



MASARYKOVA UNIVERZITA  
PŘÍRODOVĚDĚCKÁ FAKULTA  
GEOGRAFICKÝ ÚSTAV



# MAPOVÁ APLIKACE PRO FOTOGRAFICKY ZAJÍMAVÁ MÍSTA V OKOLÍ BRNA

Bakalářská práce  
**Monika Bláhová**

Vedoucí práce: doc. RNDr. Tomáš Řezník, Ph.D.

Brno 2014

## **Bibliografický záznam**

Autor	Monika Bláhová Přírodovědecká fakulta, Masarykova univerzita Geografický ústav
Název práce:	Mapová aplikace pro fotograficky zajímavá místa v okolí Brna
Studijní program:	Geografie a kartografie
Studijní obor:	Geografická kartografie a geoinformatika
Vedoucí práce:	doc. RNDr. Tomáš Řezník, Ph.D.
Akademický rok:	2013/2014
Počet stran:	51+11
Klíčová slova:	webová kartografie, Google Maps API, webová mapa, Brno, PHP, JavaScript

## **Bibliographic Entry**

Author: Monika Bláhová  
Faculty of Science, Masaryk University  
Department of Geography

Title of Thesis: Map application for photographically interesting sites near Brno city

Degree programme: Geography and Cartography

Field of Study: Geographical Cartography and Geoinformatics

Supervisor: doc. RNDr. Tomáš Řezník, Ph.D.

Academic Year: 2013/2014

Number of Pages: 51+11

Keywords: web cartography, Google Maps API, web map, Brno, PHP, JavaScript

## **Abstrakt**

Cílem této bakalářské práce je kompletní návrh a implementace webové mapové aplikace pro správu fotograficky zajímavých míst v okolí města Brna. Tato webově dostupná aplikace využívá jako referenční mapové podklady základní mapy Google, a celá mapová část aplikace je řešena s využitím Google Maps API. Součástí práce je řešení kartografických aspektů webové mapové aplikace, je navržena celková kompozice a symbolika. Prostředí je tvořeno pomocí programovacích a značkovacích jazyků PHP, HTML a CSS. Interaktivita aplikace je pak zajištěna pomocí JavaScriptu. Technologie využité pro tvorbu aplikace jsou volně dostupné. Hlavním výstupem práce je webová mapová aplikace pro správu fotograficky zajímavých míst a návrh matice funkčních a mimofunkčních požadavků obecně platné pro obdobné aplikace. Práce obsahuje CD s výslednou aplikací.

## **Abstract**

The aim of this bachelor thesis is complete design and implementation of web map application for managing photographically interesting places near Brno city. This web-enabled application uses Google base map as reference map data, and entire map part of application is dealt with using the Google Maps API. Thesis includes solution of cartographic aspects of web map application, and it is designed overall composition and symbolism. The interface is created using programming and markup languages PHP, HTML and CSS. The interactivity of application is secured using JavaScript. All technologies used for creating application are freely available. The main outcome of this work is a web map application for managing photographically interesting places and design of matrix of functional and non - functional requirements generally applicable to similar applications. The thesis includes a CD with application files.



**Masarykova univerzita**  
**Přírodovědecká fakulta**



## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

**Student:** Monika Bláhová  
**Studijní program:** Geografie a kartografie  
**Studijní obor:** Geografická kartografie a geoinformatika

Ředitel Geografického ústavu Přírodovědecké fakulty MU Vám ve smyslu Studijního a zkušebního řádu MU určuje bakalářskou práci s tématem:

**Mapová aplikace pro fotograficky zajímavá místa v okolí Brna**  
**Map application for photographically interesting sites near Brno city**

### **Zásady pro vypracování:**

Hlavním cílem této bakalářské práce je vytvoření konceptu a pilotní aplikace pro správu vybraných bodů zájmu, konkrétně fotograficky zajímavých míst v okresech Brno - město a Brno - venkov.

Pro naplnění tohoto hlavního cíle postupujte přes následující body:

1. Proveďte rešerši existujících přístupů
2. Definujte funkční a nefunkční požadavky s důrazem na kartografické aspekty (např. mapová kompozice)
3. Analyzujte dostupná data a proveďte jejich sběr
4. Vytvořte pilotní aplikaci pro správu fotograficky zajímavých míst
5. Diskutujte dosažené výsledky a možnosti dalšího rozvoje aplikace

Rozsah grafických prací: podle potřeby

Rozsah průvodní zprávy: cca 30-40 stran

Seznam odborné literatury:

BROWN, A., KRAAK, M.J.: Web Cartography - developments and prospects. 1. vyd., Taylor and Francis, London, 2001. 213 s. ISBN 0-7484-0869-X.

PETERSON, M. P. et al.: Maps and the internet. 1. vyd., Elsevier Press, Oxford, Amsterdam, 2003. 451 s. ISBN 0-780080-442013.

NOGUERAS-ISO, J., ZARAZAGA-SORIA, F. J., MURO-MEDRANO, P. R. 2005. Geographic Information Metadata for Spatial Data Infrastructures. Springer: Berlin Heidelberg. 263 pages. ISBN 3-540-24464-6.

*Jazyk závěrečné práce:* čeština

*Vedoucí bakalářské práce:* RNDr. Tomáš Řezník, Ph.D.

*Podpis vedoucího práce:* .....

*Datum zadání bakalářské práce:* říjen 2013

*Datum odevzdání bakalářské práce:* do 12. května 2014

RNDr. Vladimír Herber, CSc.  
pedagogický zástupce ředitele ústavu

*Zadání práce převzal(a):* ..... *dne* .....

## **Poděkování**

Na tomto místě bych ráda poděkovala doc. RNDr. Tomáši Řezníkovi, Ph.D. za vedení této bakalářské práce a za cenné rady a připomínky. Dále bych chtěla jmenovitě poděkovat doc. Ing. Jitce Komárkové, Ph.D. za poskytnutí materiálů pro hodnocení webových mapových aplikací. Další poděkování patří Janu Hlavsovi a Lukáši Benkemu za přínosné konzultace z oblasti PHP a webových aplikací.

## **Prohlášení**

Prohlašuji, že jsem svoji bakalářskou práci vypracovala samostatně a veškeré při práci využitě informační zdroje jsem uvedla v seznamu použité literatury

Brno 10. 5. 2014

.....

Monika Bláhová

## Obsah:

<b>1</b>	<b>ÚVOD</b> .....	<b>10</b>
1.1	Cíl práce .....	10
1.2	Charakteristika zájmového území .....	10
1.3	Vymezení základních konceptů.....	11
1.3.1	Webová mapa.....	11
1.3.2	Fotografické místo.....	12
<b>2</b>	<b>HODNOCENÍ VYBRANÝCH WEBOVÝCH MAPOVÝCH ZDROJŮ</b> .....	<b>13</b>
2.1	Dostupné metody hodnocení webových mapových zdrojů .....	14
2.2	Kritéria hodnocení webových mapových zdrojů .....	14
2.3	Hodnocení funkčních a mimofunkčních požadavků .....	16
2.4	Výsledky hodnocení .....	17
2.4.1	Hodnocení Google maps.....	17
2.4.2	Hodnocení OpenStreet maps .....	18
2.4.3	Hodnocení mapy.cz.....	19
<b>3</b>	<b>TVORBA WEBOVÉ MAPOVÉ APLIKACE S VYUŽITÍM GOOGLE MAPS API</b> .....	<b>20</b>
3.1	Použité technologie .....	20
3.2	Google Maps API .....	21
3.3	Definice funkčních a mimofunkčních požadavků .....	21
3.3.1	Funkční požadavky .....	22
3.3.2	Mimofunkční požadavky .....	22
3.4	Realizace řešení.....	22
3.4.1	Struktura navržené aplikace .....	23
3.4.2	Realizace mapové části.....	24
3.4.2.1	Technologická stránka.....	26
3.4.2.2	Vizuální stránka.....	29
3.4.3	Správa uživatelů .....	31
3.4.3.1	Registrace nového uživatele.....	31
3.4.3.2	Přihlášení a odhlášení uživatele.....	33



3.4.4	Bezpečnost.....	35
3.4.5	Nahrávání souborů.....	38
<b>3.5</b>	<b>Návrh vizuální stránky aplikace .....</b>	<b>40</b>
3.5.1	Návrh uživatelského rozhraní.....	40
3.5.2	Návrh znakové sady.....	43
<b>3.6</b>	<b>Webová implementace .....</b>	<b>44</b>
<b>4</b>	<b>ZÁVĚR .....</b>	<b>45</b>
<b>5</b>	<b>SEZNAM POUŽITÝCH ZDROJŮ.....</b>	<b>46</b>
<b>6</b>	<b>SEZNAM ZKRATEK.....</b>	<b>49</b>
<b>7</b>	<b>PŘÍLOHY.....</b>	<b>50</b>

# 1 ÚVOD

## 1.1 Cíl práce

Hlavní cíl této bakalářské práce spočívá ve vytvoření webové mapové aplikace pro správu fotograficky zajímavých míst v území vymezeném v okolí města Brna. Cílem je tedy vytvoření a propojení jednotlivých částí aplikace a následná implementace celého provedení.

Mezi dílčí úkoly této práce patří pak teoretické objasnění základních pojmů a hodnocení vybraných charakteristik dostupných webových mapových aplikací.

## 1.2 Charakteristika zájmového území

Jako zájmová oblast bylo pro vypracování práce zvoleno území odpovídající okresům Brno – město a Brno – venkov. Úkolem práce je vytvoření digitální mapy fotograficky zajímavých míst, která je tvořena v prostředí Google Maps API. Zájmové území je vymezeno pomocí obdélníku (Obr. 1), který obsahuje celou plochu okresu Brno město a zasahuje i do okresu Brno – venkov a okrajově i do okresu Blansko. Obdélník je vymezen zeměpisnými souřadnicemi 49°18' SŠ, 49°60' JŠ, 16°25' ZD a 16°6' VD.

Okres Brno – město, je tvořen statutárním městem Brnem, které je druhým největším městem České republiky a centrem Jihomoravského kraje. Počet obyvatel k 30. 6. 2012 činí 378 575 (CZSO 2012), a celková rozloha okresu je 230,2 km<sup>2</sup> (CZSO 2012). První zmínky o trvalém osídlení na místě dnešního Brna pochází z 11. století. Díky výhodné poloze se město stalo křižovatkou obchodních cest. Brno je dnes členěno na 29 městských částí, které ve své nynější podobě existují od roku 1990.

Okres Brno-venkov tvoří pás obklopující okres Brno – město. Celková rozloha okresu činí 1 499 km<sup>2</sup> a najdeme zde 187 obcí (CZSO 2011). Jedná se o průmyslově zemědělský region, vyznačující se značnou dopravní intenzitou. Okres nabízí velké množství přírodních i stavebně historických památek. Okres sousedí s okresy Žďár nad Sázavou, Blansko, Vyškov, Břeclav, Znojmo a Třebíč.

Vzhledem k tématu bakalářské práce je podstatným faktem, že v obou zmíněných okresech najdeme mnoho kulturních i přírodních zajímavostí.



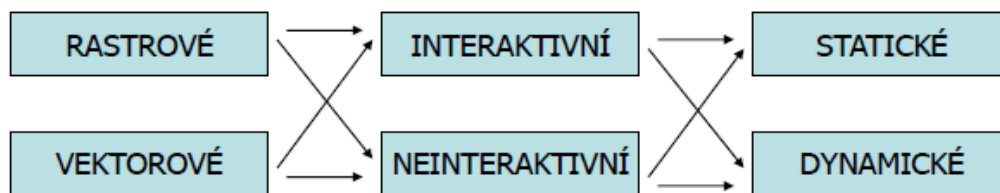
**Obr. 1:** Vymezení zájmového území

## 1.3 Vymezení základních konceptů

### 1.3.1 Webová mapa

Rozvoj internetu a technologií WWW (Word Wide Web) umožnil vytvoření nového prostředí pro sdílení informací. Tento rozvoj ovlivnil i vývoj map, které dostaly novou, digitální podobu. Tyto digitální mapy mohou uživatelům nabídnout mnohem komplexnější informace než mapy analogové. *Webová kartografie* se jako nový obor pak zabývá právě tvorbou, přenosem a zobrazováním digitálních map v prostředí WWW. Mapy vytvořené prostřednictvím webové kartografie můžeme označit jako *webové mapy*. Jedná se o datové struktury schopné sdílení v rámci sítě, které zobrazují prostorová data formou mapových podkladů. Webové mapy patří dnes mezi hlavní volně dostupné zdroje prostorových informací. Tento fakt vychází z globální dostupnosti a uživatelské jednoduchosti WWW prostředí.

Webové mapy můžeme podle několika kritérií dělit. První možností je členění dle výstupní grafické prezentace na mapy rastrové a vektorové. Podle způsobu tvorby dále rozdělujeme mapy na statické, které zobrazují stále stejnou sadu dat a mapy dynamické, které umožňují uživateli změnit zobrazovaná data. Posledním způsobem dělení je pak rozlišení map neinteraktivních, které umožňují pouze prohlížení a map interaktivních, které poskytují komunikaci mezi uživatelem a serverem. Tyto mapy poskytují mnoho funkcí jako je například změna měřítka nebo vyhledávání objektů. Jakoukoliv webovou mapu je pak možné z každého hlediska zařadit, jak vidíme na obrázku.



**Obr 2.:** Dělení webových map [21]

### 1.3.2 Fotografické místo

Úkolem vytvořené aplikace je prezentace tzv. fotografických míst. Fotografické místo si můžeme přestavit jako bod zájmu, který má své konkrétní místo v mapě, a je jasně odlišen od svého okolí i od ostatních bodů. Cílem aplikace je umožnit uživateli tyto body zájmu prohlížet, vytvářet a dále spravovat. Zvláštní vlastností každého takového bodu je fakt, že mu bude přiřazena galerie, umožňující prohlížení fotografií, pořízených právě v tomto bodě.

Výsledná mapová aplikace bude takovýchto bodů obsahovat velké množství. Přihlášeným uživatelům bude mimo jiné umožněno další body umístit do mapového pole. Pro správnou funkčnost je tedy nutné tyto body určitým způsobem třídit.

Proto byly vytvořeny kategorie fotografických míst (respektive přidáných fotografií), do kterých jsou ve finální podobě aplikace tříděny. Kategorie těchto bodů byly vytvořeny se snahou obsáhnout možná témata budoucích fotografií. Výsledné kategorie jsou následující:

- *Příroda*
- *Architektura*
- *Kultura*
- *Lidé*
- *Událost*
- *Gastronomie*
- *Momentka*

Každá z těchto vymezených kategorií je odlišena vlastním bodovým mapovým symbolem.

## **2 HODNOCENÍ VYBRANÝCH WEBOVÝCH MAPOVÝCH ZDROJŮ**

Před samotnou tvorbou vlastní webové aplikace je důležitým krokem získání poznatků o již existujících řešeních. Hodnocením webových mapových zdrojů a webových GIS (Geographic Information System) se v České republice zabývá například Komárková (2008), která se zaměřuje na hodnocení kvality webového GIS a webových map nebo Voženílek (1999), který navrhuje způsoby hodnocení map tematických.

Výsledná webová aplikace pro správu fotograficky zajímavých míst není ničím jiným než právě interaktivní webovou mapou, jejíž základní funkčnost se podobá běžným webovým mapovým portálům. Jedinou přidanou hodnotou je vlastní tvorba tematické nadstavby ve formě bodové vrstvy zobrazené nad vrstvami poskytnutými k využití v rámci mapové aplikace Google. Tato bodová vrstva je v plném provozu výsledné aplikace tvořena a spravována samotnými uživateli. To ovšem výrazně neodliší výslednou mapu od běžných mapových portálů v oblasti funkčnosti. Proto je hodnocení existujících řešení zaměřeno právě na tyto aplikace.

Pro hodnocení jsou vybrány mapové portály [maps.google.cz](http://maps.google.cz) (dále jen GM), [mapy.cz](http://mapy.cz) (dále jen MCZ) a [openstreetmap.org](http://openstreetmap.org) (dále jen OSM).

Google maps je služba poskytována firmou Google. Umožňují prohlížení mapových podkladů celého světa a patří mezi často vyhledávaný zdroj informací. K hodnocení je tato služba vybrána především z důvodu využití Googlemaps API (Application programming interface) pro tvorbu mapové aplikace pro správu fotografických míst. Tato API umožňuje bezplatné využití mapových podkladů Googlemaps pro tvorbu a správu vlastních mapových řešení.

Mapy.cz, je mapová služba provozovaná korporací seznam.cz. Jedná se o produkt firmy Mapy.cz, s.r.o. Dostupné mapové podklady zahrnují na rozdíl od Googlemaps pouze Evropu. Stejně jako v prostředí Googlemaps je i zde umožněno tvořit vlastní mapové aplikace pomocí API mapy.cz, s využitím dostupných podkladů MCZ. Právě potřeba srovnání obou aplikací, byla podmínkou k zařazení této webové mapové služby do hodnocení.

OpenStreet maps jsou na rozdíl od výše uvedených webovým map, tvořeny samotnými uživateli. Jedná se o projekt pod záštitou OpenStreetMap Foundation. Podkladová data této aplikace jsou poskytována volně pro jakékoliv účely, s podmínkou uvedení zdroje dat. Hlavní myšlenkou celého projektu je poskytovat kvalitní prostorová data, pro následnou tvorbu topografických map široké veřejnosti. Přímou na webu je navíc možné mapy editovat. Hodnocení webových map OSM tedy umožňuje reálné srovnání produktů komerčních a volně poskytovaných.

## 2.1 Dostupné metody hodnocení webových mapových zdrojů

V současné době existuje velké množství způsobů a postupů hodnocení webových map. Tato tematika je zpracovávána laiky i odborníky, což způsobuje velmi různorodou kvalitu výsledných hodnocení. Podle Voženilka (6, str. 142) je nutné provádět analýzu a hodnocení mapových děl vždy s ohledem na konkrétní způsoby jejich využívání s cílem zjistit jejich vlastnosti, kvalitu a vhodnost pro daný účel. Tento fakt naprosto platí pro analogové mapy. Většina webových map ale dnes uživatelům, mimo svoji hlavní funkci, nabízí více možností využití. Způsoby využívání webových map pak často určují sami uživatelé svými nároky.

U webových mapových zdrojů je častým předmětem hodnocení použitelnost nebo vizualizace mapy. Tyto a další aspekty lze řešit mnoha cestami. Nejčastěji využívanými postupy pro hodnocení aplikací jsou však *heuristická analýza* a *hodnocení produktu z pohledu uživatele*.

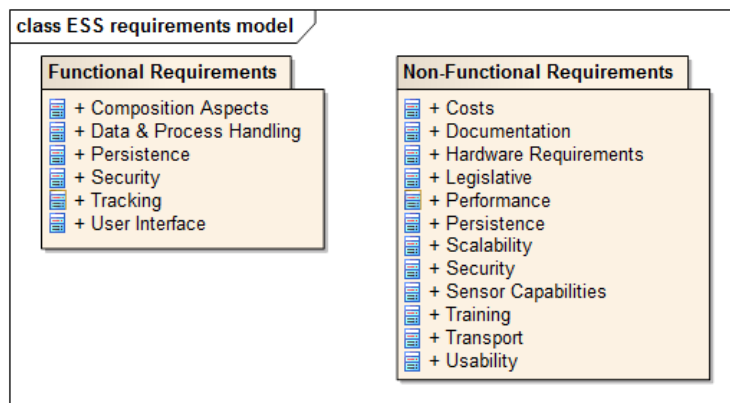
*Heuristické hodnocení (Heuristická analýza)* je dnes nejčastěji praktikovanou metodou testování použitelnosti. Principem této metody je vytvoření sady heuristik (principů) použitelnosti a následné hodnocení uživatelského rozhraní na jejich základě. Heuristiky si můžeme popsat jako sadu dotazů, na které lze odpovědět ano/ne. Výsledkem metody je pak sada problémů použitelnosti, které se hodnotí s ohledem na jejich závažnost. Tato metoda byla poprvé představena J. Nielsenem. Právě heuristická analýza je použita pro hodnocení vybraných webových mapových portálů pro účely bakalářské práce. Podle Komárkové (7, str. 323) patří mezi hlavní přednosti heuristického hodnocení menší časová náročnost a menší požadavky na speciální prostory a vybavení. Nevýhodou metody je pak podle Komárkové skutečnost, že hodnocení provádí odborníci na použitelnost, ale nejsou zapojeni reprezentanti uživatelů.

Další možností posuzování je právě *hodnocení webových zdrojů z hlediska uživatele*. U této metody hodnocení nelze definovat přesná pravidla pro postup. Navíc podle Komárkové (2008), v oblasti hodnocení webových mapových zdrojů zaměřených na uživatele, neexistuje mnoho prací. Dalším faktem zůstává, že výsledky této metody hodnocení budou vždy do jisté míry subjektivní. Tímto způsobem se zabývá například Wachowicz (2007), která řeší problematiku uživatelské spokojenosti při práci s aplikací, která přiměje uživatele se k produktu vracet opakovaně.

## 2.2 Kritéria hodnocení webových mapových zdrojů

Výsledná hodnotící kritéria jsou tedy stanovena s ohledem na požadavky, která by měla ve výsledné podobě splnit i aplikace pro správu fotograficky zajímavých míst. Hodnocení webových mapových zdrojů je provedeno ve dvou

kategoriích, konkrétně hodnocení funkčních (Functional requirements) a mimofunkčních (Non-functional requirements) požadavků. Na obrázku uvedeném níže můžeme vidět kompletní doporučený rozsah hodnocení v obou kategoriích.



**Obr. 3:** Doporučený rozsah hodnocení mapových aplikací (Řezník, 2012)

Pro komplexní hodnocení s úplnou vypovídací hodnotou je nutné hodnotit všechna kritéria současně. Jedná se o obsáhlý úkol, který by rozsahem pravděpodobně odpovídal celé bakalářské práci. Proto byla pro potřeby této práce vybrána pouze některá kritéria, ostatní jsou z hodnocení zcela vypuštěna. Cílem této části je pouze základní porovnání webových mapových aplikací, nikoliv jejich komplexní hodnocení. Proto jsou zohledněna pouze ta kritéria, která jsou uvažována při tvorbě vlastní webové mapové aplikace. I takto zjednodušené hodnocení poskytuje základní přehled o hlavních výhodách a nevýhodách hodnocených zdrojů. Jsou vybrána následující kritéria.

#### *Funkční požadavky (Functional Requirements)*

- Manipulace s daty a průběh procesů (Data & Process Handling)
- Uživatelské rozhraní (User Interface)
- Kompozice (Composition Aspects)

#### *Mimofunkční požadavky (Non – Functional Requirements)*

- Náklady (Costs)
- Dokumentace (Documentation)

V oblasti funkčních požadavků je hodnocena manipulace s daty a procesy. Do této oblasti spadá hodnocení základní funkčnosti, hlavně v rámci pohybu mapového pole, jako je posouvání nebo zoom. Dále sem spadá funkčnost mapových aplikací ve vybraných prohlížečích.

Dalším hodnocenou částí v rámci funkčních požadavků je uživatelské rozhraní. Hodnocení je v této části zaměřeno na existenci a snadnou dostupnost základních ovládacích prvků webových mapových aplikací jako jsou ovládací

panely pro posouvání nebo přiblížení. Dále je uvažována i přítomnost nástrojů měřících, a nástrojů pro tisk a uložení mapy.

Posledním hodnoceným aspektem v rámci funkčních požadavků je kompozice mapové služby. Hodnocení je v této části zaměřeno na přítomnost základních kompozičních prvků mapy. Pro vybrané webové mapové služby jsou všechna vybraná kritéria podrobně uvedena v tabulce níže (Příloha 1.).

V oblasti mimofunkčních požadavků jsou hodnoceny kritéria dvou oblastí, konkrétně nákladů a dokumentace. V oblasti nákladů je řešena pouze otázka volné dostupnosti dat. V oblasti dokumentace je hodnoceno, zda můžeme pro danou mapovou aplikaci nalézt kompletní a snadno dostupnou dokumentaci, která má vhodnou formu. Zohledněna je v hodnocení také dohledatelnost zdroje dat. Konkrétní kritéria opět najdeme v tabulce, uvedené v rámci dalšího textu (Příloha 2.).

Problematika ostatních kritérií není uvažována ani v rámci vývoje aplikace pro bakalářskou práci. Tyto body funkčnosti jsou nad rámec obsahu a mohou být předmětem dalšího vývoje aplikace na příklad v rámci navazující diplomové práce.

V rámci výsledné aplikace je funkčnost spojená s mapovým polem zajištěna prostřednictvím nástrojů Google Maps API. Tato aplikace tedy zajišťuje funkčnost v oblasti kompozice, uživatelského rozhraní a práce s daty. Zbývající otázky funkčnosti jsou řešena prostřednictvím programovacích a značkovacích jazyků.

Pro jednotlivé oblasti hodnocení jsou využita konkrétní hlediska hodnocení stanovená na základě studie (Komárková *et al.*, 2007), ze které jsou vybrána vhodná kritéria pro popsání funkčnosti aplikací ve vybraných ohledech.

Výsledek hodnocení každého kritéria je vyjádřen třemi možnostmi. Aplikace buď plně splňuje daný požadavek (označeno jako zelené pole, Příloha 1., Příloha 2.). Tento výsledek vyjadřuje, že daný prvek, informace nebo funkce je pro uživatele snadno dostupný a jednoznačně znázorněný, ideálně z hlavní stránky aplikace bez potřeby dalšího hledání. Druhou možností je částečné splnění požadavku. (označeno jako žluté pole, Příloha 1, Příloha 2.). Tímto hodnocením je vyjádřena existence daného ovládacího prvku, informace nebo služby v rámci webové mapy. Nenajdeme je již na hlavní straně, ale je nutné hledat v dokumentaci aplikace, nebo v jiné části uživatelského rozhraní. Tento proces uživateli znesnadní orientaci a znepríjemní práci s mapou. Poslední možností hodnocení je nesplnění požadavku (označeno jako červené pole, Příloha 1, Příloha 2.). V takovém případě hodnocený prvek, informace nebo funkce v aplikaci chybí.

## **2.3 Hodnocení funkčních a mimofunkčních požadavků**

V této kapitole jsou vyhodnoceny vybrané tři webové mapové služby. Jejich kvalita je hodnocena na základě předem vybraných kritérií. I při hodnocení pouhých třech zdrojů najdeme mezi aplikacemi výrazné rozdíly. Ani jedna z aplikací vybraných k hodnocení nesplňuje všechna stanovená kritéria. Hlavním



problémem je u vybraných mapových služeb absence základních kompozičních prvků, jako je legenda, nebo určité vyjádření měřítka. Z hlediska funkčnosti mají zdroje podle hodnocení velmi podobnou kvalitu.

Google maps můžeme hodnotit zdaleka nejlépe v oblasti Manipulace s daty a průběhu procesů. Uživateli je umožněna snadná práce s mapou, pomocí jednoduchého intuitivního ovládání. Naopak můžeme nalézt nedostatky v oblasti kompozice (Composition Aspects) a v oblasti uživatelského rozhraní (User Interface). Výhodou Google maps je možnost prohlížení kvalitních mapových podkladů pro celý svět. Aplikace poskytuje kvalitní satelitní snímky, a obsahuje i atraktivní funkce jako je Street View nebo Google Earth. Tyto fakty do hodnocení zahrnuty nejsou, ale nelze je opomenout při hodnocení celkové ho dojmu. Právě podobné nadstandardní vlastnosti lákají uživatele při výběru mapového zdroje.

Naopak OSM splňují dobře kritéria jak v oblasti kompozice, tak v oblasti práce s mapou (Data and Process Handling). Jako jediná aplikace poskytuje snadno přístupnou legendu, a umožňuje pracovat s měřítkem mapy.

Mapy.cz mají celkový výsledek hodnocení nejlepší. Jako u obou předchozích zdrojů najdeme i zde nedostatky. Opět chybí legenda vrstev, která je dostupná pouze z dokumentace.

## **2.4 Výsledky hodnocení**

Jako první jsou hodnoceny požadavky funkční. U každého z hodnocených kritérií je možné jednoduše stanovit, zda mapová aplikace požadavek splňuje, nesplňuje, nebo zda ho splňuje pouze částečně. Výsledky hodnocení jsou pak přehledně zobrazeny tabulkou (Příloha 1.). Jak vyplývá z hodnocení ani jedna z mapových aplikací nesplňuje všechna zadaná kritéria. Pro výsledný úsudek je ovšem důležitým faktem, že ne všechna stanovená kritéria jsou pro celkový dojem stejně důležitá. Následně jsou stejným způsobem hodnoceny i požadavky mimofunkční (Příloha 2.).

### **2.4.1 Hodnocení Google maps**

Prvním hodnoceným zdrojem byly GM. V oblasti uživatelského rozhraní má tato aplikace několik nedostatků. Neposkytuje například nástroje pro přímé měření linií. Měření liniových prvků je uskutečněno pouze formou měření trasy, kterou si uživatel může libovolně stanovit. Touto formou pak aplikace umožní měřit jak vzdušnou čarou tak pomocí liniových prvků. Velkou nevýhodou aplikace je ovšem absence nástroje pro export mapy ve formě obrázku. Sdílení mapy je umožněno pouze přes URL (Uniform Resource Locator). Další nesnází pro uživatele může být tisk mapy, který sice umožněn je, ale ne přímo z hlavní strany aplikace. Mapová služba GM nenabízí v nové verzi ani klasický nástroj pro posouvání mapového pole. Většina uživatelů ovšem používá intuitivní posouvání pomocí myši, a tak lze tento nedostatek opomenout.

V oblasti Manipulace s daty a průběhu procesů, nemá aplikace výraznější nedostatky. Jediným chybějícím prvkem je nastavitelné měřítko mapy. Tento prvek lze ale považovat spíše za nadstavbový. Celkově je aplikace velmi snadno a intuitivně ovladatelná.

V oblasti Kompozice chybí u GM některé základní prvky. Za největší nedostatek lze považovat absenci legendy vrstev. Aplikace obsahuje grafické měřítko, ale neposkytuje uživateli jeho číselné vyjádření.

Celkově se jedná o přehledný mapový zdroj, který splňuje všechny základní požadavky na funkčnost bez velkých nedostatků. Zároveň jsou prostřednictvím GM poskytovány kvalitní a celistvé mapové podklady pro celý svět. Výhodou jsou nesporně i služby Street view a Google Earth, které ostatní mapové portály nenabízejí.

Mimofunkční požadavky (Příloha 2.) splňuje mapová aplikace GM bez problémů. Jelikož se jedná o komerční produkt, nejsou mapové podklady volně poskytovány. Jsou ovšem poskytnuty prostřednictvím aplikace pro vytváření vlastních map v prostředí Google Maps API. Po vytvoření mapy a uvedení zdroje dat lze podklady využít pro prezentaci na webu.

#### **2.4.2 Hodnocení OpenStreet maps**

Jediným nekomerčním hodnoceným mapovým produktem jsou OSM. První hodnocenou oblastí je uživatelské rozhraní. OpenStreet maps jako jediné z hodnocených aplikací neposkytují možnost měření vzdálenosti pomocí linií a to ani pomocí měření tras. Mapu lze uložit i tisknout, není to uživateli ovšem umožněno přímo z hlavní strany aplikace. Stejný problém je i u nástroje pro posouvání, který v nejnovější verzi mapy na hlavní stránce chybí.

V hodnocené oblasti Manipulace s daty a průběh procesů, dosahují OSM nejlepších výsledků. Uživateli jsou umožněny všechny základní operace. Všechny hodnocené aspekty funkčnosti jsou navíc dostupné z hlavní stránky aplikace s jedinou výjimkou, kterou je změna stylu mapy. Tu uživatel ovšem snadno najde v bočním menu. Jako jediná z hodnocených, poskytuje tato aplikace uživateli možnost nastavit měřítko ručně.

V části hodnocení kompozice má mapová aplikace OSM také dobré výsledky. Všechny prvky hodnocení jsou dostupné. Číselné měřítko a legendu mapy, najdeme opět až po otevření menu.

OpenStreet maps jako jediná aplikace poskytuje mapová data volně bez podmínek využití. Na webových mapách pak můžeme najít datum veškerých editací, což dovolí uživateli přehledně sledovat změny všech úprav i jejich datum a autora.

### 2.4.3 Hodnocení mapy.cz

U mapové služby MCZ bylo opět jako první hodnoceno uživatelské rozhraní. Služba, stejně jako GM, nenabízí přímé nástroje pro měření, ale poskytuje uživateli možnost zadat do mapy vlastní trasu, a tím i délku zadaného liniového prvku. Výhodou oproti předchozím mapovým zdrojům je snadná možnost mapu vytisknout nebo uložit jako obrázek, a to přímo kliknutím na funkční klávesu na hlavní straně aplikace. Mapy.cz dále obsahují jako jediný z testovaných zdrojů nástroje jak pro přiblížení nebo oddálení mapy, tak pro její posouvání.

Z hlediska práce s daty nemá tento mapový portál výraznější nedostatky. Chybí zde možnost posouvat mapové pole pomocí kurzorových šipek, ale mapou lze bez problému intuitivně pohybovat pomocí myši. Chybí zde schopnost vycentrovat mapu na požadovaný prvek. To může způsobit horší orientaci uživatele v mapovém poli, pokud například čeká na výsledky vyhledávání. Opět zde také chybí možnost zadat ručně měřítko mapy.

V oblasti kompozice splňuje tato mapová služba kritéria nejlépe ze všech hodnocených zdrojů. Legendu mapy sice nenajdeme přímo na hlavní straně, ale pro zájemce je snadno dostupná v dokumentaci aplikace.

V oblasti mimofunkčních požadavků (Příloha 2.) nemá mapová služba nedostatky. Podobně jako u GM jsou mapové podklady poskytovány prostřednictvím API pro vytvoření vlastní mapy. Tato aplikace API mapy.cz je poskytována volně a podmínkou je pouze uvedení mapové služby mapy.cz jako zdroje dat.

## **3 TVORBA WEBOVÉ MAPOVÉ APLIKACE S VYUŽITÍM GOOGLE MAPS API**

Jak již bylo zmíněno, výsledná webová mapová aplikace je tvořena v prostředí Google Maps API (Kapitola 1). Prostředí Google maps API poskytuje rozhraní pro programování mapových aplikací nad podkladovými daty Google maps. Pro vytvoření webové mapové aplikace je nutná základní znalost standardních programovacích jazyků a technologií, které jsou popsány v následujícím textu.

### **3.1 Použité technologie**

#### **PHP**

*„PHP (zkratka pro PHP Hypertext Preprocessor) je skriptovací jazyk zabudovaný na straně serveru. To znamená, že pracuje uvnitř dokumentu HTML a propůjčuje mu tak schopnost generování požadovaného obsahu.“ [3, str. 6]*

#### **HTML**

*„HyperText Markup Language (HTML) je značkovací jazyk pro hypertext. Je jedním z hlavních jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na internetu.“ [29]*

#### **CSS**

*„Kaskádové styly (CSS) představují jednoduchý mechanismus přidávání stylů (např. fontů, barev, mezer) do webových dokumentů.“ [34, str.]*

#### **JavaScript**

*„JavaScript je mutiplatformní interpretovaný skriptovací jazyk s podporou jednoduchého objektově orientovaného programování (OOP). JavaScript běží na straně klienta (v internetovém prohlížeči) na rozdíl od PHP, Perlu, či jiných jazyků, které běží na straně serveru.“ [11]*

#### **MySQL**

*„MySQL je velmi rychlý robustní relační databázový systém. Databáze vám umožní efektivně ukládat, hledat, třídit a získávat data. MySQL server se stará*

*o to, aby k databázi mohlo přistupovat více uživatelů zároveň, a zajišťuje, aby to byli pouze oprávnění uživatelé. Jinými slovy – MySQL je multi-uživatelský a multi-threadový server. Používá SQL (Structured Query Language), což je celosvětově používaný standardní dotazovací jazyk pro databáze.“ ([28] str. 29)*

### **3.2 Google Maps API**

Výsledná mapová aplikace je v rámci bakalářské práce vytvářena v prostředí Google Maps API. Tato aplikace je pouze jedním z mnoha řešení, nabízených společností Google, která lze prostřednictvím JavaScriptu implementovat do vlastních webových stránek. Obdobou této aplikace je Google Static Maps API, která umožňuje vložení statické mapy Google, jako obrázku, na jakékoli stránky, bez nutnosti znalosti programování. Mezi další populární aplikace, které Google poskytuje, patří například Google AJAX search API, která umožní umístit na webové stránky vyhledávací nástroj Google a zpětně pak zobrazuje výsledky hledání na vašem webu.

Google Maps API byla spuštěna v roce 2005. Tento nástroj umožňuje webovým vývojářům vkládat na své stránky mapy společnosti Google, a dále je upravovat a přidávat vlastní tematické prvky jako jsou bodové či liniové vrstvy nebo vizualizace firemních dat. Toto cestou je umožněna i prezentace satelitních snímků nebo využití služby Street View. Aplikace je poskytována zdarma pro nekomerční použití. Jedinou podmínkou bezplatného využití je nepřekročení limitu 25 000 přístupů k mapě denně.

Cílem práce je vytvoření webové interaktivní mapy, využívající podkladová referenční data poskytnutá při využití Google Maps API společností Google. Další složkou této webové mapy je tematická nadstavba v podobě bodové vrstvy, vytvářené uživateli výsledné mapové aplikace. Funkčnost odpovídající přidání této tematické nadstavby, možnost úpravy vizualizace podkladových dat a možnost práce s ostatními kompozičními prvky je umožněna pouze v rámci Google Maps API. Mezi další výhody využití této API patří kvalitně zpracovaná a snadno dostupná dokumentace. Pro samotnou práci s Google API není nutná předchozí zkušenost s řešenou problematikou, nebo hlubší znalost JavaScriptu.

### **3.3 Definice funkčních a mimofunkčních požadavků**

Prvním krokem při samotné tvorbě aplikace je definice funkčních a mimofunkčních požadavků. Touto definicí je vymezena kompletní funkčnost aplikace a je možná samotná realizace celého řešení. Funkčnost celé webové mapy částečně vychází z využití Google Maps API, a možností, které toto prostředí poskytuje. Pro rámec definice obou skupin požadavků je použit stejný doporučený model jako pro hodnocení vybraných webovým mapových zdrojů v kapitole 2 (2.2 Kritéria hodnocení webových mapových zdrojů, Obr. 3).

### **3.3.1 Funkční požadavky**

V rámci požadavků funkčních jsou zohledněny oblasti uživatelského rozhraní (User Interface), manipulace s daty a průběhu procesů (Data & Process Handling), kompozice aplikace (Composition Aspects) a částečně i oblast bezpečnost (Security). Pro zpracování v rámci bakalářské práce jsou definovány následující základní funkční požadavky na výslednou mapovou aplikaci.

- aplikace je implementována v prostředí WWW
- aplikace umožní prohlížení mapových podkladů
- aplikace umožní ovládání pohybu mapového pole pomocí myši
- aplikace umožní přiblížení a oddálení mapového pole
- aplikace umožní vycentrování mapového pole na vybraný prvek
- aplikace umožní odečítání zeměpisných souřadnic
- aplikace umožní zobrazení informací a fotografií k vybraným bodům
- aplikace umožní registraci nového uživatele
- aplikace umožní přihlášení a odhlášení uživatele
- aplikace umožní vhodné zabezpečení uživatelských dat
- aplikace umožní nahrávání souborů ve formě obrázku na server
- aplikace umožní prohlížení nahraných souborů
- aplikace umožní přidat k nahraným souborům další informace
- aplikace umožní zobrazení legendy
- aplikace umožní zobrazení měřítko mapy

### **3.3.2 Mimofunkční požadavky**

V rámci požadavků mimofunkčních je jako hlavní zohledněna oblast nákladů (Costs).

- aplikace využívá volně dostupná podkladová data
- tvorba aplikace probíhá pomocí volně dostupného SW (software)

## **3.4 Realizace řešení**

V následující kapitole jsou popsány nejdůležitější kroky realizace výsledné webové mapové aplikace. Konečné provedení sestává z několika dílčích celků, bez

jejichž realizace by výsledná aplikace v mnoha ohledech nesplňovala navrženou funkčnost. Komplexnost problému si vyžádala práci nejen se samotnou mapou ale i řešení otázky administrace uživatelů, webové bezpečnosti nebo správy souborů nahrávaných uživateli na server.

Pro realizaci celého řešení je využit editor PSPad verze 4.5.8. Jedná se o volně šiřitelný (freeware) editor pro MS Windows ([20]). Jedná se o textový editor, který poskytuje vhodné prostředí pro tvorbu dokumentů CSS, PHP i JavaScript a tím umožňuje pohodlnou tvorbu webových stránek.

Celá aplikace využívá server Apache HTTP server project. Jedná se o softwarový webový server s otevřeným kódem pro GNU/Linux, BDS (Berkeley Software Distribution), Solaris, Mac OS (operační systém) X, Microsoft Windows a další platformy. V současné době dodává prohlížečům na celém světě většinu webových stránek ([15]). V rámci práce je využita verze 2.2.

Pro řešení databáze je využita MySQL databázová platforma Oracle Database. Platforma je vyvinuta společností Oracle Corporation. Pro potřeby práce je použita aktuální verze Oracle Database 11g.

Pro správu databáze je využit php MyAdmin. Jedná se o volně dostupné softwarové řešení pro správu MySQL databáze pro webové aplikace ([19]). Tento software umožňuje základní správu databáze prostřednictvím jednoduchého uživatelského rozhraní.

Samotná realizace aplikace pak probíhá pomocí technologií uvedených v předcházející kapitole. Základ webových stránek je vytvořen pomocí HTML. Na straně serveru je použit jazyk PHP, který dynamicky zobrazuje HTML stránky uživatelům. Vizuální stránka aplikace je pak zpracována pomocí stylů CSS. Funkčnost celé aplikace pak umožňuje využití JavaScriptu.

### **3.4.1 Struktura navržené aplikace**

Výsledná webová mapová aplikace má za úkol umožnit uživatelům tvorbu a správu vlastních tematických vrstev. Tento fakt ovšem vyžaduje určité omezení přístupu uživatelů k mapě. V opačném případě, tedy za předpokladu že by k aplikaci bylo možné přistupovat volně, by snadno došlo k znehodnocení informační hodnoty tematické mapy. Aplikace tedy není pouhou interaktivní webovou mapou, ale obsahuje i kompletní prostředí pro správu uživatelů. Uživatelům je umožněna snadná registrace, prostřednictvím registračního formuláře, a prostřednictvím PHP je ošetřena možnost přihlášení a odhlášení z aplikace.

Toto rozdělení jasně určuje části aplikace přístupné všem (veřejná část aplikace) a části přístupné pouze zaregistrovanému uživateli. Veřejná část zahrnuje náhled na mapové pole (dále jen hlavní strana) s vybranými body zájmu, legendu a náhled na poslední přidané fotografie u vybraných bodů. Dále

tato část aplikace zahrnuje informativní stránku popisující obsah projektu. Poslední veřejnou částí je registrační formulář. Všechny tyto části jsou dostupné prostřednictvím hlavního menu zobrazeného na hlavní straně aplikace. Poslední důležitou složkou je formulář pro přihlášení, který je umístěn na hlavní straně.

Druhá část aplikace je tedy zpřístupněna pouze uživateli, který se zaregistruje a následně přihlásí. Po přihlášení je přesměrován na hlavní stranu aplikace, kde najde opět mapové pole, legendu, formulář pro odhlášení a změněné hlavní menu. Mapové pole poskytuje uživateli prostřednictvím kliknutí přidat nový vlastní bod. Menu nově obsahuje položku „Moje fotky“, která nahradila možnost „Registrace“. V nabídce „Moje fotky“ má pak uživatel možnost spravovat vlastní nahrané soubory, nebo přidávat soubory nové. Uživatel má dále možnost zobrazit si vlastní soubory na mapovém poli.

Takto navržené uživatelské rozhraní je propojeno s kompletní databází. V databázi jsou uchovávány jak registrační údaje uživatelů, tak informace pro správu fotografických míst a souborů, které byli uživateli nahrány.

Celá aplikace pak funguje na základním principu klient – server. Jedná se o základní síťovou architekturu, ve které lze na jedné straně komunikace vymezit klienta a na druhé server. V případě zpracovávané aplikace probíhá na straně serveru komunikace s databází, a na straně klienta pak bude probíhat prezentace dat formou webové API. Komunikace pak probíhá prostřednictvím počítačové sítě, kdy klient (nebo více klientů) zasílá požadavky na server, který je zpracovává. Klient je většinou představován aplikací poskytující uživateli grafické rozhraní. Princip architektury je zobrazen na níže uvedeném obrázku.



**Obr. 4:** Princip architektury klient – server ([30])

### **3.4.2 Realizace mapové části**

Základním prvkem výsledné aplikace je pochopitelně mapová část. Tato část projektu byla kompletně realizována v prostředí Google Maps API, pomocí nástrojů v API již implementovaných. Tyto nástroje umožňují spravovat samotné mapové pole po vizuální i technické stránce a stejně tak umožňují práci se



základními kompozičními prvky mapy jako je měřítko nebo legenda. Propojením prvků obsažených v API s navrženou databází a funkčním PHP je vytvořena kompletní podoba webové mapové aplikace.

Hlavním řešeným problémem je tvorba vlastní bodové vrstvy, a zápis jednotlivých bodů do databáze. Body jsou dále propojeny s prohlížením fotografií, k nim přiřazených. Je tedy vytvořena komunikace mezi JavaScriptovým kódem na straně uživatele a procesy v rámci PHP na straně serveru.

Prostřednictvím Google Maps API je dále zpracována vizuální stránka mapy, i ostatních kompozičních prvků aplikace, které jsou v rámci API předpřipraveny. Kompletní řešení využívá rozsáhlou dokumentaci pro Google Maps API [3].

Práce s Google Maps API začíná vygenerováním tzv. API key. Tento klíč je dostupný po vytvoření účtu Google, a je unikátní pro dané použití API. Pro práci s API, vložíme do HTML hlavičky souboru JavaScriptový kód, který se nachází na adrese uvedené v rámci HTML tagu „<script></script>“. Spuštěním tohoto scriptu na začátku každé stránky proběhne načtení Google mapy. Po tomto kroku pak stačí vytvořit v rámci projektu JavaScriptový dokument v němž budeme pracovat s celým JavaScriptem a propojit ho s celou aplikací. Zápis těchto kroků je zobrazen na obrázku (Obr. 5).

```
<!DOCTYPE html>
<html>
  <head>

    <meta charset="UTF-8">
    <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key={API_KEY}&sensor=false"></script>
    <script type="text/javascript" src="skripty.js"></script>

  </head>
```

**Obr. 5:** Připojení JavaScriptu a Google Maps API k aplikaci

Jako základ pro práci slouží inicializace a nastavení základních parametrů mapy. Tyto dva kroky lze realizovat jednoduchým zápisem funkcí připravených v rámci Google Maps API. O samotnou inicializaci mapy se stará funkce „initializeMap“. Je vytvořena proměnná „map“, která nese informaci o nastavení mapy, a místě jejího vykreslení. Základní nastavení mapy je uloženo do proměnné „mapOptions“.

```
function initializeMap() {

  var mapOptions = {
    center: new google.maps.LatLng(49.19506, 16.606837), zoom: 13, minZoom: 13,
    disableDefaultUI: false,
    mapTypeControlOptions: {
      mapTypeIds: [google.maps.MapTypeId.ROADMAP, 'map_style']
    }
  }

  map = new google.maps.Map(document.getElementById("map-canvas"), mapOptions);
}
```

**Obr. 6:** Inicializace mapy a nastavení základních parametrů

### 3.4.2.1 Technologická stránka

V rámci této části práce je řešena problematika přidávání vlastních bodů do mapového pole, jejich vykreslení, a následné propojení bodů s galerií fotografií. Přihlášenému uživateli je v rámci aplikace umožněno vybrat na mapě libovolný bod, a na jeho souřadnicích vytvořit nové fotografické místo. Tento bod bude na mapě po správném uložení všech dat zaznamenán, je vázán ke konkrétnímu místu a je možné k němu nahrávat fotografie. Tyto fotografie, lze pak při zobrazení detailu daného bodu prohlížet.

Pro vytvoření provázání mezi mapovým polem (tedy Google Maps API), PHP dokumenty a databází jsou využity technologie jQuery, AJAX (Asynchronous JavaScript and XML) a JSON (JavaScript Object Notation). JQuery je JavaScriptová knihovna, umožňující manipulaci s HTML dokumenty, jako je například přidávání událostí, efektů, AJAX nebo manipulace s CSS dokumenty, prostřednictvím uživatelsky příjemné API. Jedná se o volně dostupný software.

AJAX značí obecný termín pro technologie k vývoji webových aplikací, pomocí nichž je možné měnit část obsahu webové stránky, bez nutnosti znovunačtení celého obsahu. Změny jsou uskutečněny pomocí asynchronního zpracování webových stránek prostřednictvím JavaScriptové knihovny. AJAX je obsažen v knihovně jQuery,

JSON představuje speciální datový formát, který je nezávislý na počítačové platformě, a je tedy ideálním prostředkem k výměně dat. Data jsou v něm zpracována ve formě polí nebo objektů a vstupním prvkem může být jakákoliv datová struktura, jejíž složitost je v podstatě neomezená. Výstupem je pak forma řetězce.

Prvním krokem pro vytvoření bodu je načtení jeho souřadnic představujících zeměpisnou délku a zeměpisnou šířku, a odeslání koordinát do formuláře pro vytvoření nového bodu. Odečtení souřadnic bodu je provázáno s kliknutím na mapové pole, a je realizováno pomocí nové události („event“). Souřadnice kliknutí se odečtou a jsou uloženy do proměnných. Obsah proměnných je následně vypsán na předem určené místo na stránce a uživateli je nabídnuta možnost na tyto souřadnice přidat vlastní bod. Pokud zvolí možnost přidat bod, bude přesměrován k vyplnění formuláře. Formulář obsahuje doplňující informace k bodu, a především prostor pro upload vlastní fotografie na server. Řešení tohoto kroku je zobrazeno níže (Obr. 7).

```

google.maps.event.addListener(map, 'click', function(event) {
    var myLatLng = event.latLng;
    var lat = myLatLng.lat();
    var lng = myLatLng.lng();
    var pos = "Souřadnice bodu: " + "\n<br>\nlat: " + myLatLng.lat() + "\n<br>\nlng: " + myLatLng.lng() + "\n<br>\n";
    document.getElementById("gallery_content").innerHTML=pos;
    var pos_p =
        "Pro tento bod: " + "\n<br>\nX: "
        + myLatLng.lat() + "\n<br>\nY: " + myLatLng.lng()
        + "\n<br>\n není zatím přidána žádná fotografie, chcete ji přidat?
        <a href = \"http://127.0.0.1/monika/fotky.php?zemdelka=" + lng + "&zemsirka=" + lat + "\">Přidat</a>";

    document.getElementById("gallery_content").innerHTML=pos_p;
});

```

**Obr. 7:** Odečtení souřadnic kliknutí na mapě, jejich vypsání s možností přejít k přidání bodu

Druhým krokem po získání souřadnic, je vykreslení bodů do mapového pole. Výběr bodů, které jsou aplikací zobrazeny, se řídí několika základními pravidly.

Prvním faktorem je, zda je uživatel, který si prohlíží mapu přihlášený nebo odhlášený. Pro každý přidávaný bod, vytvořený jakýmkoliv uživatelem, je vybráno, zda se bude jednat o bod (fotografii) veřejnou nebo soukromou. Tento výběr je zapsán do databáze a určuje uživatele, kterým bude umožněno ho na mapě vidět. Přihlášeným uživatelům budou k dispozici body, u nichž zvolili nastavení „soukromý“ i body jejichž nastavení je „veřejný“. Uživatel, který není do aplikace přihlášen, pak budou zobrazeny pouze body veřejné. Kontrola přihlášení je provedena pomocí kontroly aktivní PHP session (viz kapitola 3. 4. 3. 2 Přihlášení a odhlášení uživatele).

Dalším rozhodujícím faktorem pro výběr zobrazených bodů je jejich poloha v rámci celé mapy. Mapa je do výsledné aplikace vložena tak, že ani při maximálním oddálení, neposkytuje náhled na celou mapovou plochu. Je tedy nutné vybrat pro zobrazení body, které svými souřadnicemi spadají do momentálního zobrazeného výřezu mapy.

Souřadnice jeho krajních bodů závisí např. na úrovni přiblížení, nebo na tom jak uživatele mapu posouvá. Jsou tedy proměnné v závislosti na činnosti uživatele. Tyto změny jsou ošetřeny v rámci JavaScriptu, kdy dochází po změně i k překreslení bodů, podle aktuálního výřezu. Body aktuálního výřezu jsou uchovávány v poli. Toto pole je při každé manipulaci s mapou vyprázdněno a jsou do něj nahrány nové body. Tímto opatřením sice dochází k mírnému zpomalení vykreslování, ale zároveň je zabráněno duplikování bodů. Obnovení výběru bodů při přiblížení nebo změně centra mapy (tedy při posunutí), je zobrazeno na následujícím obrázku (Obr. 8).

```

google.maps.event.addListener(lmap, 'zoom_changed', function(){
    displayMarkers(lmap);
});

google.maps.event.addListener(lmap, 'center_changed', function(){
    displayMarkers(lmap);
});

```

**Obr. 8:** Funkce pro překreslení bodů při změně přiblížení nebo změně centra mapy

Při každé podobné manipulaci s mapou je tedy nutné získat souřadnice levého dolního a pravého horního rohu („\$sw\_lng“, „\$sw\_lat“, „\$ne\_lng“, „\$ne\_lat“), jimiž je vymezen aktuální čtyřúhelník zobrazené mapy, tyto souřadnice jsou získány z JavaScriptu. Následně jsou z databáze vybrány body, které svými XY souřadnicemi spadají do tohoto výřezu (v rámci PHP) a jsou vykresleny opět prostřednictvím JavaScriptu. Výsledné body splňující podmínky nastavení soukromí a spadají do vybrané oblasti, jsou uloženy do pole. Tato kontrola je opakována v cyklu, dojde tedy k vykreslení všech bodů splňujících dané podmínky. Tento postup vyžaduje kombinaci řešení v JavaScriptu a PHP. Průběh lze vidět na následujících obrázcích (Obr. 9, Obr. 10.).

```

$sw_lng = $_POST ['southwest_lng'];
$sw_lat = $_POST ['southwest_lat'];
$ne_lng = $_POST ['northeast_lng'];
$ne_lat = $_POST ['northeast_lat'];

if($sw_lat <= $ne_lat){
    $lat_min = $sw_lat;
    $lat_max = $ne_lat;
}else{
    $lat_min = $ne_lat;
    $lat_max = $sw_lat;
}

if($sw_lng <= $ne_lng){
    $lng_min = $sw_lng;
    $lng_max = $ne_lng;
}else{
    $lng_min = $ne_lng;
    $lng_max = $sw_lng;
}

$sql = "SELECT * from saved_points WHERE ( (zem_delka >= $lat_min and zem_delka <= $lat_max)
AND (zem_sirka >= $lng_min and zem_sirka <= $lng_max) )";

```

**Obr. 9:** Kontrola polohy bodů a výběr vhodných bodů z databáze

```

function displayMarkers(map) {
    var bounds = map.getBounds();
    var northeast = bounds.getNorthEast();
    var southwest = bounds.getSouthWest();

    var southwest_lng = southwest.lng();
    var southwest_lat = southwest.lat();
    var northeast_lng = northeast.lng();
    var northeast_lat = northeast.lat();

    $.ajax({
        url: "markers.php",
        type: "POST",
        data: {
            "southwest_lng" : southwest_lng,
            "southwest_lat" : southwest_lat,
            "northeast_lng" : northeast_lng,
            "northeast_lat" : northeast_lat
        }
    }).done(function(response){
        console.log(response);
    });

    response = JSON.parse(response);
    console.log(response);

    for (index in response.markers) {
        console.log(index);
        addMarker(response.markers[index]);
    };

    function addMarker(data) {
        var marker = new google.maps.Marker({
            position: new google.maps.LatLng(data.lat, data.lng),
            map: map,
            title: data.name
        });
        console.log(marker);
        marker.setMap (map);
    }
}

```

**Obr. 10:** Vykreslení bodů

Součástí vykreslování bodů je i přidělení vybrané ikony k danému bodu podle jeho kategorie zapsané v databázi. Informace o kategorii fotografie je tedy také předána do JavaScriptu pomocí formátu JSON. V JavaScriptu jsou načteny obrázky pro jednotlivé symboly kategorií a jsou podle kategorie vykresleny u každého bodu.

Druhým krokem po samotném vykreslení bodů, je jejich propojení s fotografiemi, a zobrazení fotografií každého bodu. Využito je opět asynchronní vykreslování obsahu stránek.

Po vybrání konkrétního bodu mapy, a kliknutí se do panelu na pravé straně stránky zobrazí fotky, které jsou v databázi uloženy k danému bodu. V rámci PHP probíhá porovnání na základě souřadnic podobné PHP zápisu pro výběr odpovídajících bodů. V rámci JavaScriptu probíhá opět předání dat z PHP pomocí formátu JSON. Základní zápis pro provedení tohoto kroku je znázorněn na obrázcích níže (Obr. 11, Obr. 12).

```
function displayPhotos(lat, lng) {
    $.ajax({
        url: "fotos.php",
        type: "POST",
        data: {
            "zem_sirka" : lat,
            "zem_delka" : lng
        }
    }).done(function(response){
        console.log(response);

        response = JSON.parse(response);

        for (index in response.photos) {
            document.getElementById("gallery_content").innerHTML="";
            $("#gallery_content").append(
                "<img src='./uploadedfotos/" + response.photos[index]['file_name'] + "' width='150' height='120' />"
            );
        };
    });
}
```

**Obr. 11:** Funkce pro výběr fotografií konkrétního bodu a jejich zobrazení

```
function addMarker(data) {
    var marker = new google.maps.Marker({
        position: new google.maps.LatLng(data.lat, data.lng),
        map: map,
        title: data.name
    });

    google.maps.event.addListener(marker, "click", function() {
        displayPhotos (data.lat, data.lng);
    });

    marker.setMap (map);
    markersArray.push(marker);
}
```

**Obr. 12:** Zápis pro spuštění zobrazovací funkce po kliknutí na konkrétní bod

### 3.4.2.2 Vizuální stránka

Pro zobrazení vybraného území je nutné nastavit pomocí funkcí Google Maps API základní parametry zobrazované mapy, jako jsou její hranice, nastavení

výchozí polohy jejího středu a podobně. Toto nastavení určuje vzhled mapy, zobrazené na hlavní stránce, před tím než je s ní pracováno, a také určuje rámeček, ve kterém lze s mapou pracovat.

Aplikace pro správu fotografických míst je v rámci bakalářské práce navržena pro oblast v okolí Brna. Důležitým je tedy stanovení hranic zobrazované mapy, tak aby tomuto území odpovídala. Mapové pole je definováno souřadnicemi rohů. Nastavení je uloženo do proměnné „allowedBounds“ pro další práci.

```
var allowedBounds = new google.maps.LatLngBounds(  
    new google.maps.LatLng(48.18, 16.25),  
    new google.maps.LatLng(49.60, 16.60)  
);
```

**Obr. 13:** Stanovení požadovaných hranic mapového pole

Dalším prvkem definujícím zobrazené mapové pole je centrum mapy, tedy bod, kolem něhož se vykreslí mapa, a který bude jejím středem. Opět je jednoduše definován pomocí dvou souřadnic. Nastavena je i úroveň možného přiblížení nebo oddálení mapy a jsou vybrány ovládací prvky uživatelského rozhraní. Celé nastavení lze uložit do proměnné („mapOptions“). Nastavení prvků lze měnit dle potřeby, a možnosti zobrazení jednotlivých elementů lze nalézt v dokumentaci Google Maps API.

Nastavení zvolené pro aplikaci je zobrazeno na následujícím obrázku. Jako podkladová mapa je zvolena standardní verze Google mapy, a je zachováno základní nastavení uživatelského rozhraní („disableDefaultUI“ je tedy nastaveno na „false“). V rámci tohoto nastavení se zobrazí klasická verze ovládacích prvků pro přiblížení a oddálení mapy a také ovládací panel pro pohyb s mapou. Posledním zobrazeným prvkem je měřítko, která je zapnuto nastavením „scaleControl“ na hodnotu „true“.

```
var mapOptions = {  
    center: new google.maps.LatLng(49.19506, 16.606837), zoom: 13, minZoom: 13,  
    disableDefaultUI: false,  
    scaleControl: true,  
    mapTypeControlOptions: {  
        mapTypeIds: [MAP_TYPE_ID]  
    },  
    mapTypeId: MAP_TYPE_ID  
}
```

**Obr. 14:** Nastavení centra mapy a základních prvků uživatelského rozhraní

### 3.4.3 Správa uživatelů

Jak bylo zmíněno v předchozí kapitole, je pro správný chod aplikace nutné odlišit možnosti uživatelů zaregistrovaných a uživatelů bez registrace. Proto je v rámci aplikace pro správu fotografických míst zajištěna možnost uživatelské registrace a následného přihlášení a odhlášení. V následujících podkapitolách bude nastíněno řešení této části mapové aplikace.

#### 3.4.3.1 Registrace nového uživatele

První částí administrace uživatelů je registrace nového uživatele. Pro umožnění této funkce je základem vytvoření funkčního registračního formuláře v HTML. Výchozím zdrojem pro tvorbu základních HTML prvků byla pro potřeby bakalářské práce online dostupná dokumentace na stránkách w3schools.com [32], ze které jsou čerpány základní znalosti o implementaci HTML využité v následujícím textu.

Jednoduchý formulář lze v HTML vytvořit zapsáním párových tagů `<form>` a `</form>`. Tyto tagy značí začátek a konec formuláře, a mezi ně je pak možné vkládat jednotlivé položky formuláře. Ty lze zapsat pomocí tagů `<input>`.

Důležitým atributem formuláře je „action“. Tento atribut určuje, kam budou data zadaná do formuláře odeslána. Atribut „method“ pak určuje způsob předání těchto dat, a nabývá hodnot „GET“ nebo „POST“. Na obrázcích níže najdeme v obou případech využití metody „POST“. Tato metoda zajistí, že data přenášená mezi stránkami nejsou viditelná a přenos probíhá uvnitř serveru. Při využití metody „GET“ by se hodnoty přenášených atributů a proměnných zobrazovaly přímo v URL adrese a není to proto vhodný způsob pro přenos dat z registračních nebo přihlašovacích formulářů.

Na následujícím obrázku je pak ukázka kódu pro formulář registrace uživatelů. Formulář obsahuje jednotlivé položky, konkrétně uživatelské jméno, heslo, pole pro ověření hesla, email a pole pro ověření uživatele ve formě jednoduchého statického výpočtu. Položky jsou označeny tagem „`<input>`“. Každá položka formuláře pak nese několik důležitých atributů, které hrají roli v typu vstupních dat a v jejich následném přenosu. Důležitý je atribut „type“, který udává typ formulářového pole, který HTML vygeneruje, a udává tedy, jaké vstupní hodnoty bude uživatel moci zadat či vybrat.

Ve výsledném registračním formuláři nese tento atribut hodnoty „text“, „password“ a „submit“. V případě hodnoty „text“ je vygenerováno vstupní textové pole, hodnota „password“ pak představuje vstupní textové pole se skrytými znaky. „Submit“ představuje potvrzovací tlačítko (Obr. 15). Každé položce je pak nastaveno jméno („name“), které slouží pro identifikaci dat při jejich přenosu a ověření po odeslání formuláře ke kontrole a uložení.

```

<div class="registration_form">
  <table cellpadding="2" cellspacing="5">
    <form action="registrace.php" method="post">
      <tr><td>Uživatelské jméno</td>
      <td><input type="text" name="nickname" size = "30"></td><td>Pod uživatelským jménem bude možné přidávat Vaše fotografie.</td></tr>
      <tr><td>Heslo</td>
      <td><input type="password" name="password" size = "30"></td><td>Heslo by mělo obsahovat alespoň 6 znaků.</td></tr>
      <tr><td>Ověření hesla</td>
      <td><input type="password" name="passwordconfirm" size = "30"></td></tr>
      <tr><td>E-mail</td>
      <td><input type="text" name="email" size = "30"></td><td>Váš e-mail bude sloužit jako přihlašovací údaj.</td></tr>
      <tr><td>Ověření: <img src = "captcha.png" =</td>
      <td><input type="text" name="check_number" size = "30"></td><td>Vyplňte prosím ověřovací pole.</td></tr>
      <tr><td></td>
      <td><input type="submit" name="send" value="Odeslat"></td></tr>
      <td>Po úspěšné registraci budete přihlášení a přesměrováni na hlavní stránku</td></tr>
    </form>
  </table>
</div>

```

**Obr. 15:** Zápis vytvořeného registračního formuláře v PHP

Uživatelské jméno	<input type="text"/>	Pod uživatelským jménem bude možné přidávat Vaše fotografie.
Heslo	<input type="password"/>	Heslo by mělo obsahovat alespoň 6 znaků.
Ověření hesla	<input type="password"/>	
E-mail	<input type="text"/>	Váš e-mail bude sloužit jako přihlašovací údaj.
Ověření: 6+3 =	<input type="text"/>	Vyplňte prosím ověřovací pole.
<input type="button" value="Odeslat"/>		
Po úspěšné registraci budete přihlášení a přesměrováni na hlavní stránku		

**Obr. 16:** Výsledný registrační formulář

Dalším krokem registrace je uložení dat zadaných uživatelem do formuláře. Tato část se realizuje pomocí dotazu MySQL. Data jsou uložena do databáze „fotomapdatabase“, kde je vytvořena tabulka „registred\_users“.

Prvním krokem pro využití databáze je její připojení. Připojení k databázi, je krokem, který se v řešení aplikace opakuje. Proto je tento krok, vložen do PHP funkce. Stejným způsobem je řešeno odpojení od databáze a výběr tabulky, se kterou pracujeme. Pokud je vytvořena vhodná funkce, stačí ji v případě potřeby pouze zavolat, a není tedy nutné opisovat znovu celý úsek kódu. Vytvoření a ukončení spojení s databází je zapsáno na obrázku níže (Obr. 17).

```

function database_connect () {
    $connection = mysql_connect ("localhost","monika","heslo");
    if (!$connection) {
        die ("Nepodařilo se připojit k databázi" . mysql_error ());
    }

    return $connection;
}

function database_disconnect ($connection) {
    mysql_close ($connection);
}

function database_table_choose ($table, $connection) {
    mysql_select_DB ($table, $connection);
}

```

**Obr. 17:** Připojení a odpojení MySQL databáze, výběr tabulky



Výsledný zápis pro uložení dat z formuláře je zobrazen na následujícím obrázku. Do tabulky jsou do sloupců „uzivatelske\_jmeno“, „heslo“ a „email“, vloženy hodnoty proměnných „\$uzivatelske\_jmeno“, „\$hasheduniquepassword“ a „\$email“. Tyto proměnné nesou ošetřené řetězce, které byly vyplněny do příslušných formulářových polí. Sloupec „id“ je nastaven na auto\_increment, jeho hodnota se tedy při každém novém zápisu automaticky zvýší, a proto mu není předána žádná nová hodnota.

```
$sql = "INSERT INTO registred_users (`id`, `uzivatelske_jmeno`, `heslo`, `email`) VALUES
(NULL, '$uzivateslke_jmeno', '$hasheduniquepassword', '$email')";
$result = mysql_query ($sql);
```

**Obr. 18:** Zápis nového uživatele do databáze

Pokud tento krok proběhne úspěšně je uživatel registrován. Po úspěšné registraci je pak uživatel přesměrován přímo na hlavní stranu a zároveň je automaticky přihlášen. Celé řešení procesu registrace je zobrazeno v přílohách (Příloha 3).

### 3.4.3.2 Přihlášení a odhlášení uživatele

Druhou částí administrace je umožnění uživateli se přihlásit (log in) a odhlásit (log out) z prostředí aplikace. Tento krok umožní rozdělit obsah aplikace na část pro zaregistrované a přihlášené uživatele a na část veřejnou. Je tedy nutné zabezpečit neveřejný obsah aplikace proti přístupu jakéhokoliv neautentizovaného návštěvníka. Tento krok nám umožní využití takzvaných PHP session proměnných (relací, sezení, session na úrovni připojení).

*Základní charakteristikou protokolu http je jeho bezstavovost, což znamená, že neudrží spojení mezi dvěma transakcemi. HTTP neposkytuje možnost, jak zjistit, jestli určité požadavky přišly od stejného uživatele ([28], str. 402). Právě session proměnné poskytují možnost udržet informaci o stavu aplikace a o tom jak je s ní pracováno a umožní uchovat a přenést data mezi dvěma přístupy.*

Jednoduše lze říci, že díky proměnné session je možné na každé stránce aplikace zkontrolovat, je-li uživatel přihlášený, a podle výsledku kontroly zobrazovat přístupný obsah nebo uživatele přesměrovat na jinou stránku.

V rámci aplikace pro správu fotograficky zajímavých míst, je každému uživateli, který se úspěšně registruje nebo přihlásí pomocí formulářů, založena vlastní session, která přenáší informaci o oprávněném přístupu. Tato session zaniká pomocí dalšího skriptu při jeho odhlášení. Oba zápisy použití session lze

vidět na následujících obrázcích. Na prvním obrázku lze vidět definici nové session, které je přiřazen název „user\_email“ a odpovídá proměnné „\$uniqueusername“, což je řetězec předaný z přihlašovacího formuláře. Po úspěšném zaregistrování nové session je uživateli umožněn přístup do celé aplikace.

```
session_start ();  
$_SESSION ["user_email"] = $uniqueusername;
```

**Obr. 19:** Zápis pro nastavení proměnné session

Na obrázku níže je zapsáno ukončení stejné session. Nejdříve je pomocí funkce „unset ()“ odebrána hodnota a následně je celá session ukončena. Pokud proběhne tento proces, uživateli bude po dalším přesměrování zobrazen obsah pro nepřihlášené uživatele.

```
session_start ();  
unset($_SESSION["user_email"]);  
session_destroy ();
```

**Obr. 20:** Zápis pro ukončení session

Na každé stránce aplikace je dále zabudována funkce, která ověřuje, zda je nějaká ověřená session nastavena. Pokud ano je vykreslen obsah pro přihlášeného uživatele. Pokud ověření session neproběhne úspěšně, dojde k přesměrování na hlavní stranu. Pokud by neprobíhalo toto ověření, mohl by jakýkoliv uživatel jednoduše zkopírovat adresu chráněného obsahu v prohlížeči, a tím získat přístup bez autentizace. Zápis funkce pro ověření session je zobrazen na dalším obrázku. Je vytvořena proměnná „\$session\_check“ a je nastavena na hodnotu „false“. Následně je ověřeno, jestli je správně nastavena hodnota globální proměnné „\$\_SESSION“ pomocí funkce „isset“. Dále je zkontrolováno, zda hodnota session není prázdná. Pokud obě kontroly proběhnou úspěšně hodnota proměnné „\$session\_check“ se nastaví na „true“.

```

$session_check = false;

function check_session () {
    session_start ();
    if (!isset ($_SESSION ["user_email"])) {
        if (empty ($_SESSION["user_email"])) {
            $session_check = false;
        }else{
            $session_check = true;
        }
    }else{
        $session_check = true;
    }

    return ($session_check);
}

```

**Obr. 21:** Zápis pro ověření platnosti a obsahu session

Vstup do části pro přihlášené uživatele je umožněn jak prostřednictvím přihlašovacího formuláře na hlavní straně, tak je umožněn automaticky po úspěšné registraci. V případě využití přihlašovacího formuláře je provedena kontrola dat pomocí dotazu na MySQL databázi. Je nutné zkontrolovat jednak samotný zadaný řetězec, který nesmí obsahovat nepovolené znaky. Dále je ověřeno, zda se uživatel nachází v databázi zaregistrovaných uživatelů, a zda zadal oba přihlašovací údaje správně. Kompletní řešení přihlášení a odhlášení je součástí příloh (Příloha 4).

### 3.4.4 Bezpečnost

Výstupem této bakalářské práce je webová aplikace. Sdílení dat v prostředí webu přináší uživatelům mnohé výhody, ale také obnáší přenos velkého množství dat, která je třeba správným způsobem zabezpečit. Touto problematikou se zabývá **webová bezpečnost**. Jedná se o rozsáhlý a neustále se rozvíjející obor, zabývající se zajištěním bezpečnosti webových serverů ve všech ohledech. „Nejčastěji řešené bezpečnostní hrozby pak mohou zahrnovat ([28], str. 277):

- Únik důvěrných dat
- Ztrátu nebo zničení dat
- Modifikaci dat
- Vyřazení z provozu
- Chyby v softwaru
- Odmítání“

V rámci řešené problematiky je bezpečnost řešena jen částečně. Jedná se o velmi komplexní problém, jehož řešení je časově i technologicky náročné. Přesto je nutné v realizované webové aplikaci zajistit alespoň základní zabezpečení přenosu dat. Při případném dalším rozvoji aplikace pro fotograficky zajímavá místa by bylo vhodné řešit tento problém celistvěji.

Jak bylo představeno v předcházející podkapitole, v rámci aplikace je zprovozněna kompletní administrace uživatelů. Tento fakt obnáší uchovávání údajů o uživateli v databázi. Uchováváno je jeho uživatelské jméno, email a heslo. Tato data, hlavně pak unikátní heslo každého uživatele, je pak nutné při komunikaci mezi serverem a klientem určitým způsobem zabezpečit. K tomuto zabezpečení je možné využít takzvané **šifrovací algoritmy** (šifrovací funkce, hashovací algoritmy).

„Šifrovací algoritmus je matematická funkce transformující informace do zdánlivě nesmyslného řetězce“ ([28], str. 285). Vstupem do takového algoritmu je obyčejně textový řetězec (např. uživatelské heslo) a výstupem je vygenerovaná zašifrovaná informace v podobě řetězce znaků. Tento řetězec má podle vybraného algoritmu jiné uspořádání a délku. Příklady výsledného šifrování vybraných algoritmů můžeme vidět na obrázku níže. V prvním sloupci lze vidět název algoritmu, ve druhém pak počet znaků, do nichž je vstupní řetězec zašifrován, ve třetím sloupci pak můžeme vidět ukázkou výsledné šifry, která je v některých případech zkrácená. Výrazem, který by pro příklad zašifrován byl řetězec „hello“. Některé z těchto šifrovacích algoritmů – například algoritmus md5 - byly v minulosti prolomeny, a byl rozřešen postup jejich šifrování. Proto není vhodné je z hlediska bezpečnosti dále používat.

md2	32	a9046c73e00331af68917d3804f70655
md4	32	866437cb7a794bce2b727acc0362ee27
md5	32	5d41402abc4b2a76b9719d911017c592
sha1	40	aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d
sha256	64	2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e730
sha384	96	59e1748777448c69de6b800d7a33bbfb9ff1b463e44354c3553
sha512	128	9b71d224bd62f3785d96d46ad3ea3d73319bfbcb2890caadae2d
ripemd128	32	789d569f08ed7055e94b4289a4195012
ripemd160	40	108f07b8382412612c048d07d13f814118445acd
ripemd256	64	cc1d2594aece0a064b7aed75a57283d9490fd5705ed3d66bf9a

**Obr. 22:** Příklad zašifrovaného kódu při použití vybraných šifrovacích algoritmů [18]

Pro registraci a log in uživatelů v rámci mapové aplikace je využit šifrovací algoritmus sha256. Tento algoritmus převede vstupní řetězec na 64 znaků dlouhou šifru, která je unikátní pro daný šifrovaný řetězec. Obecnou syntaxi lze vidět na obrázku níže. Dojde k jednoduchému zavolání funkce „hash()“, která slouží k vygenerování šifry, a které jsou předány dva parametry, a to zvolený šifrovací algoritmus a řetězec, který má být transformován. Tento řetězec je v našem případě uložen do konkrétní proměnné „\$variable“.

```
<?php
hash ('sha256', $variable);
?>
```

**Obr. 23:** Zápis šifrování prostřednictvím algoritmu sha256 v PHP.

V rámci vytvořené webové mapové aplikace jsou tímto způsobem ošetřeny data zadaná uživatelem při registraci. Po správném vyplnění registračního formuláře a potvrzením se provede zašifrování řetězce zadaného jako uživatelské heslo a jeho uložení do nové proměnné \$hasheduniquepassword, kterou je nahrazena vstupní proměnná „\$heslo“, jak je možné vidět na obrázku níže (Obr. 24). Tato nová proměnná je pak uložena do databáze na pozici uživatelského hesla.

```
$hasheduniquepassword = hash("sha256",$heslo);
$sql = "INSERT INTO registred_users (`id`, `uzivatelske_jmeno`, `heslo`, `email`)
VALUES (NULL, '$uzivateslke_jmeno', '$hasheduniquepassword', '$email)";
```

**Obr. 24:** Ukázka zašifrování hesla a uložení zašifrovaného řetězce do databáze.

Tento postup zajistí, že do databáze se budou ukládat již zašifrovaná data, a samotné heslo nebude tak není možné dohledat v původní formě. Výsledný zápis do databáze po úspěšné registraci lze vidět na obrázku níže (Obr. 25). Stejným způsobem je ošetřen i log in, kde se po zadání přihlašovacích údajů, provede u hesla stejný šifrovací postup a následně je srovnám se záznamem v databázi. Pokud obě hodnoty souhlasí, je uživateli umožněno přihlášení.

id	uzivatelske_jmeno	heslo	email
18	Jan_24	1f3ce40415a2081fa3eee75fc39fff8e56c22270d1a978a724...	jan@centrum.cz
19	Anna235	2b50fd5a1cba316c013317ef68f3ef0a600d40187f85653287...	anna235@gmail.com

**Obr 25:** Výsledný zápis do databáze při užití šifrovacího algoritmu sha256.

V rámci registrace uživatelů je dalším bezpečnostním opatřením užití verifikace zadaných dat prostřednictvím takzvaných **regulárních výrazů**. „Regulární výraz je v podstatě vzorek, sada znaků, který popisuje vlastnost hledaného řetězce „([28], str. 170). Pomocí těchto výrazů, lze omezit textový řetězec, který lze do formuláře zadat. V rámci regulárního výrazu můžeme použít speciální znaky, které nám umožní popsat konkrétní vlastnosti hledaného nebo porovnávaného řetězce.

V PHP je pro práci s regulárními výrazy vytvořeno několik funkcí. Pro potřeby aplikace vystačí funkce „ereg()“, která slouží k porovnávání řetězců se zadaným vzorem. Na následujícím obrázku je ukázka využití regulárních výrazů,

pro kontrolu tvaru emailové adresy. Zadaný regulární výraz umožňuje zadat pouze řetězec sestávající z libovolného počtu znaků z množiny [-a-zA-Z0-9.\_+%], která představuje libovolná malá a velká písmena, čísla, tečky, podtržítka, znaménko plus a znak procenta. Za tímto řetězcem následuje povinný symbol zavináče. Za tímto symbolem je pak místo pro další řetězec představující doménu, za nímž následuje tečka. Poslední řetězec znaků představuje místo pro zápis domény prvního řádu, která je omezena na formu velkých a malých písmen a délka řetězce musí být mezi dvěma a čtyřmi znaky.

Druhým parametrem funkce „ereg()“ je proměnná, kterou chceme s daným regulárním výrazem porovnat. V tomto případě srovnáváme proměnnou \$email, představující emailovou adresu zadanou uživatelem do registračního formuláře.

Celý výraz navíc začíná znakem (!). Tento symbol vyjadřuje v PHP negaci, takže pokud se zadaný výraz, uložený v proměnné „\$email“ nebude shodovat se vzorem, který je definován regulárním výrazem bude uživatel na chybu upozorněn vypsáním textu pomocí funkce „echo ()“. Nebude tak splněna podmínka pro úspěšné vyplnění pole registračního formuláře a proměnná „\$validity“, která představuje celkovou správnost vyplnění formuláře, bude nastavena na hodnotu „false“. V této situaci nebude uživateli umožněno se úspěšně zaregistrovat, a bude nutné jím zadaný řetězec opravit podle vzoru.

```
if (!ereg("^[-a-zA-Z0-9._+%]+@[[-a-zA-Z0-9.]+\.[a-zA-Z]{2,4}$", $email)) {  
    echo "Neplatná e-mailová adresa <br>";  
    $validity=false;  
}
```

**Obr 26:** Příklad využití regulárního výrazu pro ověření e-mailové adresy.

Stejným způsobem jsou ošetřeny ostatní položky formuláře. V případě uživatelského jména a hesla není povoleno vkládat vybrané speciální znaky, jejichž zápis by umožnil vložit do formulářového pole škodlivý skript, který by zamezil správnému chodu aplikace. Proto lze i validaci dat zahrnout do bezpečnosti aplikace. Vzhledem k využití emailu jako přihlašovacího jména je postup ověření tvaru řetězce opakováno i v rámci přihlašovacího formuláře.

Užitím šifrovacího algoritmu a regulárních výrazů je tedy zajištěna základní bezpečnost aplikace. Řešení celkové bezpečnosti není v rámci bakalářské práce uvažováno a může být předmětem budoucího vývoje aplikace.

### 3.4.5 Nahrávání souborů

Důležitou součástí funkčnosti celé aplikace pro správu fotograficky zajímavých míst je možnost uživatele nahrávat na server vlastní fotografie a místo jejich pořízení pak zobrazit nad mapovými podklady. Soubory lze v rámci

webových aplikací uchovávat různým způsobem. Časté je ukládání dat do databáze, kdy je pro konkrétní typy dat (jako jsou obrázky nebo zvukové soubory), možné vytvořit sloupec s vhodným datovým typem a realizovat tak přímý zápis souborů.

V rámci realizované aplikace je zvoleno ukládání souborů přímo do složky serveru. Prvním krokem je vytvoření HTML formuláře pro upload souboru. Tento formulář je od jednoduchých formulářů několika způsoby odlišen. Najdeme zde parametr „enctype = „multipart/form-data“, který předává serveru informaci o odeslání souboru zároveň s formulářovými daty. Druhým důležitým prvkem je vstupní pole typu „file“. Zápis pro vytvoření HTML formuláře je zobrazen na následující obrázku (Obr. 27).

```
<div class="upload_form">
  <table cellpadding="2" cellspacing="5">
    <form enctype = "multipart/form-data" action="fotky.php" method="post">
      <tr><td>Vyberte soubor, který chcete nahrát:</td><td></td><td><input name="uploadfile" type="file"></td></tr>
      <tr><td><input type="submit" value="Nahrát"></td></tr>
    </form>
  </table>
</div>
```

**Obr. 27:** Vytvoření formuláře pro upload souborů prostřednictvím HTML

Zápisem tohoto kódu je uživateli prostřednictvím PHP následně vygenerován jednoduchý formulář (Obr. 28). Po výběru požadovaného souboru proběhne v několika krocích jeho zpracování. Po potvrzení možnosti „Nahrát“ je soubor přesunut do dočasného úložiště na serveru. Pokud z tohoto dočasného úložiště není přesunut dále dříve, než je ukončen běh stávajícího skriptu je tento soubor ztracen. Na dočasném úložišti proběhne například kontrola nahraných dat a následně přesunutí souboru na trvalé serverové úložiště.



**Obr. 28:** Výsledný formulář pro upload souboru

Nutným krokem je rozsáhlejší kontrola nahrávaných dat. Kontrola probíhá pomocí kontroly splnění nebo nesplnění definovaných podmínek. Základní údaje o souboru jsou uloženy do proměnných, které lze porovnávat s vytvořenými vzory. Důležitá je kontrola formátu souboru, jeho velikosti a kontrola toho, zda nebyl formulář odeslán prázdný. Pokud tyto podmínky nejsou splněny, uživatel je vyzván k opravě.

Pro příklad kontroly lze uvést kontrolu formátu souboru. Pro nahrávání fotografií jsou zvoleny povolené přípony .png, .jpeg a .jpg, jakožto nejčastější a nejvhodnější formáty pro ukládání obrázků. Tyto přípony jsou obsaženy v kontrolním poli v proměnné „\$allowed\_file\_types“. Ze vstupních dat dostáváme

hodnotu proměnné „\$upload\_file\_type“, která nese informaci o typu souboru, nahraného uživatelem. Tato hodnota je následně porovnána s polem. Opět je definována proměnná „\$datavalidity“ určující úspěch nebo neúspěch kontroly dat. Při nesplnění podmínky je proměnná nastavena na hodnotu „false“ a uživateli je vypsána chybová hláška. Zápis pro tuto kontrolu, lze vidět na následujícím obrázku (Obr. 29).

```
$allowed_file_types = array ("image/jpeg", "image/pjpeg", "image/jpg", "image/png");

if (!in_array($upload_file_type, $allowed_file_types)) {
    echo "Nahráný soubor není ve správném formátu";
    $datavalidity = false;
    exit;
}
```

**Obr. 29:** Zápis pro kontrolu přípony nahrávaného souboru

Formulář pro upload souboru je součástí formuláře pro zadání základních charakteristik fotografie (jejího názvu, kategorie, popisku atd.). Celý tento formulář je po úspěšném vyplnění a nahrání souborů odeslán a jsou z něj do databáze uložena data pro zápis nového bodu. Celé řešení nahrávání fotografie je součástí příloh (Příloha 5).

## 3.5 Návrh vizuální stránky aplikace

Posledním krokem při realizaci webové aplikace je návrh kompletního vizuálního designu. Pro tento krok realizace je využito několik volně dostupných aplikací, které v tomto ohledu částečně usnadní práci. Vzhledem k faktu, že vizuální stránka mapového pole byla představena v předcházející kapitole, je v následujícím textu popsána pouze vizuální stránka samotného uživatelského rozhraní.

### 3.5.1 Návrh uživatelského rozhraní

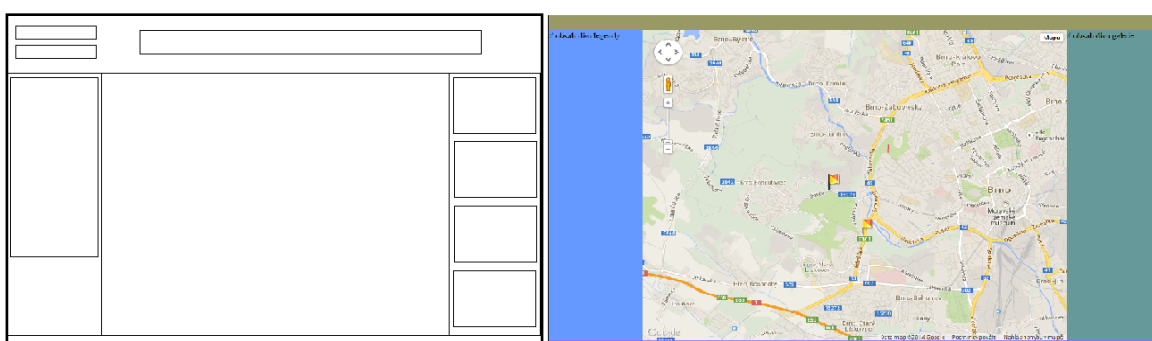
Základní rozvržení hlavní strany vychází z velké části z požadované funkčnosti. Nejpodstatnější část tedy logicky představuje mapové pole, které tvoří základní stavební prvek aplikace a je zároveň hlavním funkčním prvkem celého řešení, který bude předpokládaným uživatelem využíván nejčastěji. Mapové pole je tedy pochopitelně na hlavní straně umístěno uprostřed a zabírá podstatnou část plochy. Tento hlavní prvek je uzavřen v rámci z dalších bloků.

V levém panelu je umístěna legenda mapy, v pravém panelu je pak vytvořen prostor pro otevření galerie fotografií nebo zobrazení fotografie. Tyto dva bloky najdeme pouze na hlavní straně aplikace spolu s mapovou částí. Ostatní části aplikace neobsahují mapové pole a je tedy nepotřebné vykreslovat uživatelům mapovou legendu nebo galerii konkrétního bodu.



Blok obsahující legendu je umístěn co nejtěsněji u mapového pole a je částečně zvýrazněn. Tento krok umožní uživateli snadnější orientaci mezi legendou a mapovým polem, a celkově mu usnadní práci s aplikací a interpretaci mapového pole.

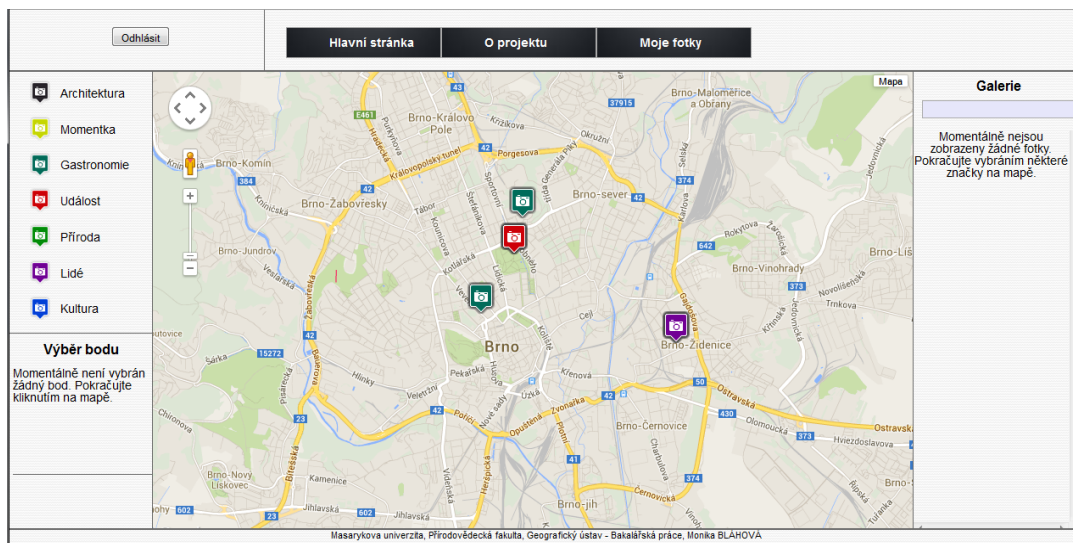
Další dva bloky tedy horní blok nesoucí horizontálně uspořádané menu a přihlašovací nebo odhlašovací formulář, a blok patičky, se opakují na všech stránkách navržené aplikace, pouze s mírně upraveným obsahem v ohledu na to zda je uživatel přihlášen nebo ne. Původní návrh rozvržení hlavní strany a následně základní rozvržení je možné vidět na následujícím obrázku (Obr. 30). Pro vytvoření prvotního layoutu stránky byla využita aplikace Layzilla [10]. Jedná se o automatický online generátor CSS a HTML layoutů, prostřednictvím kterého lze jednoduše vytvořit rozvržení stránky a využít vygenerovaný kód.



**Obr. 30:** Návrh rozvržení hlavní strany a prvotní realizace rozložení

Jelikož se ve všech případech jedná o prvky, které bude aplikace vykreslovat opakovaně, je v rámci HTML kódu vhodné vytvořit PHP funkce, vykreslující jednotlivé úseky (jako například hlavičku, patičku) a volat v rámci aplikace tyto funkce. Celá aplikace sestává na základě funkčnosti z několika stránek, návrh rozvržení celé webové aplikace je součástí příloh (Příloha 6).

Tento prvotní návrh tedy umožnil rozdělení prvků a úloh k jednotlivým elementům stránky. K výslednému designu bylo využito CSS stylů, pomocí nichž byly jednotlivé prvky aplikace nastaveny do požadované vizuální podoby. Pro každý element, byla vytvořena třída v CSS dokumentu, který je součástí aplikace. Příkladem může být finální podoba hlavní stránky, zobrazená na následujícím obrázku (Obr. 31), se všemi požadovanými prvky, která je nastýlována právě pomocí CSS. Kompletní výslednou vizuální stránku aplikace a jejich složek je možné najít v přílohách (Příloha 7).



**Obr. 31:** Výsledná podoba hlavní strany aplikace

U prvků stránky je nutné vyřešit jejich uspořádání při zmenšení okna prohlížeče, nebo při malém rozlišení zobrazovacího zařízení. Tento krok je vyřešen jednoduchým vložením celého obsahu stránky (veškerých HTML tagů <div> a jejich obsahu) do jednoho bloku, kterému jsou nastaveny minimální hodnoty šířky a výšky. Pokud je okno zmenšeno pod tyto hodnoty, zůstane obsah stránky stejný a uživateli se pro prohlížení celého okna zobrazí posuvníky. Toto řešení částečně komplikuje práci uživatele při malém rozlišení, ovšem jednoduše zachovává veškeré grafické prvky stránky.

V rámci tohoto bloku, který obsahuje veškeré prvky stránky, je aplikaci globálně určeno i nastavení dílčích prvků, jako je například font, velikost nebo barva písma. V případě potřeby, jako například u formulářů, jsou pak tyto hodnoty mírně upraveny.

V rámci zobrazení fotografií jednoho bodu, je při vyšší počtu fotografií přidáno stránkování, které umožní prohlížení celé galerie fotografií bez dlouhého posouvání obsahu. Posuvník je přidán jen v rámci prohlížení jedné stránky galerie, která je nastavena pro zobrazení 5 fotografií.

Důležitou součástí při dotváření vizuální stránky celého projektu je i sjednocení vzhledu v jednotlivých prohlížečích. V rámci realizace byl brán zřetel na zobrazení v prohlížečích Mozilla Firefox, Google Chrome a Internet Explorer. Výsledná vizuální stránka aplikace je podobná ve všech zmíněných webových prohlížečích. Porovnání náhledu na hlavní stranu je znázorněno na obrázku níže (Obr. 32).



**Obr. 32:** Porovnání zobrazení hlavní strany v testovaných prohlížečích (z leva Internet Explorer, Google Chrome, Mozilla Firefox)

### 3.5.2 Návrh znakové sady

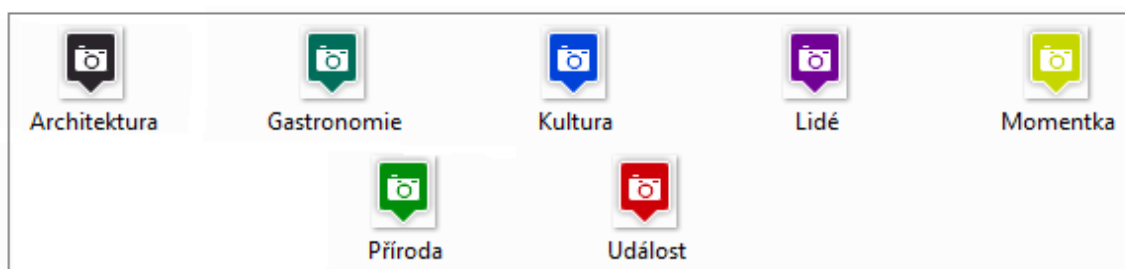
Výsledná webová mapová aplikace má za úkol zobrazovat body zájmu, uspořádané do několika kategorií. Proto byl pro každou kategorii fotografických míst vybrán jeden odlišitelný symbol.

Pro výběr symbolů byla využita online databáze mapových znaků Map Icons Collection [13]. Tato databáze poskytuje širokou nabídku znaků, a umožňuje zvolit styl vybraných symbolů. Jedná se o aplikaci podporovanou firmou Google, v rámci které je možné získat bezplatně mapové symboly mimo jiné pro využití v rámci Google Maps API. Databáze nabízí obrovské množství mapových symbolů, pro většinu objektů, jevů nebo oborů lidské činnosti, které jsou vyjádřitelné pomocí map.

Pro potřeby výsledné webové mapové aplikace byly tedy vybrány mapové symboly právě z této databáze. Vytvoření vlastního znakového klíče a rozvoj kartografické stránky aplikace je vhodným tématem dalšího rozvoje. Vzhledem k potřebě zaznamenat do mapového pole polohu bodu o konkrétních souřadnicích a následně vyjádřit jednu charakteristiku daného jevu, jsou vybrány bodové (figurální) symboly. Jako vodící znak je zobrazen jednoduchý geometrický znak vybraný z databáze. Definiční bod tohoto znaku se pak nachází ve středu základny vybraného znaku. Znaky nevyjadřují kvantitativní charakteristiky, je tedy zachována stejná velikost všech symbolů.

Vyjádřena je pak pomocí těchto symbolů kvalitativní charakteristika daného jevu, konkrétně kategorie fotografického místa. Ta je odlišena pomocí barevných variant. Barvy jsou vybrány s ohledem na barevnost mapových podkladů tak, aby byly jednotlivé symboly jasně rozeznatelné nad vybranými mapovými podklady (Obr. 33).

Všechny body jsou pak přehledně zobrazeny legendou dostupnou na hlavní straně aplikace. V legendě jsou zobrazeny všechny možné zobrazené symboly, které mohou být uživatelem do mapy přidány. Je tedy statickým prvkem stránky.



**Obr. 33:** Mapové symboly využité v rámci aplikace [13]

Tyto ikony jsou k jednotlivým bodům přidávány v závislosti na uživatelem vybrané kategorii. Ta je uchovávána v databázi, spolu s ostatními informacemi o fotografii a přidaném bodu.

Zbývající mapové symboly a celková prezentace dat vychází z využití podkladové mapy společnosti Google. Celková symbolika tedy odpovídá právě této mapě. V rámci funkčnosti Google Maps API je možné měnit nebo upravit styl podkladových map. Pro potřeby bakalářské práce byla ovšem ponechána klasická verze mapy, která umožní uživateli snadnou orientaci v podkladové mapě. Zachovány jsou i veškeré původní mapové popisy. Jedinou změnu představuje vypnutí zobrazení turistických bodů zájmu, ke kterým jsou standardně v rámci Google map přiřazena informační okna. Pokud by se zobrazovala i tato okna, mohla by překrývat výslednou bodovou vrstvu, a proto bylo zobrazení těchto bodů v nastavení mapy vypnuto.

### 3.6 Webová implementace

Závěrečným úkolem je implementace aplikace do prostředí WWW. Důležitou podmínkou při výběru hostingu byla podpora co možná nejbližší verze PHP a MySQL. Pro zajištění této funkčnosti byl vybrán bezplatný webový hosting moxo.cz. Výběr tohoto hostingu přesto vyžadoval na několika místech úpravu zdrojového kódu tak aby, odpovídal možnostem, které moxo.cz pro neplacený hosting poskytuje. Bylo třeba upravit některé části vytvořeného PHP kódu, tak aby odpovídal verzi PHP 5.4, která je serverem podporována. Příkladem může být nahrazení funkce „ereg ()“, sloužící pro definici regulárních výrazů funkcí „preg\_match ()“. Žádná z těchto úprav neovlivní funkčnost aplikace, jedná se pouze o úpravy na úrovni zápisu. Výsledná aplikace je dostupná na adrese <http://fotomap.moxo.cz>.

## 4 ZÁVĚR

Hlavním cílem této bakalářské práce bylo navrhnout a vytvořit a webovou mapovou aplikaci pro tvorbu a správu fotograficky zajímavých míst s využitím možností Google Maps API. Využití mapové aplikace od společnosti Google, se ukázalo jako ideální řešení pro rychlý a snadný import interaktivních mapových podkladů do jakékoliv webové aplikace. Vzhledem k tempu rozvoje by se pro podobná řešení jevila tato API jako ideální i v případě budoucího rozvoje tohoto projektu.

Matice požadavků na aplikaci, která je v rámci práce definována, a tvoří výchozí bod pro samotnou realizaci, může představovat základní rámec pro tvorbu obdobných projektů a aplikací. Celkově se při realizaci podařilo implementovat veškerou funkčnost navrženou právě v rámci této matice. Výsledná aplikace tedy pracuje podobně jako webové mapové aplikace, které byly hodnoceny v kapitole 2. Výsledná aplikace pro správu fotograficky zajímavých míst tedy obsahuje funkční část administrace uživatelů a jejich souborů, i funkční mapovou část, která obsahuje základní ovládací prvky a umožňuje správu bodové vrstvy fotograficky zajímavých míst.

Při realizaci celého projektu se otevřelo mnoho dílčích otázek a problémů, jejichž řešení může být otázkou dalšího rozvoje aplikace, nebo otázkou nového samostatného projektu. Mezi hlavní otázky dalšího rozvoje aplikace patří například optimalizace přidávání a manipulace s body zájmu, která je v rámci realizace navržena pouze pro základní funkčnost. Rozvinuta by jistě měla být i funkčnost aplikace, například o export nebo tisk mapy. Dalším budoucím tématem by mohl být také například rozvoj celé aplikace pro mobilní zařízení, nebo napojení na sociální sítě, kde by bylo možné jednotlivé fotografie mezi uživateli sdílet s návazností na mapové podklady. Tyto problémy jsou ovšem momentálně nad rámec práce.

Rozšíření realizace by mohlo být otázkou navazující práce diplomové, kde by bylo vhodné řešit i otázku funkčnosti vizuální stránky aplikace, která je v rámci této části řešení z větší části opomenuta. Stejně tak je opomenuta i důležitá součást funkčnosti spojená s mapovým polem. Řešení by mělo zahrnovat například generalizaci mapových objektů při změnách měřítko uživatelem. V současné podobě aplikace jsou tyto složky funkčnosti opomenuty.

Další oblastí, která by mohla být řešena v budoucnu, je samotný zdrojový kód. Ten by mohl být celkově optimalizován a lépe uspořádán, pro usnadnění chodu celé aplikace. Tento krok ovšem vyžaduje hlubší teoretické nastudování využitých programovacích jazyků.

## 5 SEZNAM POUŽITÝCH ZDROJŮ

- [1] BROWN, A., KRAAK, M. J.: *Web Cartography – developments and prospects*. 1. vyd., Taylor and Francis, London, 2001. 213 str. ISBN 0-7484-0869-X.
- [2] CASTAGNETTO, J., RAWAT, H., SCGUMANN, S., SCOLLO, C., VELIATH, D.: *Programujeme PHP profesionálně*. 1. vyd., Computer Press, Praha, 2001. 656 str. ISBN 80-7226-310-2
- [3] *Google developers: Code samples* [online]. 2014 [cit. 2. 4. 2014]. Dostupné z WWW: <<https://developers.google.com/maps/documentation/javascript/examples>>
- [4] *Jak psát web: Úvod do formulářů* [online]. 2014 [cit. 2. 5. 2014]. Dostupné z WWW: <<http://www.jakpsatweb.cz/formulare.html>>
- [5] *jQuery: API documentation* [online]. 2014 [cit. 30. 4. 2014]. Dostupné z WWW: <<http://api.jquery.com/>>
- [6] *JSON: Úvod do JSON* [online]. 2014 [cit. 2. 4. 2014]. Dostupné z WWW: <<http://www.json.org/json-cz.html>>
- [7] KOMÁRKOVÁ, J. *Kvalita webových geografických informačních systémů*. Pardubice: Univerzita Pardubice, 2008. 128 s. ISBN 978-80-7395-056-9.
- [8] KOMÁRKOVÁ, J., SEDLÁK, P., HUB, M. SLAVÍKOVÁ, V. Utilization of Heuristics for Usability Evaluation of Web-based GIS Applications: Využití heuristik pro hodnocení použitelnosti webových gis aplikací. In: *Geografie pro život ve 21. Století: Sborník příspěvků z XXII. Sjezdu České geografické společnosti pořádaného Ostravskou univerzitou v Ostravě 31. Srpna – 3. Září 2010*. Ostrava: Ostravská univerzita v Ostravě, 2010, str. 321-324. ISBN 978-80-7368-903-2.
- [9] KOSEK, J.: *HTML Tvorba dokonalých webových stránek, Podrobný průvodce.*, 1. vyd., Grada Publishing, Praha, 1998. 296 str. ISBN 80-7169-608-0
- [10] *Layzilla: HTML layout generation* [online]. 2014 [cit. 2. 4. 2014]. Dostupné z WWW: <<http://www.layzilla.com/>>
- [11] LEHOCKÝ, Z. – BOBEK, T. *JavaScript – 1. Lekce*. [online]. 2014 [cit. 13. 3. 2014]. Dostupné z: WWW: <<http://programujte.com/clanek/2005123003-javascript-1-lekce>>
- [12] MACH, J.: *PHP pro úplné začátečníky*. 1. vyd., Computer Press, Praha, 2002. 128. str. ISBN: 80-7226-633-0

- [13] *Map Icons Collection* [online]. 2014 [cit. 2. 4. 2014]. Dostupné z WWW: <[http://mapicons.nicolasmollet.com/markers/media/photo/?custom\\_color=544a4c](http://mapicons.nicolasmollet.com/markers/media/photo/?custom_color=544a4c)>
- [14] *Mozilla developer network: JavaScript Guide* [online]. 2014 [cit. 2. 4. 2014]. Dostupné z WWW: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>>
- [15] *Netcraft: April 2014 web server survey* [online]. 2014 [cit. 2. 4. 2014]. Dostupné z WWW: <<http://news.netcraft.com/archives/category/web-server-survey/>>
- [16] NOGUERAS-ISO, J., ZARAZAGA-SORIA, F. J., MURO-MEDRANO, P. R. 2005. *Geographic Information Metadata for Spatial Data Infrastructures*. Springer: Berlin Heidelberg. 263 pages. ISBN 3-540-24464-6.
- [17] PETERSON, M. P. et al.: *Maps and the internet*. 1. vyd., Elsevier Press, Oxford, Amsterdam, 2003. 451 s. ISBN 0-780080-442013.
- [18] *PHP Documentation: Hash Functions* [online]. 2014 [cit. 15. 3. 2014]. Dostupné z WWW: <<http://www.php.net/manual/en/function.hash.php>>
- [19] *phpMyAdmin* [online]. 2014 [cit. 12. 3. 2014]. Dostupné z WWW: <[http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)>
- [20] *PSPad Freeware editor* [online]. 2014 [cit. 10. 3. 2014]. Dostupné z WWW: <<http://www.pspad.com/cz/>>
- [21] ŘEZNÍK, T. 2012. *Návrh webových map*. Masarykova univerzita: Studijní materiály předmětu PŘF:Z8188 Webová kartografie – úvod [online]. 2014 [cit. 19. ledna 2014]. Dostupné z: WWW: <[https://is.muni.cz/auth/el/1431/podzim2012/Z8188/um/Webkart\\_12\\_navrh\\_map.pdf](https://is.muni.cz/auth/el/1431/podzim2012/Z8188/um/Webkart_12_navrh_map.pdf)>
- [22] ŘEZNÍK, T. 2012. *Emergency Support System: Management of Geographic Information for Command and Control Systems*. [Habilitation work]. Masaryk University/University of Defense. 191 s.
- [23] ŠKULÉTY, R.: *JavaScript – programujeme internetové aplikace*, 1. vyd., Computer Press, Praha, 2001. 208 str. ISBN: 80-7226-457-5
- [24] VOŽENÍLEK, V.: *Aplikovaná kartografie I - tematické mapy*. 1. vyd., Univerzita Palackého v Olomouci, Olomouc, 1999. 168 str. ISBN: 80-7067-971-9
- [25] VOŽENÍLEK V., KAŇOK, J., a kol.: *Metody tematické kartografie: Vizualizace prostorových jevů*, 1. vyd, Vydavatelství UP, Olomouc, 2011, 216 str. ISBN: 978-80-244-2790-4

- [26] *Výukový web HTML, CSS, PHP & MySQL pro začátečníky* [online]. 2014 [cit. 20. 4. 2014]. Dostupné z WWW: <<http://tvorba-webu.kx.cz/php/priklady.php>>
- [27] WACHOWICZ, M., CUI, V., VULLINGS, W., BULENS, J. The effects of web mapping applications on user satisfaction: an empirical study. In: *International perspectives on maps and the Internet*. 2007, s. 397 – 416.
- [28] WELLING, L., THOMSON, L.: *PHP a MySQL rozvoj webových aplikací.*, SoftPress s.r.o., Praha, 2002. 718 str. ISBN 80-86497-20-8
- [29] Wikipedie: *HyperText Markup Language* [online]. 2014 [cit. 13. 3. 2014]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/HTML>>
- [30] *Wikipedia: Client – server model* [online]. 2014 [cit. 2. 4. 2014]. Dostupné z: WWW: <[http://en.wikipedia.org/wiki/Client\\_server](http://en.wikipedia.org/wiki/Client_server)>
- [31] *W3schools: AJAX tutorial* [online]. 2014 [cit. 2. 4. 2014]. Dostupné z WWW: <<http://www.w3schools.com/ajax/default.ASP>>
- [32] *W3schools: HTML 4 and HTML 5 tutorial* [online]. 2014 [cit. 2. 4. 2014]. Dostupné z WWW: <<http://www.w3schools.com/html/DEFAULT.asp>>
- [33] *W3schools: jQuery tutorial* [online]. 2014 [cit. 1. 5. 2014]. Dostupné z WWW:<<http://www.w3schools.com/Jquery/default.asp>>
- [34] W3C: *Kaskádové styly – domovská stránka* [online] 2014. [cit. 21. 3. 2014]. Dostupný z: WWW: <<http://www.w3.org/Style/CSS/Overview.cs.html>>



## **6 SEZNAM ZKRATEK**

WWW – Word Wide Web (celosvětová síť)

API – Application Programming Interface (rozhraní pro programování aplikací)

SŠ – severní šířka

JŠ – jižní šířka

VD – východní délka

ZD – západní délka

SW - software

URL – Uniform Resource Locator (Jednotný lokátor zdrojů)

GIS – Geographic Information System (Geografický informační systém)

HTTP – Hypertext Transfer Protocol

OS – Operation System (Operační systém)

BSD – Berkeley Software Distribution

AJAX - Asynchronous JavaScript and XML

JSON - JavaScript Object Notation (Javascriptový objektový zápis)

## 7 PŘÍLOHY

### Vázané přílohy

- Příl. 1. Hodnocení funkčních a mimofunkčních požadavků pro vybrané webové mapové zdroje
- Příl. 2. Hodnocení mimofunkčních požadavků pro vybrané webové mapové zdroje
- Příl. 3. Registrace nového uživatele
- Příl. 4. Přihlášení a odhlášení uživatele
- Příl. 5. Nahrávání fotografií
- Příl. 6. Rozvržení aplikace
- Příl. 7. Vizuální řešení aplikace

### Přílohy na CD

- /icons* obrázky pro použité mapové symboly
- /uploadedfotos* připravená složka pro upload fotografií
- /styly.css* dokument CSS, obsahující kompletní styly aplikace
- /skripty.js* kompletní JavaScript dokument pro aplikaci
- /images* obrázky aplikace
- /index.php* PHP dokument pro hlavní stranu aplikace
- /oprojektu.php* PHP dokument pro stránku „O projektu“
- /registrace.php* PHP dokument pro stránku „Registrace“
- /mojefotky.php* PHP dokument pro stránku „Moje fotky“
- /fotky.php* PHP dokument pro přidávání nových fotografií
- /delete\_foto.php* PHP dokument pro mazání záznamů o fotografiích
- /fotos.php* PHP dokument pro výběr aktuálních zobrazených fotografií
- /markers.php* PHP dokument pro výběr aktuálních zobrazených bodů
- /kontrola\_log\_in.php* PHP dokument pro přihlášení uživatele

<i>/logout.php</i>	PHP dokument pro odhlášení uživatele
<i>/funkce_buil_layout.php</i>	PHP dokument pro vykreslení prvků stránky
<i>/funkce_database.php</i>	PHP dokument pro funkce spojené s databází
<i>/funkce_session.php</i>	PHP dokument pro funkce spojené s sessions
<i>/Fotomapdatabase.sql</i>	SQL kód databáze

Příl. 1. Hodnocení funkčních požadavků pro vybrané webové mapové zdroje

**Tab. 1:** Legenda k hodnocení funkčních požadavků pro vybrané mapové služby

ANO	DOSTUPNÉ	NE

**Tab. 2:** Hodnocení funkčních požadavků pro vybrané mapové služby

Uživatelské rozhraní			
	GM	OSM	MCZ
Je k dispozici měření vzdušnou čarou?			
Je k dispozici měření pomocí liniových prvků?			
Lze mapu uložit jako obrázek?			
Je umožněn tisk mapy?			
Je dostupný nástroj pro přiblížení/oddálení?			
Je dostupný nástroj pro posouvání?			
Je dostupný vyhledávací nástroj?			
Manipulace s daty a průběh procesů			
	GM	OSM	MCZ
Lze s mapou pohybovat pomocí myši?			
Lze s mapovou pohybovat pomocí kurzorových šipek?			
Lze mapu přiblížit pomocí dvojkliku?			
Lze přiblížit oblast pomocí kolečka myši?			
Funguje v Google Chrome?			
Funguje v Mozilla Firefox?			
Funguje v Internet Explorer?			
Lze mapu vycentrovat na prvek?			
Je možné nastavit styl zobrazovaných dat?			
Lze ručně nastavit měřítko mapy?			
Kompozice			
	GM	OSM	MCZ
Je k dispozici grafické měřítko?			
Je k dispozici číselné měřítko?			
Je dostupná legenda všech vrstev?			
Zabírá mapové pole podstatnou část plochy?			

Příl. 2. Hodnocení mimofunkčních požadavků pro vybrané webové mapové zdroje

**Tab. 3:** Legenda k hodnocení mimofunkčních požadavků pro vybrané mapové služby

ANO	DOSTUPNÉ	NE

**Tab. 4:** Hodnocení mimofunkčních požadavků pro vybrané mapové služby

Dokumentace			
	GM	OSM	MCZ
Lze zjistit datum pořízení dat?			
Lze zjistit zdroj dat?			
Náklady			
	GM	OSM	MCZ
Jsou mapové podklady volně dostupné?			

### Příl. 3. Registrace nového uživatele

<?php

```
require ("funkce_build_layout.php");
require ("funkce_database.php");
require ("funkce_session.php");

$result = true;

if (isset ($_POST ['send'])) {
    $uzivateslke_jmeno = $_POST ['nickname'];
    $heslo = $_POST ['password'];
    $potvrzeni_hesla = $_POST ['passwordconfirm'];
    $email=$_POST ['email'];
    $check_number = $_POST ['check_number'];

    $connection = database_connect ();
    database_table_choose ("fotomapdatabase",$connection);

    $validity = true;

    if ($uzivateslke_jmeno =='') {
        echo "Zadejte prosím Vaše uživatelské jméno <br>";
        $validity=false;
    }
    if ($heslo =='') {
        echo "Zadejte prosím Vaše heslo <br>";
        $validity=false;
    }
    if ($potvrzeni_hesla =='') {
        echo "Potvrďte prosím Vaše heslo <br>";
        $validity=false;
    }
    if ($heslo!=$potvrzeni_hesla) {
        echo "Hesla nesouhlasí<br>";
        $validity=false;
    }
    if ($email =='') {
        echo "Zadejte prosím Váš e-mail <br>";
        $validity=false;
    }
    if (!ereg ("^[-a-zA-Z0-9._+%]+@[ -a-zA-Z0-9.] +\.[a-zA-Z]{2,4}$", $email)) {
        echo "Neplatná e-mailová adresa <br>";
        $validity=false;
    }
    if (strlen ($heslo) < 6) {
        echo "Heslo musí mít alespoň 6 znaků <br>";
        $validity=false;
    }
    if (ereg ("^[-a-zA-Z0-9._]$", $uzivateslke_jmeno)) {
        echo "Uživatelské jméno obsahuje nepovolené znaky
        <br>";
        $validity=false;
    }
    $q = "SELECT `email` FROM `registred_users` WHERE
    `email`='$email'";
    $check= mysql_query ($q);
```

```

$num_of_emails = mysql_fetch_array($check);
if (!empty($num_of_emails)) {
    echo "Tato emailová adresa je již registrována";
    $validity=false;
}

if ($check_number != 9){
    echo "Chybně vyplněné ověření";
    $validity=false;
}

if ($validity==true) {
    $hasheduniquepassword = hash("sha256",$heslo);
    $sql = "INSERT INTO registred_users (`id`,`uzivatelske_jmeno`,`heslo`,`email`) VALUES (NULL,'$uzivateslke_jmeno','$hasheduniquepassword','$email')";
    $result = mysql_query ($sql);
    if (!($result == true)){
        echo "Registrace nebyla provedena<br>";
    } else{
        session_start ();
        $_SESSION ["user_email"] = 'email';
        header ("Location: mapa.php");
    }
}
database_disconnect ($connection);
}

```

#### Příl. 4. Přihlášení a odhlášení uživatele

```
<?php

    require ("funkce_build_layout.php");
    require ("funkce_database.php");
    require ("funkce_session.php");

    ob_start();

    $connection = database_connect ();

    database_table_choose ('fotomapdatabase', $connection);

    $uniqueusername=$_POST['uniqueusername'];
    $uniquepassword=$_POST['uniquepassword'];

    $validity=true;

    if (!ereg("^[-a-zA-Z0-9._+%]+@[[-a-zA-Z0-9.]+\.[a-zA-Z]{2,4}$",
    $uniqueusername)) {
        echo "Neplatná e-mailová adresa <br>";
        $validity=false;
    }

    if ($validity==true){
        $hasheduniquepassword = hash("sha256",$uniquepassword);
        $sql="SELECT * FROM registred_users WHERE
        email='$uniqueusername' and heslo='$hasheduniquepassword'";
        $result=mysql_query($sql);

        $count=mysql_num_rows($result);
    }else {
        echo "Neplatné heslo nebo email";
        $count = 0;
    }

    if($count==1){
        session_start ();
        $_SESSION ["user_email"] = $uniqueusername;
    }

    header ("Location:mapa.php");

    ob_end_flush ();
?>

<?php

    require ("funkce_build_layout.php");
    require ("funkce_database.php");
    require ("funkce_session.php");

    session_start ();
    unset($_SESSION["user_email"]);
    session_destroy ();

    header ("Location:mapa.php");
?>
```



## Příl. 5. Nahrávání fotografií

<?php

```
require ("funkce_build_layout.php");
require ("funkce_database.php");
require ("funkce_session.php");

$get_coords = true;
$zem_sirka = $_GET ['zemsirka'];
$zem_delka = $_GET ['zemdelka'];

$zem_sirka = number_format($zem_sirka,6);
$zem_delka = number_format($zem_delka,6);

if (!isset ($_GET ['zemsirka']) || !isset ($_GET ['zemdelka'])) {
    if (empty ($_GET ['zemsirka']) || empty ($_GET ['zemdelka'])) {
        $get_coords = false;
    }else{
        $get_coords = true;
    }
}
}else{
    $get_coords = true;
}

if ($get_id == true && $get_coords == false) {
    $set_form = false;
}
if ($get_id == false && $get_coords == true) {
    $set_form = true;
}
if ($get_id == false && $get_coords == false) {
    $set_form = true;
}
if ($get_id == true && $get_coords == true) {
    $set_form = true;
}

if (isset ($_POST ['send'])) {
    $photo_name=$_POST ['name'];
    $photo_description=$_POST ['description'];
    $photo_category=$_POST ['category'];
    $photo_privacy=$_POST ['privacy'];

    $description_length = strlen ($photo_description);
    $photo_description = htmlspecialchars ($photo_description);
    list($width, $height, $type, $attr) =
    getimagesize($_FILES['uploadfile']['tmp_name']);

    $inputvalidity = true;

    if ($description_length > 250) {
        echo "Popis je příliš dlouhý (maximální délka je 250 znaků)";
        $inputvalidity=false;
    }
    if ($photo_name =='') {
        echo "Zadejte prosím název fotografie <br>";
        $inputvalidity=false;
    }
    if ($height < $min_file_height || $height > $max_file_height) {
        echo "Soubor má nevhovující výšku (vhodná výška je mezi 480
        a 1200 px)<br>";
        $inputvalidity=false;
    }
    if ($width < $min_file_width || $width > $max_file_width) {
        echo "Soubor má nevhovující šířku (vhodná výška je mezi 640
        a 1600 px)<br>";
    }
}
```

```

        $inputvalidity=false;
    }
    if ($photo_description == '') {
        echo "Prosím zadejte popis fotografie <br>";
        $inputvalidity=false;
    }
    if ($photo_category == '') {
        echo "Vyberte prosím kategorii <br>";
        $inputvalidity=false;
    }
    if ($photo_privacy == '') {
        echo "Zvolte prosím nastavení soukromí <br>";
        $inputvalidity=false;
    }

    if (ereg ("^[-a-zA-Z0-9._]$", $photo_name)) {
        echo "Název fotografie obsahuje nepovolené znaky <br>";
        $inputvalidity=false;
    }
    if (ereg ("^[-a-zA-Z0-9._!]+$", $photo_description)) {
        echo "Popis fotografie obsahuje nepovolené znaky <br>";
        $inputvalidity=false;
    }
}

if ($inputvalidity == true) {
    if ($_FILES['uploadfile']['error'] > 0) {
        echo "Nebyl vybrán soubor pro upload, nebo se vyskytla chyba<br>";
        fotoform_build_new ($zem_sirka,$zem_delka);
    } else{
        $upload_file_name = $_FILES ['uploadfile']['name'];
        $upload_file_type = $_FILES ['uploadfile']['type'];
        $upload_file_size = $_FILES ['uploadfile']['size'];
        $upload_file_tmp_name = $_FILES ['uploadfile']['tmp_name'];
        $name_reverse = strrev($upload_file_name);
        $name_parts_array = explode(".", $name_reverse);
        $name_add_array = $name_parts_array [0];
        $name_add = strrev ($name_add_array);
        $hashed_upload_file_name = hash ('sha256', $upload_file_name);
        $add_hashed_upload_file_name = "" . ($hashed_upload_file_name) . "."
        . $name_add;

        $max_file_size = 768000;
        $max_file_height = 1200;
        $max_file_width = 1600;
        $min_file_height = 480;
        $min_file_width = 640;

        $allowed_file_types = array
("image/jpeg", "image/pjpeg", "image/jpg");
        $path = ".\\uploadedfotos\\";
        $datavalidity = true;

        $connection = database_connect ();
        database_table_choose ("fotomapdatabase", $connection);

        while(true){
            $checksqli = "SELECT * FROM saved_points WHERE
            file_name='$add_hashed_upload_file_name'";
            $scheckresult = mysql_query($checksqli);
            $scheckcount = mysql_num_rows($scheckresult);

            if ($scheckcount != 0) {
                $sold_name_with_rnd = $hashed_upload_file_name . rand
                (0,999);
                $new_name = hash ('sha256', $sold_name_with_rnd);
            }
        }
    }
}

```

```

        $add_hashed_upload_file_name = $new_name . "." .
        $name_add;
    } else{
        break;
    }
}
database_disconnect ($connection);
$path_name = $path . $add_hashed_upload_file_name;

if ($upload_file_size == 0) {
    echo "Nahráný soubor má nulovou velikost<br>";
    $datavalidity = false;
}

if (!in_array($upload_file_type,$allowed_file_types)) {
    echo "Nahráný soubor není ve správném formátu<br>";
    $datavalidity = false;
}

if ($upload_file_size >= $max_file_size) {
    echo "Nahráný soubor je příliš velký<br>";
    $datavalidity = false;
}

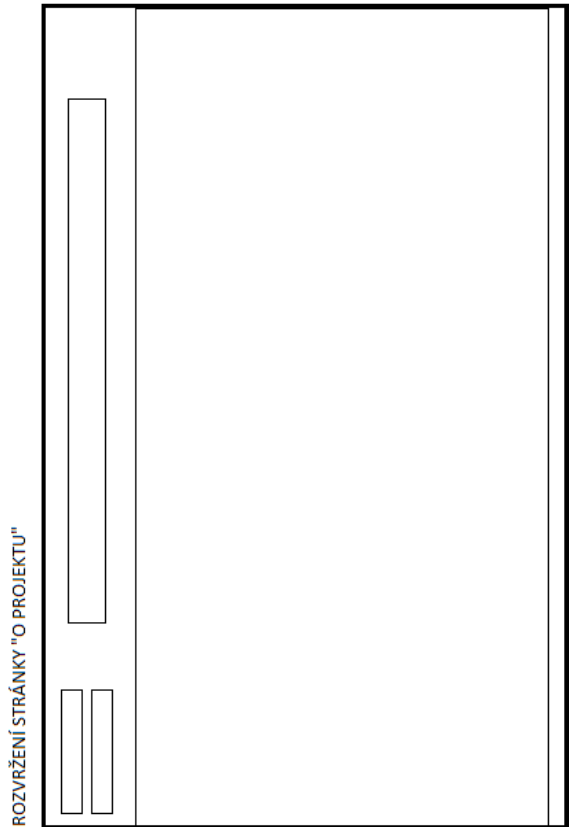
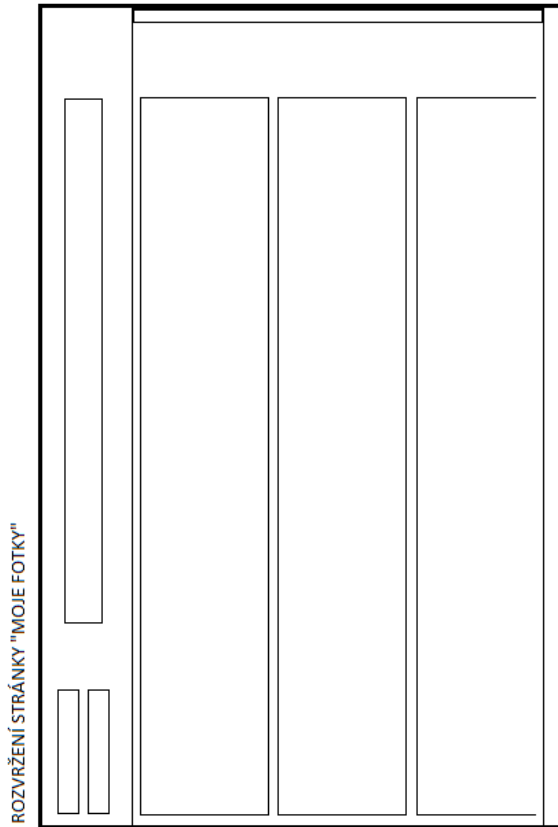
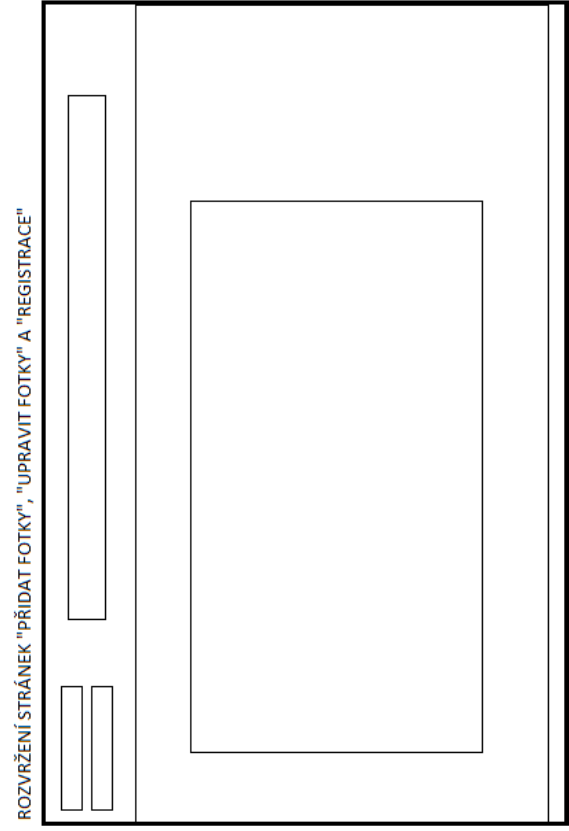
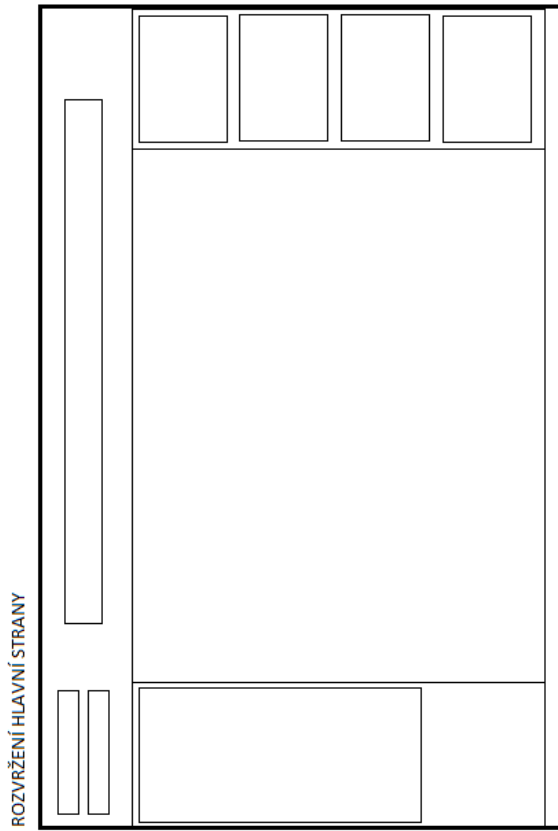
if ($inputvalidity == true && $datavalidity == true) {
    $formresult = true;
} else{
    $formresult = false;
}

if ($formresult == false) {
    echo "Soubor nebyl úspěšně nahrán";
    fotoform_build ($zem_sirka,$zem_delka);
} else{
    if (move_uploaded_file($_FILES['uploadfile']['tmp_name'],
$path_name)) {
        $connection = database_connect ();
        database_table_choose ("fotomapdatabase",$connection);
        $newsq = "INSERT INTO saved_points
        (`id`,`zem_sirka`,`zem_delka`,`foto_user`,`foto_name`,`
        `foto_category`,`foto_privacy`,`filē_name`,`foto_descr
        iption`) VALUES
        (NULL,'$zem_sirka','$zem_delka','$user_id','$photo_nam
        e','$photo_category','$photo_privacy','$add_hashed_upl
        oad_file_name','$photo_description)";
        $result = mysql_query ($newsq);
        database_disconnect ($connection);
        echo "Soubor ". ( $_FILES['uploadfile']['name']). "
        byl úspěšně nahrán";
        header("Location:mojefotky.php");
    } else {
        echo "Při nahrávání souboru nastala chyba";
        fotoform_build_new ($zem_sirka,$zem_delka);
    }
}

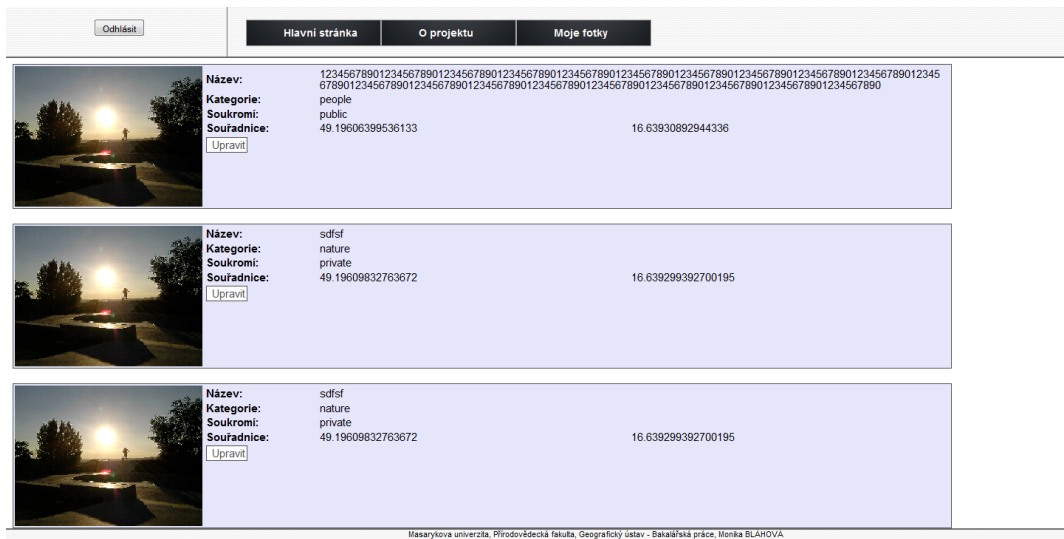
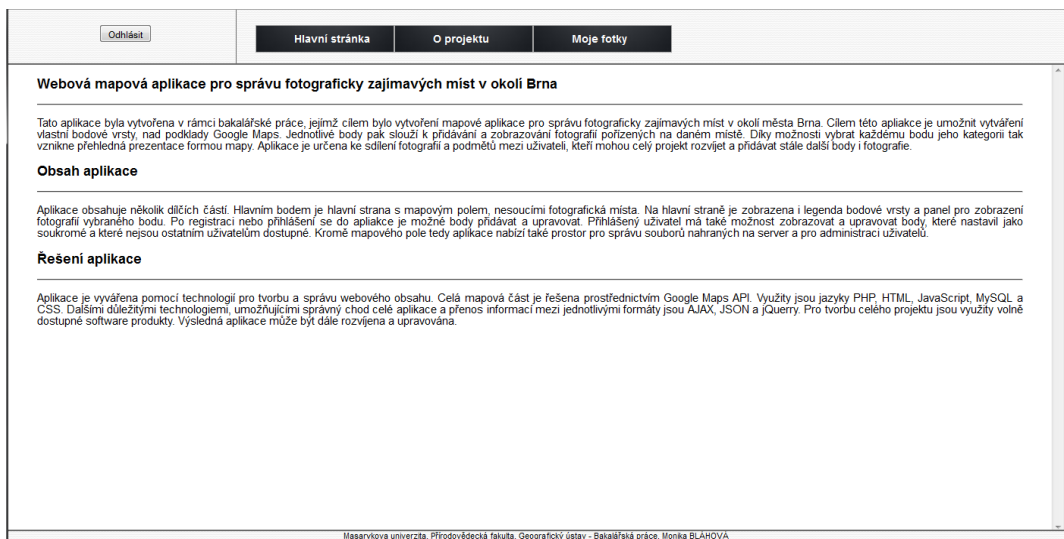
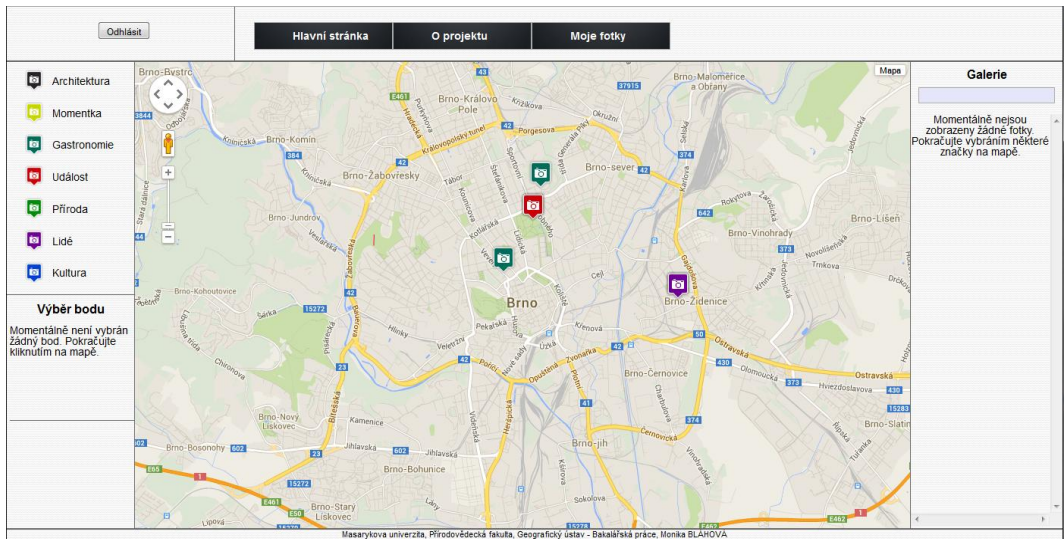
} else{
    fotoform_build_new ($zem_sirka,$zem_delka);
}
?>

```

Příl. 6. Rozvržení aplikace



Príl. 7. Vizualní řešení aplikace



Vyberte soubor, který chcete nahrát: [Procházet](#) Soubor nevybrán.

Poloha bodu: Zeměpisná šířka: Zeměpisná délka:

Název:

Popis:

Kategorie:

Nastavení soukromí:

Uživatelské jméno:  Pod uživatelským jménem bude možné přidávat Vaše fotografie.

Heslo:  Heslo by mělo obsahovat alespoň 6 znaků.

Ověření hesla:

E-mail:  Váš e-mail bude sloužit jako přihlašovací údaj.

Ověření:  =  Vyplňte prosím ověřovací pole.

Po úspěšné registraci budete přihlášení a přesměrováni na hlavní stránku