoretické části práce bylo provést průzkum trhu a nalézt a popsat stávající produkty podobného zaměření, ať již na webové či mobilní platformě, které skýtají prvky sociální sítě. Tyto produkty byly podrobně popsány společně s rozdíly oproti vyvíjenému prototypu. V rámci teoretické části autor navrhnul mapové, databázové a programové technologie, které jsou vzhledem k jeho zkušenostem s touto prací nejvhodnější pro produkční prostředí. Snahou bylo porovnat různé alternativy a doporučit, jaké z nich by byly nejvhodnější. A to jak z pohledu začínajícího programátora, jakým byl na začátku on sám, tak i z pohledu realizace pokročilejších projektů. Je však nutné zmínit, že takové projekty bývají řešeny převážně proprietární cestou, na rozdíl od této práce. Ta měla mimo jiné zkusit technologie nabízené volně k použití na bázi free a opensource softwaru.

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

Dynamika Obyvatelstva. TOUŠEK, Václav, Josef KUNC a Jiří VYSTOUPIL. Ekonomická a sociální geografie. Plzeň: Vydavatelství a nakladatelství Aleš Čeněk, 2008, s. 68-90. ISBN 978-80-7380-114-4.

Geocoder Nominatim for OpenLayers. *GitHub.com* [online]. [cit. 2020-06-04]. Dostupné z: <https://github.com/jonataswalker/ol-geocoder>

GeoJSON. ESRI [online]. 2020 [cit. 2020-05-30]. Dostupné z: <https://doc.arcgis.com/en/arcgis-online/reference/geojson.htm>

Intensify - templated. Templated [online]. [cit. 2020-05-30]. Dostupné z: <https://templated.co/intensify>

JARCOVJÁK, David. TESTOVÁNÍ APLIKACE FOURSQUARE PRO SLEDOVÁNÍ POHYBU OSOB V OLOMOUCI [online]. Olomouc, 2013 [cit. 2020-05-30]. Dostupné z: http://www.geoinformatics.upol.cz/dprace/bakalarske/jarcovjak13/download/jarcovja_k_text.pdf. Bakalářská práce. Univerzita Palackého v Olomouci.

JQuery - AJAX Introduction. W3schools [online]. 2020 [cit. 2020-05-30]. Dostupné z: https://www.w3schools.com/jquery/jquery_ajax_intro.asp

JQuery návod - úvodní článek. JQuery návod [online]. [cit. 2020-05-30]. Dostupné z: <http://jquery-navod.cz/kategorie-ostatni-clanky/1-uvodni-clanek>

Mobile Operating System Market Share Worldwide. Statcounter.com [online]. 2020 [cit. 2020-05-30]. Dostupné z: <https://gs.statcounter.com/os-market-share/mobile/worldwide>

NĚTEK, Rostislav a Tomáš BURIAN. Free and open source v geoinformaticce. Olomouc: Univerzita Palackého, 2018. ISBN 9788024452913.

NĚTEK, Rostislav. Rich internet application pro podporu rozhodovacích procesů integrovaného záchranného systému [online]. Olomouc, 2015 [cit. 2020-05-30]. Dostupné z: <https://theses.cz/id/n6yi1j/portal.pdf>. Disertační práce. Univerzita Palackého v Olomouci.

NOVÁKOVÁ, Petra. Využívání geolokačních služeb v ČR [online]. Praha, 2015 [cit. 2020-05-30]. Dostupné z: <https://is.cuni.cz/webapps/zzp/download/130152346/?lang=cs>. Bakalářská práce. UNIVERZITA KARLOVA V PRAZE.

OpenLayers [online]. 2019 [cit. 2020-05-30]. Dostupné z: <https://openlayers.org/>

PHP MySQL Ajax Live Search. *Tutorial Republic* [online]. [cit. 2020-06-04]. Dostupné z: <https://www.tutorialrepublic.com/php-tutorial/php-mysql-ajax-live-search.php>

Sessions. Pehápko [online]. [cit. 2020-05-30]. Dostupné z:
<http://www.pehapko.cz/programujeme-v-php/sessions>

Sublime Text [online]. 2020 [cit. 2020-05-30]. Dostupné z:
<https://www.sublimetext.com/>

Virrantaus, K., Markkula, J., Garmash, A., Terziyan, Y.V., 2001. Developing GIS-Supported LocationBased Services. In: Proc. of WGIS'2001 –First International Workshop on Web Geographical Information Systems., Kyoto, Japan. , 423–432.

VONDRÁKOVÁ, Alena. POHYB MĚSTSKÉ POPULACE A JEHO KARTOGRAFICKÁ VIZUALIZACE [online]. Olomouc, 2007 [cit. 2020-05-30]. Dostupné z:
<http://www.geoinformatics.upol.cz/dprace/bakalarske/vondrakova07/>. Bakalářská práce. Univerzita Palackého v Olomouci.

What is JSON? W3schools [online]. 2020 [cit. 2020-05-30]. Dostupné z:
https://www.w3schools.com/whatis/whatis_json.asp

PŘÍLOHY

SEZNAM PŘÍLOH

Volné přílohy

Příloha 1 Poster

Příloha 2 CD

Popis struktury CD

Vit_bp – text bakalářské práce

Maplay – adresář obsahující aplikaci Maplay

Poster

Web – adresář obsahující webové stránky k práci