



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Aplikace pro vyhodnocování návštěvnosti galerie

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie

Autor práce: **Zdeněk Devátý**
Vedoucí práce: Ing. Jana Vitvarová, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Application for evaluation of gallery visits

Bachelor thesis

Study programme: B2646 – Information Technology
Study branch: 1802R007 – Information Technology
Author: **Zdeněk Devátý**
Supervisor: Ing. Jana Vitvarová, Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Zdeněk Devátý**
Osobní číslo: **M15000010**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Aplikace pro vyhodnocování návštěvnosti galerie**
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Proveďte analýzu možností sběru dat o návštěvách v galerii. Zaměřte se na možnosti, které zatěžují návštěvníky co nejméně a zachovávají jejich anonymitu.
2. Specifikujte požadavky na aplikaci pro podporu vyhodnocování návštěvnosti galerie. Aplikace má mimo jiné umožnit vizualizovat návštěvy v závislosti na bydlišti a vytvářet různé přehledy.
3. Implementujte výše specifikovanou aplikaci.
4. Umožněte exporty dat do různých typů souborů a vytvořte REST API, aby mohly třetí strany přistupovat k vybraným datům.
5. Aplikaci otestujte na vzorku dat z Galerie Lázně v Liberci.



Rozsah grafických prací: dle potřeby dokumentace

Rozsah pracovní zprávy: 30–40 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

- [1] RICHARDSON, L., S. RUBY. RESTful Web Services: Web services for the real world. O'Reilly Media, 2007. ISBN-13: 978-0596529260.
- [2] McCONNELL, S. Code Complete: A Practical Handbook of Software Construction. Microsoft Press, 2nd Edition, 2004. ISBN-13: 978-0735619678.

Vedoucí bakalářské práce:

Ing. Jana Vitvarová, Ph.D.

Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: 10. října 2017

Termín odevzdání bakalářské práce: 14. května 2018

prof. Ing. Zdeněk Pliva, Ph.D.
děkan



Kolář
doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2017

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 14.5.2018

Podpis:



Abstrakt

Práce je zaměřená na přípravu softwarového nástroje pro podporu sběru a vyhodnocování dat o návštěvnosti galerie. Jsou zde analyzovány různé možnosti sběru dat o samotných návštěvnicích a jejich návštěvách. Následně je navržen software umožňující automatizovaný sběr informací, jejich vyhodnocování pro účely zlepšení nabízených služeb, efektivnější propagaci a poskytování anonymizovaných dat veřejnosti ve formě grafu, mapy a REST API.

Klíčová slova:

sběr dat, vizualizace dat, open data

Abstract

The goal of the thesis is to develop a software tool to help the collection and evaluation of the data about gallery visits. First, various possible methods of data collection are analysed, then, a software is designed and implemented. The software allows an automated data collection about visitors, evaluation of the data for enhancing service offers and more effective advertising, and provides anonymised data to public via graphs, maps and REST API.

Key words:

data collection, data visualisation, open data

Obsah

Seznam zkratk	8
1 Analýza dostupnosti a sběru dat	12
1.1 Pasivní sběr dat	12
1.1.1 Pohyb návštěvníků po budově	12
1.1.2 Informace o návštěvnících	13
1.1.3 Častí návštěvníci	13
1.2 Aktivní sběr dat	14
1.2.1 Zkoumání popularity děl	14
1.2.2 Zpětná vazba při odchodu z budovy	14
1.2.3 Původ návštěvníků	15
2 Specifikace požadavků na aplikaci	16
2.1 Uživatelská a technická specifikace	16
2.1.1 Funkční požadavky	16
2.1.2 Nefunkční požadavky	18
2.1.3 Systémové požadavky	19
3 Příprava dat	21
3.1 Záznamy o návštěvnosti galerie za rok 2017	21
3.2 Přiřazení PSČ ke geografickým lokacím	22
3.3 Přiřazení PSČ k oblastem v Libereckém kraji	24
3.4 Možnosti dělení oblastí podle PSČ	24
3.5 Plakátovací plochy v Liberci	26
4 Implementace aplikace	28
4.1 Prezenční vrstva	29
4.1.1 Zobrazení dat	30
4.1.2 Přizpůsobení	31
4.2 Datová vrstva	31
4.3 Aplikační vrstva	32
4.3.1 Správa uživatelů	32
4.3.2 Vkládání dat	34
4.3.3 Import dat ze souboru	35
4.3.4 Export dat do souboru	35
4.3.5 REST API	36

4.3.6	Graf	37
4.3.7	Mapa	39
4.3.8	Rychlost aplikace	40
4.3.9	Bezpečnost aplikace	40
4.3.10	Kompatibilita aplikace	41
5	Závěr	43

Seznam zkratek

PSČ	Poštovní směrovací číslo
API	Application Programming Interface
REST	Representational State Transfer
URI	Uniform Resource Identifier
WGS84	World Geodetic System 1984
CSV	Comma-Separated Values
JSON	JavaScript Object Notation
PHP	PHP: Hypertext Preprocessor
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
TLS	Transport Layer Security
SQL	Structured Query Language
HTTP	Hypertext Transfer Protocol
CSRF	Cross-site Request Forgery
XSS	Cross-site Scripting
AJAX	Asynchronous Javascript and XML

Seznam obrázků

2.1	Diagram užití (use case diagram) aplikace	17
3.1	Softmax	25
3.2	Reklamní plochy společnosti Rengl v Liberci	26
4.1	Schéma aplikace	29
4.2	Diagram tříd (class diagram) serverové části aplikace	29
4.3	Vizualizační část aplikace	30
4.4	ER diagram databáze	31
4.5	Správa uživatelů	33
4.6	Přihlašovací formulář aplikace	34
4.7	REST API	37
4.8	Příklad vykreslení grafu	38
4.9	Kartogram	40
4.10	Testování zadávání dat na starších zařízeních	42

Seznam tabulek

3.1	Výňatek z původní tabulky dat	21
3.2	Oblasti PSČ v Liberecké kraji	23
4.1	Příklad exportovaných dat	36
4.2	Podporované prohlížeče a platformy	42

Úvod

Oblastní galerie v Liberci má zájem o zlepšení sběru a využití dat o původu návštěvníků. V současné době jsou v galerii sbírána data o poštovních směrovacích číslech oblastí, ze kterých návštěvníci pocházejí. Avšak tyto informace jsou vkládány do tabulky bez jednoduchého uživatelského rozhraní, což není příliš přehledné a rychlé, a vytváří to prostor pro chyby a ztrátu dat. Dalším problémem současného systému je, že je nutné data před odevzdáním zřizovateli ručně zpracovat. Navíc když už galerie data o návštěvnících sbírá, také by chtěla tato data využít pro svou vlastní analýzu. Galerie má také zájem o prozkoumání možností sbírání dalších dat, které by mohly pomoci v komunikaci s veřejností a potenciálními návštěvníky.

Prvním cílem této práce tedy je zanalyzovat v současnosti dostupná data a možnosti sběru dalších dat. Je třeba se zaměřit na takové možnosti, které nijak nekomplikují návštěvu galerie. Povinné vyplňování dotazníků by mohlo návštěvníky odradit, což samozřejmě není žádoucím účinkem. Soukromí návštěvníků musí být samozřejmě také zachováno.

Hlavní částí práce je specifikovat a implementovat aplikaci pro podporu vyhodnocování návštěvnosti. Tato aplikace má umožnit zaměstnancům galerie zadávat data do databáze, zobrazit grafy vyjadřující návštěvnost galerie a mapu ve formě kartogramu původu návštěvníků. Vizuální přehledy mají být dostupné i pro veřejnost. Pro předávání dat zřizovateli má aplikace vytvořit shrnutí návštěvnosti a poskytnout je v různých textových formátech.

Součástí aplikace má být také REST API umožňující automatizované využití dat v jiné aplikaci.

Aplikace má být s drobnými úpravami připravená pro použití v různých galeriích. Pro testování však má být použit vzorek dat z liberecké Galerie Lázně.

1 Analýza dostupnosti a sběru dat

Galerie umění a podobné instituce mají různé možnosti získávání dat o návštěvnicích, z nichž některá již sbírají a další sbírat mohou. Je možné využít pasivních nebo aktivních metod. Je vhodné zachovat anonymitu návštěvníků a sběr dat by neměl návštěvníky obtěžovat.

1.1 Pasivní sběr dat

Pasivní sběr dat probíhá bez zásahu návštěvníka, tudíž jej nijak neobtěžuje. Pro určité pasivně sbírané údaje je však potřeba souhlas nebo alespoň upozornění subjektu.

1.1.1 Pohyb návštěvníků po budově

Na trhu je mnoho dostupných řešení pro počítání lidí procházejících určitým úsekem. Tyto systémy mohou pracovat pomocí optických či mechanických senzorů, RFID snímačů nebo lze využít i záznamy ze stávajících bezpečnostních kamer.

Kamery

Pro analýzu zájmu o jednotlivé expozice by bylo možné využít stávající systémy bezpečnostních kamer. Ty by vzhledem ke svému určení měly s dostatečnou kvalitou obrazu zaznamenávat většinu veřejných prostor galerie. Tyto záznamy je třeba dále zpracovat a na základě vizuální podobnosti rozpoznávat jednotlivé osoby. Podle toho by již bylo možné sestavit vzorce chování typických návštěvníků. Pro zpracování záznamů by bylo možné využít nástrojů strojového učení.

Dalším, technicky jednodušším řešením by bylo sloučit data z jednotlivých kamer do jednoho obrazu a vytvořit teplotní mapu ukazující místa, na která chodí nejvíce návštěvníků, na kterých se nejdéle zdržují, či o která naopak návštěvníci nejeví zájem. K tomu by mělo stačit poměrně jednoduché zpracování obrazu, které umí i nástroje s otevřeným zdrojovým kódem. Podmínkou by bylo, aby kamery byly nepohyblivé. Pokud by se kamery pohybovaly v průběhu dne, bylo by mnohem složitější obraz složit.

RFID snímače

RFID čipy se často využívají ve sportovní časomíře, kde mají závodníci čip připevněný buď na těle, nebo vytištěný přímo na startovním čísle. Tyto čipy jsou pasivní

a obsahují jednoznačný identifikátor závodníka, který snímá RFID čtečka na okraji snímaného místa nebo vytvořená z kabelů natažených pod tratí. Takový systém spolehlivě zaznamenává časy i pro tisíce závodníků během krátkého časového intervalu, tudíž by nebyl problém se spolehlivostí ani při větším počtu návštěvníků.

Tyto čipy by mohly být umístěny například v náramku, který by si návštěvníci dali na ruku jako například v bazénech, ale to by nemuselo být všem návštěvníkům příjemné a mohlo by to rušit atmosféru, která v galerii panuje. Místo toho by mohli návštěvníci dostávat vstupenky, na kterých by byly RFID čipy s anténou. Tyto by mohly být ve formě vratných karet (návštěvník je při odchodu vhodí do turniketu, který mu následně umožní průchod), nebo nevratné a prodávané místo současných papírových vstupenek. Vzhledem k tomu, že identifikátor čipu má 96 bitů, není zde problém s nedostatkem unikátních označení a cena RFID čipu také není příliš vysoká. Pro UHF čip se pohybuje mezi 0,07 a 0,15 USD (1,49 - 3,20 CZK)[1], takže by se to na ceně vstupenky výrazně neprojevilo.

Použití RFID čipů by umožnilo přesné sledování pohybu návštěvníků po budově, jako například kolik času v které místnosti stráví, v jakém pořadí místnosti navštěvují a zda-li se někdy do místností vrací.

Optické snímače

V rámech dveří mohou být umístěna optická počítadla, například infračervený PYRO senzor od společnosti Eco-Counter[2], která by počítala návštěvníky procházející z/do místnosti. Pokud by v rámu byly umístěny dva snímače za sebou, dalo by se softwarově rozpoznat, kterým směrem návštěvník prošel (dovnitř, nebo ven z místnosti), čímž bychom měli k dispozici o něco podrobnější data.

Mechanické snímače

Variantou podobnou optickým počítadlům jsou mechanické snímače, například ve formě sčítací koberce jako je produkt Instant Counting od společnosti Eco-Counter[3]. Výhodou je jednodušší instalace (není nutné zabudování do dveří) a zřejmě vyšší spolehlivost při průchodu více lidí zároveň. Umožňují také určení směru průchodu. Výrobce však neuvádí cenu, což je faktor, který by mohl při výběru rozhodnout.

1.1.2 Informace o návštěvnících

Pro pasivní získání dat o návštěvnících by bylo třeba použít systém, který by tato data získával například z chytrých telefonů návštěvníků, což by ovšem bylo nejen náročné na implementaci, ale především by to mohlo představovat velké narušení soukromí. Proto by pro tento účel bylo třeba se návštěvníků přímo dotázat neboli využít aktivní sběr dat.

1.1.3 Častí návštěvníci

Návštěvníkům by mohla být nabízena věrnostní kartička, která by umožnila sledovat, jak často a kdy chodí do galerie. Tato kartička by také mohla obsahovat

RFID čip a v případě, že by byl v galerii zaveden výše zmíněný systém RFID snímačů, by bylo možné ještě lépe porozumět návykům návštěvníků. V tomto případě by o konkrétních návštěvnících bylo dostupné větší množství informací, jelikož při registraci do věrnostního programu jsou lidé ochotni o sobě poskytnout podrobné informace. Registrace do tohoto programu by se řadila mezi aktivní sběr dat, ale vzhledem k tomu, že by ji návštěvník absolvoval pouze jednou a mohl by být motivován slevou ze vstupného, pravděpodobně by odpovídání na otázky nepovažoval za obtěžující.

1.2 Aktivní sběr dat

Aktivní sběr dat může být pro návštěvníka obtěžující, protože vyžaduje, aby věnoval čas a pozornost průzkumu či dotazníku. Některé obchody a instituce mají zavedený terminál u východu, kde zákazník může, pokud chce, stisknout tlačítko, zda byl spokojen se službami, případně zanechat další poznámky. Jiné společnosti toto řeší papírovými nebo elektronickými dotazníky. Společnosti jako banky volají klientům po návštěvě pobočky, aby získaly zpětnou vazbu.

1.2.1 Zkoumání popularity děl

Užitečnou informací pro galerii jsou statistiky oblíbenosti jednotlivých vystavených děl. Podle toho by mohla lépe připravovat další výstavy a vybírat nová díla.

Aktivní použití snímačů

Návštěvníci by mohli dostat při vstupu RFID čip, který mohou u každého vystaveného díla přiložit ke snímači, a tím označit, že se jim dané dílo líbí.

Aby byli návštěvníci motivováni označovat oblíbená díla, galerie by jim měla nabídnout možnost si později na Internetu prohlédnout díla, která při návštěvě označili, a získat o nich dodatečné informace.

1.2.2 Zpětná vazba při odchodu z budovy

Bezprostředně po návštěvě expozice by bylo možné získat přímo zpětnou vazbu od návštěvníka, zjistit, co jej zaujalo a co pro něj naopak zajímavé není. Tímto bychom získali názor návštěvníků, bylo by ale potřeba najít kompromis mezi zatěžováním návštěvníků a podrobností sbíraných dat.

Dotazník

Při odchodu by mohla galerie požádat návštěvníka o zpětnou vazbu formou dotazníku, a to buď elektronického nebo papírového.

Elektronický dotazník by mohl být na terminálu u východu, kde by byl spuštěn jako jednoduchá aplikace, která interaktivně provede návštěvníka několika otázkami. Výhodou tohoto řešení oproti papírovému by bylo snadnější získání dat (data

se ukládají přímo do databáze) a možnost vytvořit interaktivní dotazník. Nevýhodou by naopak byla vyšší pořizovací cena a omezená kapacita. Terminál by mohl být tvořen běžným spotřebitelským tabletem, ale do ceny by se musel započítat i vývoj aplikace. Kapacita takového zařízení by byla omezena na jednoho člověka na terminál, zatímco papírové dotazníky může vyplňovat libovolný počet lidí naráz.

Pokud by bylo vyplnění dotazníku povinné, mohlo by to mít negativní dopad na spokojenost návštěvníků, pokud by naopak bylo pouze pár listů dotazníku položených na nějakém nevýrazném místě, nebyl by získán dostatek dat. Bylo by tedy potřeba návštěvníka zaujmout a navést jej k dotazníku, ale nevyžadovat jeho vyplnění.

Dotazníkem by bylo možné získat mnoho informací, ale ten by zároveň musel zůstat krátký a stručný, aby návštěvníka od svého vyplňování neodradil.

Tlačítka pro zpětnou vazbu

Další variantou aktivního sběru dat by mohly být tlačítka u východu, v jakékoliv podobě, kde by návštěvník mohl dát najevo, jak byl s návštěvou či expozicí spokojen. Toto by návštěvníka téměř vůbec nezdrželo a byla by větší pravděpodobnost, že by nějakou zpětnou vazbu poskytl. Nevýhodou tohoto řešení by bylo, že by neposkytovalo žádná vysvětlení, proč byl návštěvník spokojen/nespokojen, takže by se z takových dat hůře dělaly závěry či doporučení.

1.2.3 Původ návštěvníků

Pro galerii by bylo výhodné vědět, odkud její návštěvníci pochází. Mohla díky tomu zlepšit propagaci - například výběr výlepových ploch pro plakáty. Tyto údaje také potřebují vykazovat zřizovateli.

Poštovní směrovací číslo by bylo pro určení původu návštěvníků pro tento účel dostatečně přesná informace, protože lze předpokládat, že se obyvatel pohybuje po širším okolí svého bydliště. Zároveň bychom také po návštěvníkovi neměli požadovat přesnější informaci, aby byla zachována dostatečná anonymita.

2 Specifikace požadavků na aplikaci

Při přípravě a testování aplikace je jako vzor galerie použita Oblastní galerie Liberec, známá také jako Galerie Lázně. Liberecký kraj, její zřizovatel, jí nařizuje vykazovat statistiku návštěvnosti podle míst původu návštěvníků.

Galerie si vybrala jako informaci o původu návštěvníků poštovní směrovací čísla. V současné době galerie informace sbírá tak, že návštěvník je již u vstupu při koupi vstupenky dotázán, z jakého PSČ pochází. Zjištěné údaje jsou následně zadány do tabulky.

Tento systém však není vyhovující, vzhledem k tomu, že tabulka nemá jednoduché uživatelské rozhraní, ani žádným způsobem ošetřené vstupy. Tímto v datech vznikají nekonzistentní údaje a data se pak obtížně zpracovávají. Také zpracování a předávání dat zřizovateli není nijak zautomatizováno. Data se musí každý rok ručně zpracovat v tabulkovém editoru a zřizovateli poslat.

Nakonec když už galerie data sbírá, chce je zároveň využít pro svou vlastní potřebu. Z tohoto důvodu je vhodné data přehledným způsobem vizualizovat.

Cílem této aplikace je tedy umožnit snadné zadávání dat, jejich vizualizaci a automatické zpracování a předávání zřizovateli.

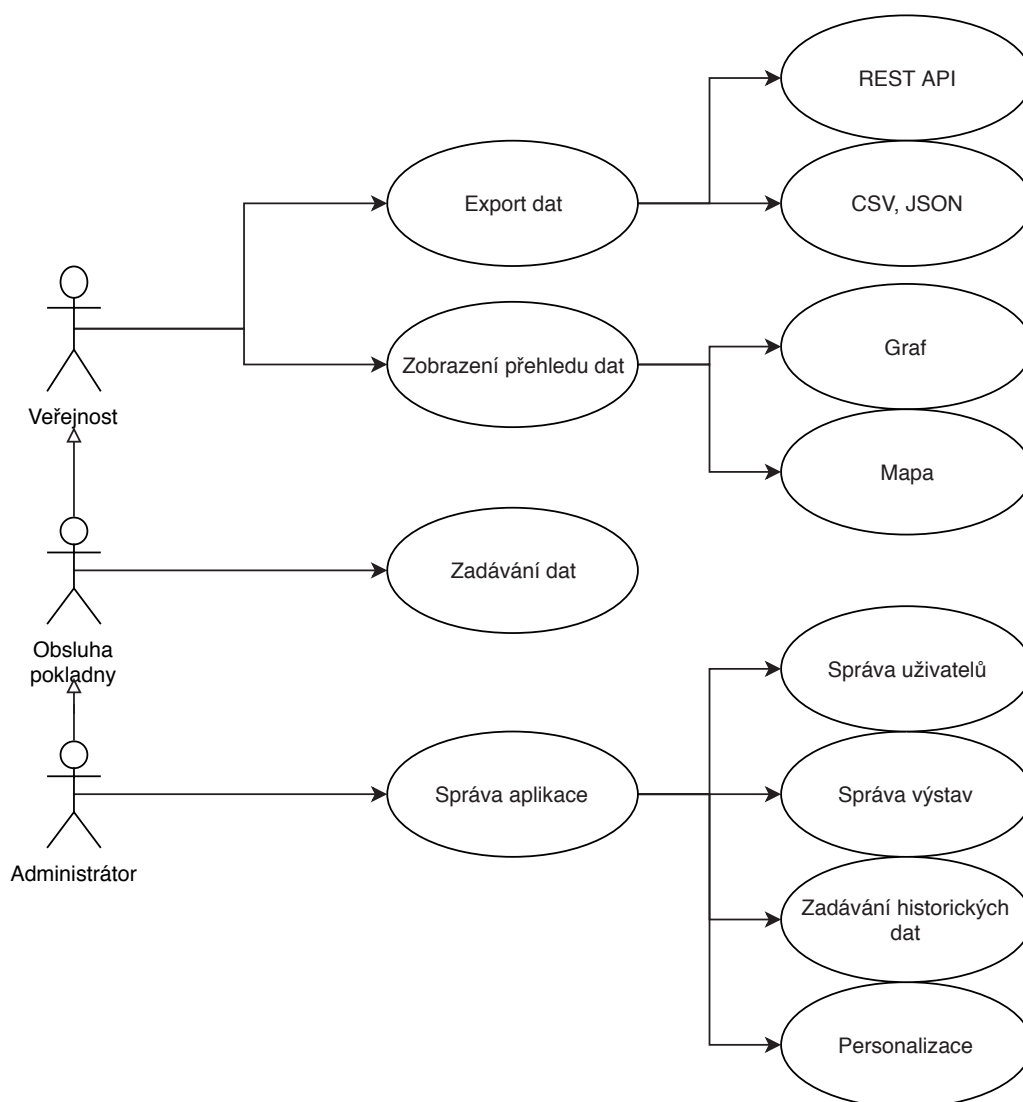
2.1 Uživatelská a technická specifikace

Uživatelské požadavky na aplikaci se dají rozdělit do dvou skupin - funkční a nefunkční. Funkční vyjadřují přímo požadované vlastnosti aplikace, neboli „co má aplikace umět,“ a nefunkční požadavky vlastnosti této aplikace upřesňují. Například zaznamenávání dat a jejich vizualizace jsou funkční požadavky, naopak bezpečnost a uživatelská přívětivost či moderní vzhled jsou nefunkční požadavky.

Systémové požadavky představují technická omezení, informace o dostupných technologiích a technických požadavcích na funkčnost (například podporované klientské aplikace).

2.1.1 Funkční požadavky

Na obrázku 2.1 jsou ve formě use-case diagramu znázorněné základní funkční požadavky na aplikaci, jenž jsou níže rozepsané.



Obrázek 2.1: Diagram užití (use case diagram) aplikace

Zadávání dat

Současný systém zadávání dat do tabulky není optimální, proto bude dalším cílem vytvořit grafické rozhraní, do kterého budou při prodeji vstupenek zapisována zjištěná data.

Aplikace tedy má být primárně připravena pro přímé zadávání údajů do databáze přes webovou aplikaci.

Vzhledem k tomu, že někdy přichází větší skupina návštěvníků (například zájezd nebo škola) ze stejné oblasti najednou, chce mít galerie i možnost snadného vícenásobného zadání PSČ do databáze.

K tomu má mít administrátor aplikace možnost přidat data v textovém formátu CSV, jelikož jsou data v současnosti sbírána do tabulek Google Sheets. Po drobné úpravě formátu dat z těchto tabulek do požadovaného tvaru má mít administrátor možnost vložit informace o návštěvách za libovolné období hromadně.

Zobrazení přehledu dat

Další funkcí aplikace je vyhodnocovat a vizualizovat návštěvnost galerie v grafu a mapě. Výstupem má být zobrazení bydliště návštěvníků na mapě a vygenerované přehledy návštěvnosti.

Uživatel má tedy mít možnost původ návštěvníků seskupit podle krajů, okresů nebo podle přesnějších oblastí v Libereckém kraji.

Přehledy mají obsahovat grafy vyjadřující počty návštěvníků. Tyto počty mají být rozdělené podle původu návštěvníků a seskupené po dnech, týdnech či měsících mají vyjadřovat zvolené libovolné období. Toto období má být volitelné podle počátečního a závěrečného datumu, případně podle výstavy - od začátku do konce.

Mapa má být v podobě kartogramu, kde je barvami vyznačeno, kolik lidí z jakých oblastí přichází. Zobrazení mapy se má řídit stejnými volbami jako zobrazení grafu, vyjma seskupení podle času (vzhledem k tomu, že v mapě není na rozdíl od grafu zobrazen rozměr času).

Export dat

Aplikace má také umožnit zapsat vyhodnocená data do souborů různých formátů. Má být k dispozici možnost stažení dat (seskupených podle kritérií vizualizace) ve formátech JSON a CSV.

Vybraná data poté mají být zpřístupněna pro použití třetími stranami například pro výzkum. Proto bude potřeba vytvořit REST API obsahující informace o počtech návštěv po dnech, včetně dalších informací o specifikách daného dne. Toto umožní automatické zpracování dat zřizovatelem, webovou aplikací galerie nebo třetími stranami.

Rest API je rozhraní poskytující informace přímo adresovatelné pomocí URI. Podstatnou vlastností je idempotence a bezpečnost funkcí, aby výsledek operace nebyl ovlivněn například opakováním volání stejné funkce. Další požadovanou vlastností je bezstavovost, tedy aby každý požadavek obsahoval všechny informace potřebné k vyplnění požadavku. Jednotlivé zdroje také mají být propojené, aby se klient mohl ve službě sám orientovat.[5]

Správa aplikace

Administrátor aplikace má mít možnost spravovat uživatelské účty a výstavy a hromadně nahrávat data. Vzhledem k tomu, že má aplikace být univerzální, má mít administrátor možnost personalizovat aplikaci, například nastavit logo instituce, název instituce a ikonu aplikace.

2.1.2 Nefunkční požadavky

Dodržení kodexu ICC/ESOMAR

Aplikace má dodržovat pravidla Mezinárodního kodexu ICC/ESOMAR[4]. Tento kodex marketingového a sociálního výzkumu je vytvořený Mezinárodní obchodní

komorou (ICC) a ESOMAR, globálním sdružením datové, výzkumné a „insight“ komunity. Tento Kodex je navržen jako komplexní rámec samoregulace pro ty, kteří se zabývají výzkumem trhu, sociologickým výzkumem a datovou analytikou.

Kodex definuje rozdíl mezi primárními daty, což jsou „údaje shromážděné pro účely výzkumu“, a sekundárními daty, „údaji shromážděnými pro jiný účel a následně využitými v oblasti výzkumu“. Vzhledem k tomu, že jsou údaje o PSČ získávány především z nařízení zřizovatele a jejich využití v analýze návštěvnosti je pouze nově přidaná možnost, při analýze je nutné k nim přistupovat jako k sekundárním datům. Ovšem rozhraním pro zadávání dat budou procházet povinně sbírané údaje ještě před jejich prvotním využitím, proto je potřeba na tuto část systému nahlížet jako na systém pro práci s primárními daty, to znamená, že návštěvník musí být seznámen a musí souhlasit se způsobem využití dat.

Zabezpečení

Vzhledem k tomu, že informace uchovávané o návštěvnících neumožňují přímou identifikaci (nejsou moc konkrétní), nebude potřeba použít příliš silné zabezpečení. Ovšem ne všechna data budou určena pro zveřejnění, proto má systém být dostatečně zabezpečený a robustní.

Má také umožnit různým uživatelům přístup do různých částí aplikace. Data ve formě map a grafů mají být zpřístupněna pro veřejnost na webové stránce galerie nebo na stránce samostatné, přímo k tomu určené. Dalším, kdo má mít k aplikaci přístup, je obsluha pokladny, která informace zadá do databáze. Nastavení aplikace a její pokročilé funkce mají být přístupné už pouze administrátorovi.

Vhodné uživatelské rozhraní

Pro zadávání dat má být systém rychlý, ideálně by měl umožnit ovládání pouze klávesnicí, aby bylo zadávání dat co nejrychlejší a nezdržovalo ani návštěvníka, ani zadávajícího zaměstnance. Má také být jednoduchý a předvídatelný, aby nebylo potřeba zaměstnance práci se systémem učit.

Vizualizační část má být přehledná a taktéž jednoduše ovladatelná.

Přizpůsobení zadávání dat menším zařízením

Zadávání dat má mít jednoduché uživatelské rozhraní a má být přehledná i na menších zařízeních. Má totiž umožňovat použití různých velikostí obrazovky, aby aplikace byla nezávislá na použitém terminálu. Zadávání pak má být stejně možné na počítači, jakož i na tabletu či mobilním telefonu.

2.1.3 Systémové požadavky

Pro účely testování bude aplikace dostupná na sdíleném hostingu, kde je spuštěn webový server Apache, databáze MySQL a PHP 7.1. Vzhledem k omezením testovacího webhostingu, kde není možné získat SSH přístup, je možné použít na straně serveru pouze základní prostředí PHP bez přídatných knihoven.

Server poskytuje zabezpečení přenosu pomocí protokolu SSL, jenž je dnes doporučován pro veškeré webové stránky. To umožňuje bezpečný přenos všech dat, včetně těch citlivých jako například přihlašovacích údajů.

Aplikace má dosáhnout plné funkčnosti na všech hlavních moderních prohlížečích na významných platformách pro osobní počítače.

3 Příprava dat

Pro vytvoření aplikace bylo potřeba získat, analyzovat a upravit velké množství dat. Pro všechnu práci s těmito daty byl použit programovací jazyk Python. Ten byl zvolen díky tomu, že umožňuje rychle psát jednoduché aplikace, obsahuje vysokou abstrakci a je možné využívat některých výhod funkcionálního programování. Také jsou pro Python dostupné balíky pro práci se soubory různých formátů a s protokolem HTTP pro získávání dat z webových stránek.

3.1 Záznamy o návštěvnosti galerie za rok 2017

Tabulka 3.1: Výňatek z původní tabulky dat

3. 12.	9. 12.	10.12.	12.12.	13.12.	14.12.	15.12.	16.12.	17.12.
DDR	11000	46001	54301	53701	čt - zdarma	46601	46001	47001
	11000	46001	54301	53701	176 lidí	46601	46001	47001
	46841	11000	54301	11000		15000	46001	46001
	46841	11000	54301	11000		15000	46001	39816
	46841	50002	54301	46001		46010	46001	39816
	28401	50002	54301	27735		46010	46001	46331
	18000	11000	54301			46001	46001	46331
	18000	46001	54301			46001	46001	14000
	18000	46001	54301			29301	46001	14000
	18000	46334	46001				46001	14000
	18000	46334					46001	46001
	46001	51101					46001	46001
	46001	46001					46001	46001
	46001	46001					46001	46001
	46001	46001					46001	16900

Pro testování aplikace byla získána data, která galerie sbírala během roku 2017. Informace o PSČ byly dosud sbírány do tabulek v on-line tabulkovém editoru, a to v nekonzistentní podobě. Na jednotlivé listy reprezentující měsíce byly do sloupců vkládány PSČ za jednotlivé dny, jak je vidět v tabulce 3.1. Hlavičky tabulek byly psány ručně, čímž byly datумы popisovány v různých tvarech. To by komplikovalo algoritmické zpracování těchto dat. Pro původní účel tabulek byl ale tento způsob dostačující, jelikož se data nezpracovávala podle jednotlivých dnů, nýbrž pouze

podle měsíců. Datумы tedy sloužily pouze jako orientační prvek pro zadávajícího. Pro analýzu v této aplikaci je ale potřeba data rozdělit na dny.

Dalším problémem byla data vyjadřující počet návštěvníků. Zatímco začátkem roku byly směrovací čísla do tabulky zadávána ve správném tvaru (na každý řádek jedno PSČ vyjadřující původ jednoho návštěvníka), a to včetně dnů, kdy byl vstup do galerie zdarma, v průběhu roku nastala pravděpodobně změna v personálu na pokladně a místo zápisu PSČ u každého návštěvníka byl u těchto dnů pouze zápis „ n lidí“, kde n je počet lidí navštěvujících galerii v ten den (jak je vidět v tabulce 3.1). Později se v datech objevily další anomálie, což vedlo k rozhodnutí, že budou data pro zadání do databáze ručně předzpracována.

Data byla z tabulkového editoru exportována do formátu CSV a následně upravena v pokročilém textovém editoru. Ten umožnil snadnou změnu formátu datumů pomocí regulárních výrazů. Zde byla také odstraněna část neplatných dat. Pro účely testování a ukázky aplikace nebyla použita data z konce roku, ale kvalitnější data ze začátku roku 2017, zopakovaná pro konec roku 2017. Tím byla vytvořena množina dat obsahující všechny potřebné údaje pro vizualizaci dat podle oblastí. Tato data byla také použita pro rozšíření až do poloviny roku 2018, aby bylo už na první pohled v aplikaci zřejmé, jak vizualizace vypadá.

Tyto soubory, stále ve formátu CSV, byly poté zpracovány a nahrány do databáze pomocí nástroje v administraci aplikace, který je popsán v podkapitole „Import dat ze souboru“ na straně 35.

3.2 Přiřazení PSČ ke geografickým lokacím

Původní myšlenkou bylo rozdělit Českou Republiku podle PSČ a zobrazit počty návštěvníků z těchto oblastí. Byly zkoumány možnosti zobrazování takových oblastí na dynamické mapě, avšak žádné z vyzkoušených mapových rozhraní rozdělení podle PSČ zobrazovat neumožňovalo (např. [6]). Pro tento účel byly vyzkoušeny rozhraní Mapy API společnosti Seznam.cz, A.S.¹, aplikace OpenStreetMap nadace OpenStreetMap Foundation a Maps API od společnosti Google, Inc.. Na konci této kapitoly na straně 24 jsou popsány pokusy o vytvoření vlastních ohraničení pomocí paralelního lineárního klasifikátoru a Voroného teselace. Toto bylo vyzkoušeno, jelikož nebylo nalezeno ani žádné veřejně dostupné ohraničení oblastí různých poštovních směrovacích čísel.

Vzhledem k tomu, že zobrazování dat podle jednotlivých PSČ by bylo příliš podrobné a nebyl nalezen jednoduchý způsob jejich zobrazování, byly po domluvě s galerií vybrány tři druhy dělení oblastí: podle krajů, podle okresů a podle oblastí v Liberecké kraji, shrnuté v tabulce 3.2.

¹<https://api.mapy.cz/>

Tabulka 3.2: Oblasti a začátky jejich PSČ, na něž byl rozdělen Liberecký kraj

Začátek PSČ	Název
4631, 460	Liberec
4633	Chrastava, Hrádek
4634	za kopcem
4635	Osečná
4636	Hejnice, Nové Město
4637	Habartice
464	Frýdlant
468	Jablonecko
466	Jablonec
463	Hodkovice
470	Česká Lípa
471	Mimoň, Cvikov, Jablonné
472	Doksy
473	Nový Bor
511	Turnov
513	Semily
512	Podkrkonoší
514	Jilemnice

Vzhledem k zjednodušenému dělení bylo možné získat podklady pro kartogram kombinací informací z České pošty a dat OpenStreetMap.

Česká pošta poskytuje na svých webových stránkách² soubor „Seznam PSČ částí obcí a obcí bez částí“, který obsahuje tabulku s PSČ, názvem obce, názvem pošty, názvem okresu a kódem okresu.

Data OpenStreetMap, poskytovaná nadací OpenStreetMap Foundation volně k použití, jsou rozdělená podle úrovní administrativního dělení, takže se dají přímo použít pro kartogram dělený podle krajů a okresů. Tato data jsou dostupná přes webovou aplikaci OSM Boundaries Map³. Ta poskytuje API, kde je možné si zvolit oblasti, jejich ohraničení, případně ohraničení jejich sjednocení, z čehož server vytvoří GeoJSON soubor.

Vygenerovaný GeoJSON soubor obsahuje informace o oblastech, jejich identifikační číslo, název a identifikační čísla oblastí, do nichž náleží (například Liberec obsahuje informaci o tom, že náleží do okresu Liberec, Libereckého kraje na Severovýchodě Čech a do České Republiky).

Pro optimální uložení do databáze bylo potřeba získat z těchto dat názvy krajů s jejich identifikačními čísly, k nim přiřadit okresy a k okresům přiřadit jednotlivá směrovací čísla, která do nich náleží. Avšak kódy okresů v datech České pošty nekorespondoval s kódy okresů v datech OpenStreetMap.

K tomuto účelu byl vytvořen Python skript, který načetl data o krajích, zjistil jejich názvy a kódy, poté načetl data o okresech, u nichž získal kromě názvů a kódů

²<https://www.ceskaposta.cz/ke-stazeni/zakaznicke-vystupy>

³<https://wambachers-osm.website/boundaries/>

také kód kraje, ke kterému náleží, a nakonec načetl výše zmíněná data o směrovacích číslech od České pošty. Z těchto dat získával informace o směrovacím čísle a názvu okresu, který byl porovnán s názvem okresu z dat OpenStreetMap. Ke každému směrovacímu číslu byl tím přiřazen identifikátor okresu, do něhož náleží. Přestože byly okresy přiřazovány pouze slovně, nikoliv pomocí číselných identifikátorů, přiřazení proběhlo bez problémů a kompletně.

Zmíněný Python skript nakonec vytvořil DML Insert příkaz pro přímé vložení dat do databáze.

3.3 Přiřazení PSČ k oblastem v Libereckém kraji

Některá poštovní směrovací čísla neposkytují dostatečně přesné určení polohy pro účely zkoumání původu návštěvníků v rámci jednoho města. V případě Liberce se jedná například o PSČ 460 01, které pokrývá velkou část okolí Liberce, ale i několik částí centra města. Avšak vzhledem k tomu, že bylo zjištěno, že po Liberci jsou plakátovací plochy společnosti, jejíž služby galerie využívá, rozloženy rovnoměrně a rozmístění plakátů si tato společnost řídí sama, bylo rozhodnuto, že Liberec ve vizualizacích rozdělen nebude. Rozdělen bude pouze Liberecký kraj na části podle začátku jejich směrovacího čísla, podle tabulky 3.2.

Avšak toto dělení Libereckého kraje neodpovídá žádným konkrétním administrativním dělením, proto je potřeba provést průnik více oblastí podrobnějšího administrativního dělení. To sice není triviální operace, ale tento problém opět umožňuje řešit aplikace OSM Boundaries Map.

Pro získání parametrů pro použití API OSM Boundaries Map byl vytvořen další Python skript, který postupuje přes zadané začátky PSČ, najde k nim odpovídající záznamy v souboru od České pošty, z těchto záznamů získá názvy obcí patřících k daným PSČ a ty pak vyhledá v názvech oblastí v datech OpenStreetMap. Experimentálně bylo zjištěno, že nejlepší výsledky tohoto přiřazení jsou dosaženy při použití OpenStreetMap administrativního členění na úrovni 8 (zhruba odpovídajícímu jednotlivým obcím a okolí). Při použití úrovně 9 nebyla pokryta všechna místa v Libereckém kraji a na úrovni 7 program nebyl schopen rozčlenit všechny oblasti.

Tím byla získána identifikační čísla administrativních oblastí, ta byla následně zadána do OSM Boundaries Map API, které poskytlo GeoJSON soubory obsahující ohraničení jejich průniku. Tyto soubory již mohou být přímo použity v kartogramu ve vizualizaci.

3.4 Možnosti dělení oblastí podle PSČ

V této práci byly oblasti rozděleny na kraje, okresy a na zvláštní oblasti v Libereckém kraji přímo podle přání galerie. Zkoumány ale byly různé metody, kterými by se dala mapa republiky rozdělit podle poštovních směrovacích čísel.

Použita k tomu mohou být veřejně dostupná data RÚIAN⁴, která obsahují údaje

⁴www.cuzk.cz/RUIAN/RUIAN.aspx

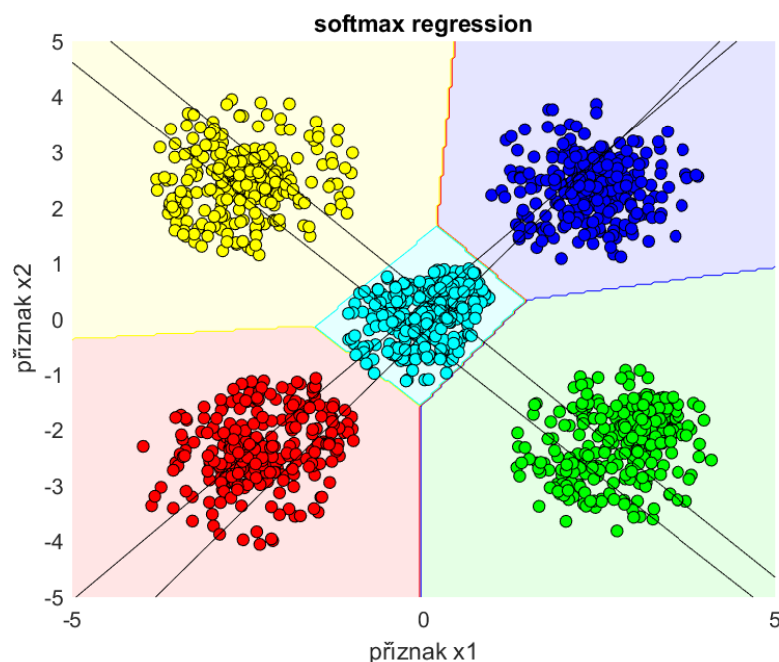
o základních územních prvcích, včetně PSČ.

Tato data byla dostupná v textovém formátu CSV a byla rozdělena podle obcí, tzn. 6 258 textových souborů. Pro další zpracování bylo potřeba soubory sloučit a pro snadnější použití oddělit pouze potřebná data, tedy souřadnice a směrovací čísla. Toho bylo dosaženo vytvořením a spuštěním jednoduchého parsovacího kódu v jazyku Python.

Pro samotné rozdělení oblastí bylo vyzkoušeno použití klasifikátoru softmax a bylo zvažováno také použití algoritmu Voroného teselace.

Paralelní lineární klasifikátor

Paralelní soustava binárních lineárních klasifikátorů umožňuje klasifikaci do více tříd. Při trénování se vytvoří model obsahující vektory rozdělující oblasti, kam jednotlivé body náleží. Podle těchto vektorů by byla následně rozdělena mapa. Na obrázku 3.1 je vidět, jakým způsobem paralelní lineární klasifikátor rozděluje oblasti.



Obrázek 3.1: Vizualizace paralelního lineárního klasifikátoru s funkcí softmax pro 5 tříd

Vzhledem k velkému počtu dat (2 913 787 bodů a 2 672 jednotlivých směrovacích čísel - tříd) byl vybrán efektivní paralelní lineární klasifikátor s funkcí softmax, ovšem i při použití této metody byla značně omezena velikost trénovacích dat, neboť v průběhu výpočtu je potřeba vícekrát vytvořit matici o velikosti [počet dat x počet tříd], což v tomto případě vyžaduje téměř 60 GB paměti, s čímž na běžných počítačích není prakticky možné pracovat. Při snížení počtu vzorků na méně než desetinu bylo možné model vypočítat, ale byl příliš málo přesný. Tato nepřesnost vznikala

jak nedostatkem vzorků, tak i kvůli faktu, že paralelní lineární klasifikátor nedokáže ohraničit nekonvexní plochy, což se v některých oblastech projevilo zásadním způsobem, takže model nebyl schopen konvergovat.

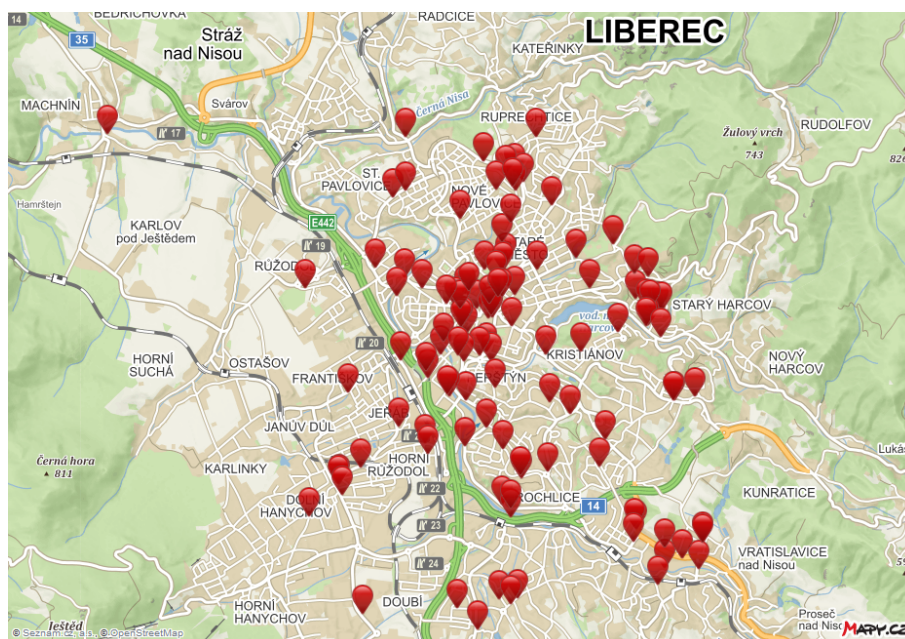
Voroného teselace

Řešení Voroného teselací pro nekonvexní plochy existuje, jak bylo publikováno v článku [7], ale toto řešení vyzkoušeno nebylo. Zůstává tedy jako možná cesta dalšího výzkumu.

3.5 Plakátovací plochy v Liberci

V rámci zkoumání možností využití dat bylo zvažováno objednávání plakátovacích ploch na základě původu návštěvníků galerie. Pro toto využití bylo potřeba zjistit, jak jsou plakátovací plochy po Liberci rozmístěny. Poskytovatel plakátovacích ploch pro Galerii Lázně má na svých stránkách zveřejněný seznam těchto ploch. Každý záznam obsahuje popis, fotografii a odkaz na interaktivní mapu. Součástí tohoto odkazu jsou WGS84 souřadnice dané plochy.

Tyto souřadnice byly ze stránky získány pomocí balíku requests a uloženy do souboru JSON. Tento soubor byl pak načten javascriptovým modulem a zobrazen pomocí rozhraní Mapy API v4.14 poskytovaného společností Seznam.cz, A.S.. Tím vznikla interaktivní mapa zobrazující body, na kterých se nachází plakátovací plochy této společnosti, zobrazená na obrázku 3.2.



Obrázek 3.2: Reklamní plochy společnosti Rengl v Liberci

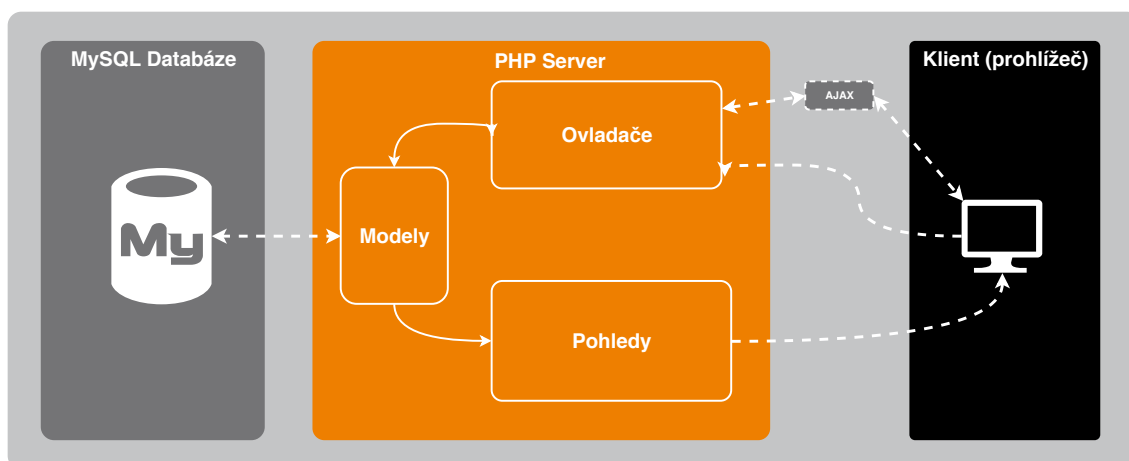
Díky tomu bylo na mapě vidět, že plakátovací plochy jsou rozmístěny zhruba rovnoměrně, s hustotou klesající se vzdáleností od středu města. Důležitým výstu-

pem z tohoto zkoumání bylo tedy to, že plakátovací plochy po Liberci nejsou nijak významně seskupené a nemá proto smysl podle nich rozdělit Liberec na více částí.

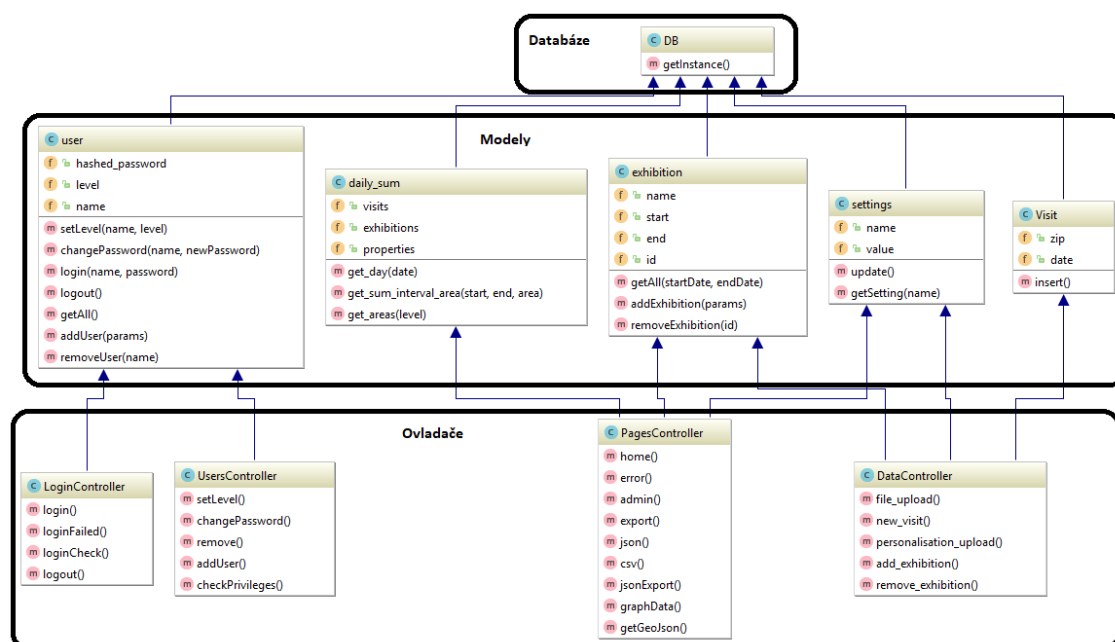
4 Implementace aplikace

Pro vývoj aplikace byly využity některé principy architektury model-view-controller (MVC). Veškerou komunikaci s databází provádí modely, avšak na rozdíl od klasického pojetí tohoto vzoru zde je logika aplikace obsluhována zejména v ovladačích (controllers), nikoliv v modelech. Na obrázku 4.1 je znázorněno základní schéma aplikace, tvořeno třemi vrstvami, a to databází, serverovou částí a klientem. Na obrázku 4.2 je znázorněn diagram tříd (class diagram) serverové části aplikace.

Aplikace je členěná do třech sémantických celků určených pro tři různé uživatele a veřejné REST API. Základním zobrazením a hlavním výstupem z aplikace je vizualizace obsahující přehledy návštěv instituce. Po přihlášení je uživatel s oprávněním zadávání dat přesunut na stránku určenou pro provoz na pokladně. Pokud má uživatel oprávnění administrátora, může se přesunout na stránku umožňující upravovat nastavení aplikace, spravovat uživatele, zadávat probíhající výstavy a hromadně vkládat údaje o návštěvách.



Obrázek 4.1: Schéma aplikace



Obrázek 4.2: Diagram tříd (class diagram) serverové části aplikace

4.1 Prezenční vrstva

Pro uživatelské rozhraní aplikace byl použit CSS framework Semantic UI ¹. Díky tomuto rozhraní získala aplikace moderní vzhled a uživatelsky přívětivé prostředí.

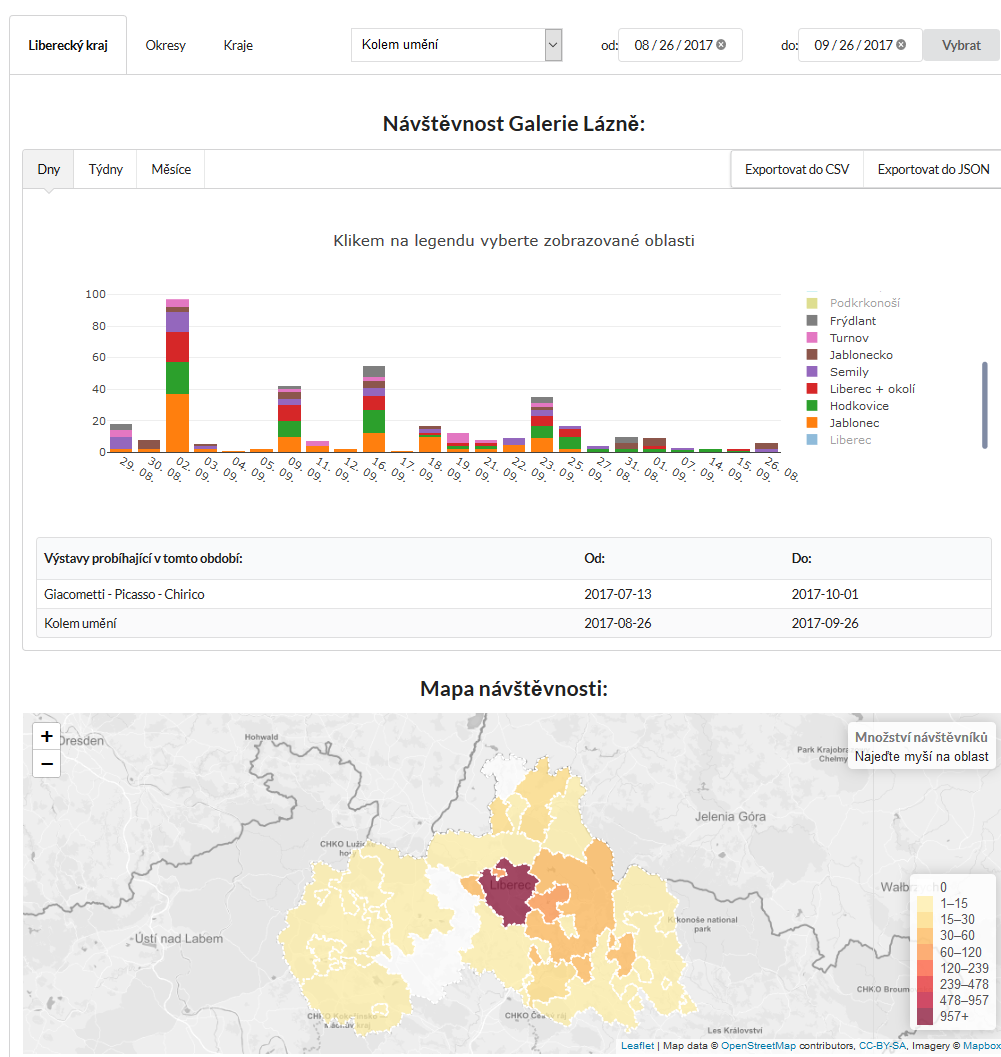
S pomocí Semantic UI bylo vytvořeno menu a byl upraven vzhled tlačítek a zadávacích polí. Tabulky pro správu uživatelů a výstav získaly svůj vzhled a aktivní prvky také díky tomuto frameworku.

¹<https://semantic-ui.com/>

4.1.1 Zobrazení dat

Při zobrazení dat je relativně mnoho ovládacích prvků, což uživateli zhoršuje schopnost orientace. V průběhu testování se ukázalo, že tedy bude nutné použít vizuální oddělovací prvky, které uživateli orientaci v aplikaci usnadní.

V tomto bylo opět nápomocné Semantic UI, z něhož bylo využito možnosti vytvořit „segment“ stránky a k němu připojit menu, aby bylo na první pohled zřejmé, k čemu se menu vztahuje a co ovlivňuje. Tyto segmenty lze do sebe i vnořovat.



Obrázek 4.3: Vizualizační část aplikace

Například nastavení dělení oblastí a nastavení počátečního a koncového datumu, které se vztahuje ke grafu, výpisu výstav i kartogramu, je zřetelně odděleno od nastavení seskupení časové osy, které se vztahuje pouze na graf a k němu patřící seznam výstav, export do souborů vychází z dat nastavených pro graf, proto je také umístěn v jednom celku s grafem. Toto dělení je zřejmé na obrázku 4.3.

4.1.2 Přizpůsobení

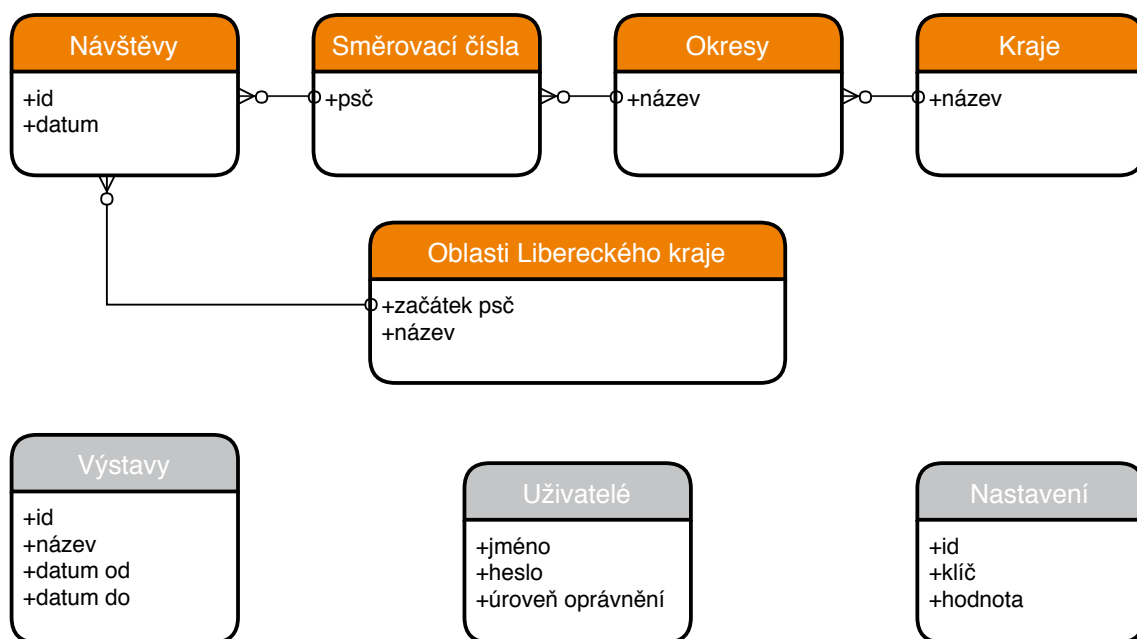
Vzhledem k tomu, že je aplikace připravená pro použití libovolnou institucí, nejen Galerii Lázně, obsahuje také možnost přizpůsobení vzhledu konkrétní instituci. V nastavení tedy může administrátor nahrát ikonu aplikace neboli „favicon“, logo instituce, zobrazované v levé části hlavního menu, a název instituce, jímž je nadepsaná stránka vizualizace. Nastavený název instituce je vidět na obrázku 4.3.

4.2 Datová vrstva

Databáze udržuje informace o jednotlivých návštěvách - jejich datумы a původ návštěvníků. Konkrétní návštěva je v databázi propojena s příslušným směrovacím číslem, to je propojeno s okresem, do něž náleží, a okres má určený kraj, jehož je podoblastí. Díky těmto propojením je možné návštěvníka přímo propojit s oblastí, ze které pochází s požadovanou přesností dělení. V databázi je také udržován seznam speciálních oblastí v rámci Libereckého kraje a příslušných začátků PSČ. V případě zvolení dělení podle těchto oblastí databáze propojí uživatele s oblastmi a počty návštěvníků sčítá podle nich. tyto vztahy jsou znázorněny v diagramu na obrázku 4.4.

Při vývoji aplikace byla snaha maximalizovat využití datové vrstvy, jelikož SQL databáze je systém umožňující nejefektivnější a nejrychlejší zpracování dat na serveru.

V databázi jsou také uloženy veškeré výstavy, nastavení a uživatelé. Vyjma loga a ikony aplikace je databáze jediným trvalým úložištěm dat.



Obrázek 4.4: Diagram entitně-relačního modelu databáze

Řešení neplatných PSČ

Některá PSČ byla v minulosti zrušena či změněna, a někteří návštěvníci si stále pamatují čísla původní a neví o jejich změně. Tato nesprávná pak předávají u pokladny. V případě, že by návštěvník řekl neplatné číslo, nebylo by jej možné přiřadit k oblasti původu. Takových čísel bylo v testovacích datech nalezeno mnoho a je zřejmé, že se budou objevovat i nadále. Zanedbat tuto skupinu návštěvníků by nebylo vhodné, proto bylo implementováno řešení tohoto problému.

Toto řešení vychází ze zkušenosti, že jsou poštovní směrovací čísla rozdělena podle geografické polohy a numericky blízká čísla si jsou blízká i geograficky. Zejména pak blízká čísla obvykle patří do stejných nadřazených celků. Proto při zadání směrovacího čísla do databáze není přímo vloženo dané číslo, ale je nalezeno nejbližší platné číslo a to je návštěvníkovi přiřazeno. Tento způsob řešení problému nemusí být vždy úplně přesný, ale informační hodnotu dat výrazně zvyšuje a ve většině případů je zpřesnění dostatečné.

Neplatná směrovací čísla jsou řešena až v datové vrstvě pomocí SQL DML příkazů, aby nebyla zásadně omezena rychlost ani při hromadném zadávání dat.

4.3 Aplikační vrstva

4.3.1 Správa uživatelů

Na základě úvahy nad způsobem používání aplikace byly vytvořeny tři úrovně uživatelských oprávnění. Každá vyšší úroveň má práva jako úroveň nižší, ale k nim má navíc přidána oprávnění další.

Základní úrovní uživatelských oprávnění je oprávnění pro zobrazení výstupů z aplikace neboli stránky se zobrazením dat. Vzhledem k tomu, že má být tato část aplikace jako veřejná, není pro její zobrazení uživatel dotázán na žádné přihlašovací údaje, ani se nemusí nikde registrovat, stránka je zobrazena ihned.

Veřejnost by však z pochopitelných důvodů neměla mít možnost zadávat žádné údaje do databáze. Pro zadávání údajů by měla mít oprávnění především obsluha pokladny. Ovšem zde, i když se může jednat o důvěryhodnou osobu, není zodpovědnost navíc za administrátorskou část aplikace žádoucí.

Proto je v aplikaci třetí úroveň oprávnění. Uživatel s tímto oprávněním má přístup do administrace, kde má možnost využívat veškerou její funkcionalitu. Má také možnost spravovat uživatelské účty. V uživatelském rozhraní pro správu uživatelů, vyobrazené na obrázku 4.5, může přidávat uživatele, měnit jejich oprávnění, měnit jejich přístupová hesla, nebo uživatelský účet odstranit.

Ačkoliv u webových aplikací bývá běžné, že uživatel o registraci zažádá sám vyplněním formuláře a tato žádost je vyřízena automaticky, zde bylo usouzeno, že bude vhodné, aby uživatelské účty spravoval administrátor této aplikace. Ten je totiž osobou zodpovědnou za správný chod aplikace, má o kolektivu ve své organizaci přehled a ví, komu jaká práva mají náležet. Zaměstnanci mají také možnost jej jednoduše kontaktovat v případě, že ztratí či zapomenou přihlašovací údaje ke svému účtu. Administrátor sám sobě nemůže odebrat oprávnění ani odstranit svůj

účet, aby nedošlo ke ztrátě přístupu k aplikaci. V takovém případě by bylo nutné přidat uživatele ručně do databáze, což je akce komplikovaná, protože v databázi nejsou uložena hesla přímo, ale šifrovaně. To není úkon určený pro uživatele ani administrátora této aplikace. Proto je této situaci z velké části zamezeno tím, že administrátorský účet může být smazán pouze jiným administrátorským účtem.

Správa uživatelů

Jméno	Administrátorská oprávnění	Heslo	Odstranit
administrátor		<div>Nové heslo</div> <div>Změnit</div>	
galerie	Odebrat	<div>Nové heslo</div> <div>Změnit</div>	✕
pokladna	Udělit	<div>Nové heslo</div> <div>Změnit</div>	✕
<div>Uživatelské jméno</div>	<input type="checkbox"/>	<div>Heslo</div>	<div> <div></div> <div>Přidat uživatele</div> </div>

Obrázek 4.5: Správa uživatelů v administrační části aplikace

Uživatel by se žádnou akcí v uživatelském rozhraní neměl dostat na místo, kam nemá přístup. Pokud by se však například stalo, že má odkaz na stránku s omezeným přístupem uloženou v záložkách prohlížeče a je již automaticky odhlášen, aplikace jej přesune na přihlašovací stránku.

Komunikace prohlížeče se serverem však zdaleka není omezena na akce zobrazené v adresním řádku, obsahuje i komunikaci uživateli úplně skrytou. Zde je již předpokládáno, že se o přístup pokouší vnější útočník, proto na žádost například o přidání uživatele do databáze, přidání záznamu nebo nahrání obrázku bez příslušných oprávnění server odpoví pouze chybovou stránkou.

Aplikace je připravená také na možnost mít více než tyto tři úrovně oprávnění, například je možné přidat účet „superadministrátora“, který by měl jako jediný možnost upravovat uživatelské účty administrátorů. Další možností je omezit nejméně oprávněného uživatele pouze na přihlášené uživatele, tedy umožnit administrátorovi kontrolovat přístup i do vizualizační části aplikace. Na tyto možnosti je aplikace připravena. Vyžadovala by pouze drobnou změnu v kódu kontrolerů.

REST API je ze správy účtů úplně vyjmuté, zde tedy není možné přístup omezit. To vychází z bezstavovosti RESTful služeb, jež by ani autentizaci umožňovat neměly (pokud se nejedná pouze o přístupový token v URL).

Přihlašovací obrazovka

Přihlašovací obrazovku (zobrazenou na obrázku 4.6) může uživatel vyvolat kliknutím na tlačítko v hlavním menu (Odhlásit se / Přihlásit se), kde je dotázán na uživatelské jméno a heslo. Po zadání údajů je vyhledán uživatelský účet v databázi a ověřeno heslo. V případě neplatných údajů je uživatel o této skutečnosti informován, v případě úspěšného přihlášení je uživatel přesunut do aplikace.

Přihlašovací formulář aplikace se zobrazí na vyžádání nebo při nedostatečných oprávněních pro zobrazení některé stránky.

Obrázek 4.6: Přihlašovací formulář aplikace

Bezpečnost hesel

Mnoho počítačových útoků míří na získání hesel uživatelů, protože uživatelé často používají stejná hesla pro více služeb. Z tohoto důvodu je důležité co nejvíce takový útok ztížit.

Na straně klienta je bezpečnost hesla zajištěna prohlížečem. Pro zadávání hesel je totiž použit HTML `<input>` tag s atributem specifickým pro hesla. Veškerá komunikace se serverem pak probíhá přes protokol TLS 1.2, což poskytuje dostatečnou bezpečnost přenosu dat, včetně jinak nezašifrovaných hesel.

Přijatá a z TLS komunikace rozšifrovaná hesla určená pro uložení do databáze server zašifruje pomocí PHP funkce `password_hash`. Tato funkce umožňuje zašifrovat heslo silným jednosměrným algoritmem. Vzhledem k tomu, že se kryptografie stále vyvíjí, je v aplikaci zvolena výchozí hašovací funkce, která zajistí, aby byl vždy použit algoritmus považovaný v dané době (resp. verzi PHP) za nejvhodnější. V současnosti tedy funkce využívá hašovací funkci Bcrypt. Pro ověření hesla je použita PHP funkce `password_verify`, která porovnává zadané heslo s odpovídajícím hashem z databáze.

4.3.2 Vkládání dat

Jednoduché uživatelské rozhraní pro zadávání dat umožňuje na přání galerie vložit jedno a více poštovních směrovacích čísel do databáze. Pokud totiž přijde do galerie skupina návštěvníků, obvykle pochází ze stejného místa. V takovém případě

je zásadní výhodou, když do databáze nemusí obsluha zadávat každého návštěvníka zvlášť. Tato stránka je také uzpůsobena pro ovládání klávesnicí, což usnadňuje a urychluje zadávání dat.

Pole pro zadávání PSČ a počtu lidí jsou nastavena na odpovídající formáty, tudíž uživateli není dovoleno zadat neplatné hodnoty a například při použití softwarové klávesnice jsou zobrazeny pouze číselné klávesy. Toto omezení však zajišťuje prohlížeč, proto na straně serveru je vstup znovu ověřován, aby nedošlo k zadání neplatných údajů či útoku na databázi.

Po odeslání zadávacího formuláře je uživateli ihned zobrazen prázdný formulář pro zadání další položky. Server mezitím nový údaj zpracuje, ověří a odešle do databáze, která jej přiřadí ke správnému PSČ a uloží.

4.3.3 Import dat ze souboru

Administrátor také v aplikaci může zadat již nasbíraná data například z tabulkového procesoru. Tato data je potřeba vkládat ve formátu CSV. Mohou být odděleny standardními oddělovači (čárka, středník). V záhlaví tabulky jsou očekávány datумы ve formátu rok-měsíc-den a pod nimi sloupce s poštovními směrovacími čísly, kde každému návštěvníkovi odpovídá jedna položka.

Tyto soubory mohou obsahovat položky pro jednotlivé dny nebo měsíce (jak je tomu v případě Oblastní galerie), roky či libovolné období. Z dat konkrétní instituce pak do tohoto formátu může data převádět například individuálně připravený Python skript.

Když je soubor administrátorem do aplikace nahrán, přesune se na server do dočasné složky, načte se pomocí PHP funkce `fgetcsv`, vytvoří se seznam dnů obsažených v souboru, a následně je iterováno přes všechna PSČ, která jsou postupně vkládána do databáze. Datová vrstva zde řeší přidávání návštěv s neplatnými poštovními směrovacími čísly.

Tímto způsobem byla vkládána testovací data do aplikace, čímž bylo ověřeno, že tento import funguje spolehlivě a dostatečně rychle.

4.3.4 Export dat do souboru

Export dat pro třetí strany (zejména pro zřizovatele) je umožněn v textových formátech JSON a CSV. Rozsah dní obsažených v exportovaném souboru se řídí nastavením exportu a je tedy možné tyto dny seskupit podle týdnů nebo měsíců, stejně jako u grafu. Také zde platí nastavení úrovně členění oblastí podle kraj, okresů nebo oblastí v Libereckém kraji určené v záhlaví exportu.

CSV

Pro CSV je na začátku souboru zapsáno označení pořadí bajtů (BOM - Byte Order Mark) - bajty `0xEF 0xBB 0xBF` neboli „UTF-8 signatura“. To bylo implementováno z toho důvodu, že některé starší verze Microsoft Excel správně nerozpoznaly UTF-8 kódování.

Poté je do záhlaví CSV souboru zapsáno záhlaví - pole „Oblast“ a jednotlivé dny, případně rozsah dnů. Na další řádky jsou zapsány v prvním sloupci názvy jednotlivých oblastí a v dalších sloupcích jsou součty návštěvníků pro danou oblast a dané datum/rozsah datumů. V tabulce 4.1 se nachází výňatek z dat exportovaných do CSV souboru. Data jsou seskupena po měsících a oblasti jsou rozdělené podle okresů.

Tabulka 4.1: Příklad exportovaných dat

Oblast	13. 08. - 13. 09. 2017	13. 09. - 13. 10. 2017
okres Liberec	2697	1332
okres Hlavní město Praha	1367	292
okres Jablonec nad Nisou	278	209
okres Semily	165	81
okres Česká Lípa	57	39
okres Mladá Boleslav	47	10
okres Děčín	60	7
okres Hradec Králové	44	8

JSON

Do souboru JSON jsou ukládány oblasti jako klíče, k nim jsou přiřazeny jednotlivé datumy a k nim náležící počty návštěvníků pro danou oblast.

Seřazení oblastí

Oblasti v exportovaných souborech obou formátů jsou seřazené sestupně podle celkového součtu počtu lidí, kteří v daném období přišli.

HTTP hlavička

Oba formáty mají v HTTP hlavičce nastaven odpovídající typ internetového média (Content-Type) jako `text/csv`, respektive `application/json`, a `Content-Disposition` jako `attachment`, aby se prohlížeč nepokoušel soubor otevřít, ale aby zrovna nabídl uživateli uložení na disk.

4.3.5 REST API

Pro automatizované předávání dat třetím stranám je dostupné REST API, které poskytuje informace o návštěvnosti, případně o dalších podrobnostech daného dne. Při volání API server klientovi pošle JSON soubor, zobrazený na obrázku 4.7, kde první položka je odkaz na předcházející den. Druhá položka obsahuje informace o datumu, ke kterému se data vztahují, počet lidí, kteří v daný den galerii navštívili, výstavy, které v ten den probíhaly, případně jestli se jednalo o den se vstupem zdarma. Tato poslední položka může být nahrazena jinou vlastností dne, případně může obsahovat více vlastností (například zkrácená otevírací doba, svátek, atd.)

```

1 {
2   "next": "?controller=pages&action=json&day=2018-05-10",
3   "stats": {
4     "visits": "12",
5     "properties": [
6       "free"
7     ],
8     "exhibitions": [
9       "Bydlet spolu",
10      "Alberto Kalach",
11      "Victorie Langer/Caramello"
12    ]
13  }
14 }

```

Obrázek 4.7: Ukázka JSON souboru získaného z REST API pro den 11. 5. 2018

Z definice REST API vychází, že by mělo být bezstavové, proto zde není žádná identifikace uživatele, ale na rozdíl od dalších částí aplikace zde ani nemůže být dodatečně implementována. REST API tedy vždy bude veřejné (pokud nebude úplně zakázáno).

Komunikace s REST API probíhá výhradně metodou GET. Příkazy jsou snadno generovatelné, a to buď sériovým přístupem (každý den obsahuje odkaz na den předchozí), nebo přímým zavoláním libovolného dne zadáním požadovaného datumu ve tvaru `rrrr-mm-dd`.

4.3.6 Graf

Graf ve vizualizační části je sloupcový graf, kde osa x (vodorovná) znázorňuje čas a osa y (svislá) počet návštěvníků. V grafu může uživatel ovlivnit parametry rozsahu časové osy, rozlišení časové osy a rozlišení svislé osy.

Rozlišení svislé osy je určeno globálním parametrem v záhlaví vizualizační stránky a je společné pro graf, export i mapu; jedná se o dělení na oblasti v Libereckém kraji a na okresy a kraje v rámci celé České republiky.

Rozsah časové osy je taktéž společný pro celou vizualizaci a je možno jej nastavit buď přímo, pomocí dvou selektorů datumu, nebo podle probíhající výstavy. Pokud tedy uživatel chce vidět, jak chodili do galerie návštěvníci během konkrétní výstavy, zda-li například přišlo nejvíce lidí na vernisáž, zvolí si v záhlaví výstavu, o níž má zájem, a statistiky této výstavy se mu zobrazí.

Ve chvíli, kdy uživatel změní parametry, je odeslán požadavek na server, uživateli se zobrazí ikona načítání, zatímco server data připravuje a prohlížeč je v běhovém prostředí javascriptu zpracovává.

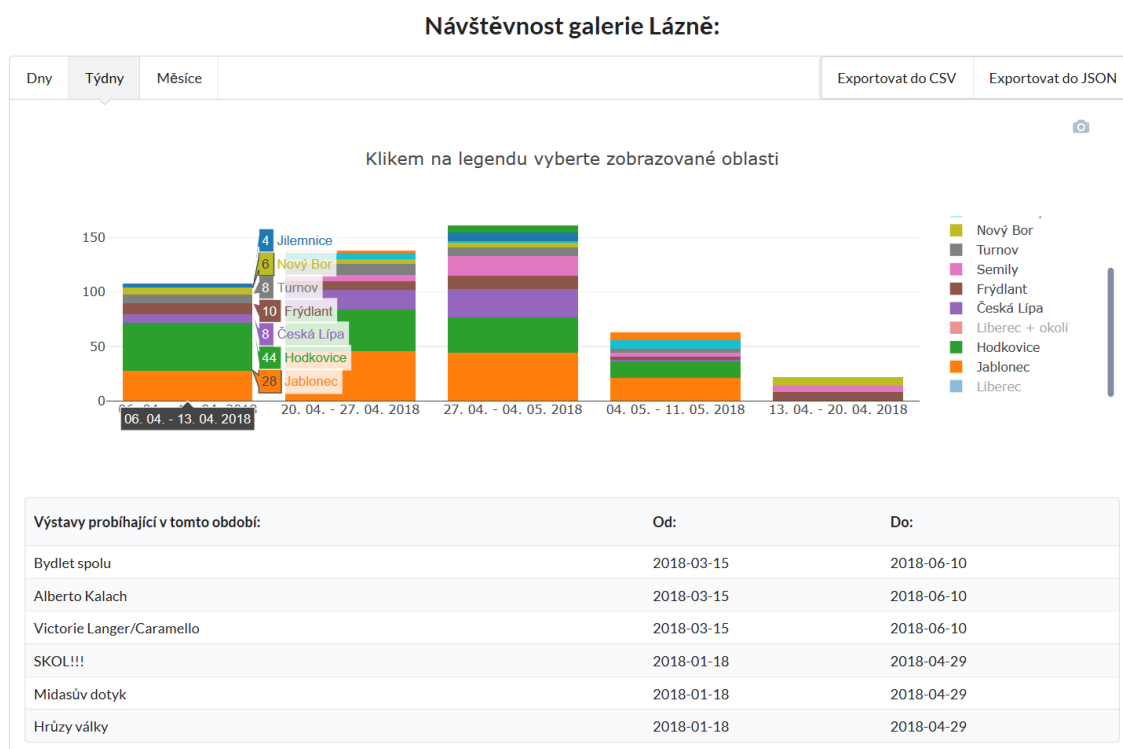
Server nejdříve získá veškeré aktuální oblasti pro zvolenou úroveň dělení. Poté iteruje přes oblasti a postupně získává součty návštěvníků pro jednotlivé dny, případně týdny nebo měsíce, podle nastavených parametrů. Zároveň ukládá data o tom, k jakým dnům se interval vztahuje, aby bylo možné tuto informaci uživateli zobrazit jako popisek časové osy.

Když server získá všechny požadované informace, setřídí je pomocí lambda funkce, která porovnává součet všech návštěvníků v daném období pro každou oblast, aby byl graf přehledný a nejvýznamnější oblasti byly ve sloupci vždy vespod.

Nakonec server získá z databáze seznam názvů a datumů všech výstav, které alespoň zčásti zasahovaly do zvoleného období.

Všechna tato data server zakomponuje do jednoho JSON souboru a pošle zpět prohlížeči. Vzhledem k tomu, že je použita metoda AJAX, prohlížeč nemusí obnovovat celou stránku, ale pouze relevantní části. V tomto případě je překreslen pouze graf a k němu se vztahující seznam výstav.

Pro zobrazení grafu je použita knihovna Plotly.js, postavená na základu knihovny pro zpracování a vizualizaci dat D3.js. JSON soubor poslaný serverem je připravený přímo pro vložení do této knihovny. Zde je tedy pouze provedeno zjednodušení uživatelského rozhraní, aby uživatel nebyl maten zbytečně velkým množstvím ovládacích prvků.



Obrázek 4.8: Příklad vykreslení grafu

Ve zobrazeném grafu uživatel vidí rozložení návštěvnosti. Je zobrazena legenda pro barevně rozlišené oblasti a při najetí kurzorem na sloupec grafu uživatel vidí přesné počty návštěvníků z každé oblasti, ze které v daném období návštěvníci přišli.

Pokud uživatel nepožaduje zobrazit všechny oblasti, například pokud chce z grafu vyjmout data pro samotný Liberec, aby získal přehlednější informace o návštěvnících z okolních oblastí, může kliknout na položku v legendě a odpovídající oblasti se z grafu skryjí. Tímto si může uživatel zvolit zobrazení libovolných oblastí.

Pod grafem je vypsána tabulka výstav, které probíhaly ve vybraném období, včetně datumů, kdy začínaly a končily.

Na obrázku 4.8 je vidět graf s rozlišením časové osy na týdny, dělením podle oblastí v Libereckém kraji a vybranými oblastmi mimo Liberec. Součástí grafu je i seznam výstav probíhajících v daném období a možnost exportu vybraných dat.

4.3.7 Mapa

Druhou částí vizualizace je kartogram. Kartogram (anglicky choropleth map) je mapa, kde je barevně vyjádřena intenzita sledovaného jevu na daném území. Kartogram vyjadřuje dohromady hodnoty určitých oblastí, na rozdíl od teplotní mapy, která zobrazuje hodnoty spojitě po celé ploše, bez rozlišení konkrétních oblastí.

Parametry této mapy se určují taktéž v horním panelu vizualizační stránky. Vzhledem k tomu, že na této mapě ale není vyobrazen rozměr času, nemají na mapu vliv parametry rozlišení času neboli seskupení podle dnů, týdnů, či měsíců.

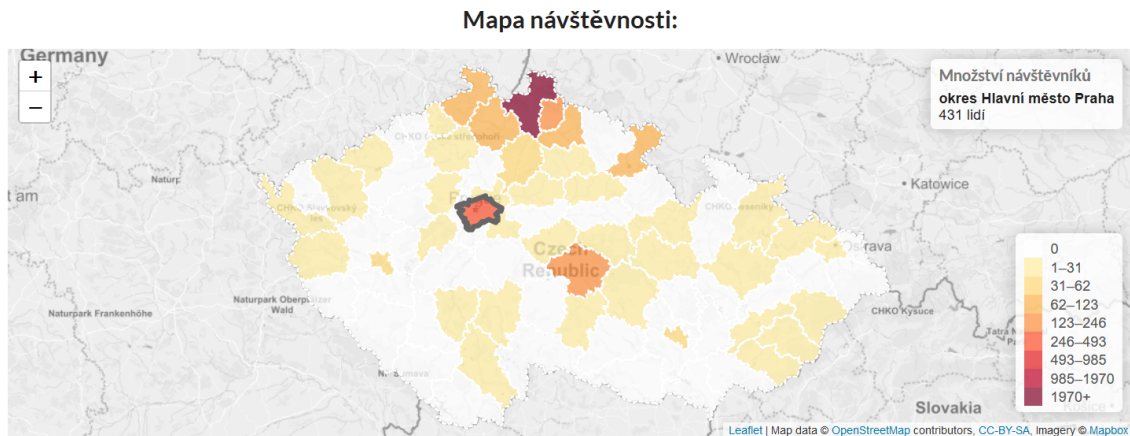
Při načtení mapy či při změně parametrů je poslán AJAX požadavek na server a uživateli je opět zobrazena ikona symbolizující načítání.

Server mezitím načte z disku připravené GeoJSON soubory a pro každou oblast se dotáže databáze, kolik lidí tuto oblast za dané období navštívilo. Tato informace je pak přidána do GeoJSON souborů, které jsou pak předány klientovi.

Když prohlížeč dostane upravené GeoJSON soubory, předá je knihovně leaflet.js pro zpracování. Při prvním načtení tato stránka načte mapu z projektu OpenStreetMap, při opakovaných načteních jsou již pouze odstraněny vrstvy, které se budou překreslovat, mapa nikoliv. Podle zvoleného dělení oblastí se také nastaví střed a přiblížení mapy, aby mapa pokrývala požadovanou oblast.

Skript připravující mapu načte soubor GeoJSON, podle kterého knihovna leaflet.js vytyčí hranice oblastí. Poté jsou spočítány rozsahy pro barvy, kterými se má mapa vybarvit. Vzhledem k tomu, že rozdíly mezi počty návštěvníků nejsou rovnoměrné, ale naprostá většina návštěvníků pochází z jedné oblasti, a ostatní oblasti jsou původem mnohem menšího množství návštěvníků, bylo vybráno pro měřítko logaritmické rozložení. Hodnoty jsou vypočítány od celkového počtu návštěvníků za období ve všech oblastech, od této hodnoty je pak odvozeno 9 rozsahů pro škálu barev od tmavě červené po bílou. Tmavě červená zde vyjadřuje více jak polovinu všech návštěvníků, světlejší čtvrtinu všech návštěvníků, ještě světlejší osminu a takto škála pokračuje až do bílé. Podle této barevné škály jsou nabarveny oblasti v grafu a škála s rozsahy barev je pak zobrazena v legendě. Vzhledem k tomu, že rozsahy barev nejsou celá čísla, jsou před zobrazením v legendě zaokrouhleny, aby aplikace byla uživatelsky přívětivější.

Když uživatel kurzorem namíří na určitou oblast, v pravém horním rohu se zobrazí informace o oblasti - její název a počet návštěvníků - aby uživatel mohl přehledně zjistit přesné hodnoty. Při kliknutí na oblast se mapa přiblíží, případně oddálí, a zaměří na danou oblast.



Obrázek 4.9: Kartogram s rozlišením oblastí na okresy a vybraným okresem Hlavní město Praha, zobrazující počet návštěvníků pocházejících z tohoto okresu

4.3.8 Rychlost aplikace

Ačkoliv aplikační vrstva neobsahuje žádné příliš časově složité algoritmy, při běhu na testovacím serveru trvá načtení vizualizační stránky dlouhou dobu, řádově jednotky vteřin, v závislosti na zvolených parametrech. Toto je způsobeno velkým množstvím zpracovávaných dat aplikační vrstvou. Virtuální testovací server, na kterém je aplikace spuštěna, je součástí ekonomického sdíleného webhostingu, proto zde není pro aplikaci přidělen dostatek procesorového času ani paměti. Zpracování některých úloh také virtuální server nezvládne paralelně, jelikož má k dispozici velmi malé množství operační paměti. Virtuální server samotný má k dispozici 2^{28} bajtů (256 MB) operační paměti, ale prakticky dostupných pro aplikaci je pouze několik desítek MB. Toto omezení činí problém zejména při zpracování GeoJSON souborů, které obsahují velké množství dat.

Před zveřejněním proto bude aplikace přesunuta na jiný server, kde bude mít k dispozici více systémových prostředků. Možnou optimalizací do budoucna také zůstává zjednodušení GeoJSON dat, aby neobsahovala tolik podrobné detaily a díky tomu bylo množství zpracovávaných dat menší. Další možností by bylo vybraných agregovaných dat do samotné databáze.

4.3.9 Bezpečnost aplikace

Vzhledem k tomu, že aplikace na základě uživatelských vstupů (ale i obecně HTTP požadavků) generuje dotazy do databáze, je potřeba tyto vstupy ošetřit proti SQL injection útokům. Například přihlašovací formulář vyhledává uživatelské jméno přímo v databázi. Tato část aplikace je veřejně dostupná a během vývoje bylo zjištěno, že zde může útočník snadno spustit svůj SQL kód, který vloží do pole pro přihlašovací jméno.

Kdyby útočník nějakým způsobem získal přístup do administrace (což by mohlo být způsobeno lidskou chybou, chybou v aplikaci, nebo úspěšně provedeným CSRF

útokem), mohl by zvýšit způsobenou škodu také například provedením XSS útoku neboli vložením skriptu například do záhlaví vizualizační stránky, jenž je nastavitelné v administraci.

Proti těmto útokům je v aplikaci použita knihovna HTML Purifier. Ta z uživatelských vstupů a z dalších HTTP požadavků odstraní potenciálně nebezpečné znaky, které by mohly umožnit útočníkovi vniknutí do systému či způsobení škod v systému.

Není prakticky možné, aby byl systém naprosto nedobytný. Ale díky kombinaci ošetření vstupů, využívání převážně HTTP POST požadavků pro kritické operace, zabezpečení přenosu protokolem TLS a dalších bezpečnostních opatření je vektor útoku značně zúžen. Pro potenciálního útočníka by tedy nemělo být snadné způsobit v aplikaci škodu.

4.3.10 Kompatibilita aplikace

Prohlížeče, na kterých byla testována kompatibilita a správná funkčnost aplikace, byly vybrány jako zástupci významných renderovacích jader a vzhledem k odlišnému chování na různých operačních systémech byly otestovány i na zástupcích významných operačních systémů. Funkčnost aplikace na mobilních telefonech testována nebyla, jelikož je aplikace mířena především pro použití na počítači.

Výjimkou je část pro zadávání dat, která byla vytvořena s důrazem na maximální kompatibilitu, aby bylo možné využít na pokladně libovolné terminály. Tato stránka ke své funkci nepotřebuje javascript, ani žádnou funkcionalitu nově uvedenou v HTML5, proto je možné ji zobrazit a používat na téměř libovolných zařízeních s webovým prohlížečem. Na obrázku 4.10 je vidět testování části aplikace pro zadávání dat na vestavěném prohlížeči dnes již velmi zastaralého zařízení Sony Ericsson J10i2 ELM, a i zde aplikace nabízí plnou funkcionalitu.



Obrázek 4.10: Testování zadávání dat na starších zařízeních

Správná funkčnost aplikace je otestována na prohlížečích v tabulce 4.2.

Tabulka 4.2: Podporované prohlížeče a platformy

Prohlížeč	Verze	Platforma
Mozilla Firefox	59.0.2	Linux Mint 18.1 Serena
Chromium	65.0.3325.181	Linux Mint 18.1 Serena
Mozilla Firefox	60.0b16	Windows 10
Chromium	68.0.3416.0	Windows 10
Microsoft Edge	41.16299.3701.0	Windows 10
Apple Safari	10.0.1	macOS 10.12 Sierra

Ve všech zde zmíněných testovaných prohlížečích aplikace funguje bez problémů. Kvůli použití funkcí HTML5, které vzhled některých prvků (zejména selektorů datumu a drop-down menu) nechávají na rozhodnutí prohlížeče, se uživatelské prostředí po vizuální stránce mezi prohlížeči a platformami mírně mění. To však nijak neomezuje funkčnost, ale ani vzhled a použitelnost aplikace.

Na některých počítačích stále ještě používaný Internet Explorer v seznamu podporovaných prohlížečů uveden není, jelikož zde není možné zprovoznit jednu z nejdůležitějších částí vizualizace, knihovnu Plotly.js. Zadávání dat a administrace zde však funkční jsou.

Pro testování aplikace byla použita data Galerie Lázně a aplikace byla průběžně testována, kompletní akceptační testy se předpokládají před zavedením aplikace do provozu.

5 Závěr

V rámci práce byly prozkoumány možnosti pasivního i aktivního sběru dat o návštěvnicích a návštěvách v galeriích. Pro vytvoření aplikace se zatím použil osvědčený dosavadní způsob sběru, a to že jsou návštěvníci u vstupu dotazováni na poštovní směrovací čísla jejich trvalého pobytu.

Avšak způsob zadávání a uchovávání těchto dat byl zásadně změněn. Byla vytvořena webová aplikace, umožňující konzistentní, snadné, rychlé a bezpečné zadávání a uložení informací o původu návštěvníků. Byla také vytvořena vizualizace těchto dat ve formě grafů a map, dále možnost jejich exportování do textových formátů CSV a JSON a REST API pro automatizované čtení dat.

Od původního záměru dělení oblastí přímo podle směrovacích čísel, pro které byly vyzkoušeny i některé algoritmy, bylo z důvodu přílišné podrobnosti po dohodě s galerií upuštěno.

Uživatel si tedy ve vizualizaci může vybrat mezi dělením České republiky podle krajů a okresů a dělením Libereckého kraje na vybrané oblasti (například Liberec, Jablonec, Hodkovice, Semily, Česká Lípa, Podkrkonoší, ...). Uživatel si vybere rozsah datumů, mezi kterými chce zobrazit statistiky, a aplikace mu vygeneruje stránku s grafem a kartogramem, ze které se dozví požadované informace.

Značná část vývoje aplikace spočívala v přípravě dat. Vzhledem k tomu, že potřebná geografická data v mapových API dostupná nebyla, bylo nutné najít způsoby, jak tato data získat. Pro tento účel byly vytvořeny skripty v jazyku Python, které využívají vybraná veřejně dostupná data (například z České pošty a databáze OpenStreetMap), kombinují je a vytváří z nich data potřebná pro využití v aplikaci.

Momentálně probíhá jednání s Galerií Lázně o způsobu uvedení aplikace do provozu. Je také připravován server, který by vyšší zátěž v produkčním nasazení unesl. Pokud půjde přesun a jednání podle současných předpokladů, bude moci galerie začít využívat systém do podzimu roku 2018.

Literatura

- [1] How much does an RFID tag cost today? *RFID Journal* [online] Emerald Expositions, LLC, 2017 [cit. 2017-10-03]. Dostupné z: <https://www.rfidjournal.com/faq/show?85>
- [2] Pyroelektrický senzor PYRO. *Eco-Counter* [PDF] Brno: o.p.s. partnerství [cit. 2017-10-05]. Dostupné z: <http://www.scitace.cz/Scitace/files/a8/a8423b4c-e0ea-4ec6-9283-6bd2861aa09f.pdf>
- [3] People counting mat. *Instant Counting* [PDF] Montreal: Eco-Counter [cit. 2017-10-06]. Dostupné z: http://www.instantcounting.com/files/INSTANTCNT_45547_1415284389578_depliant_ic_e_1.2_092014_impo_bd.pdf
- [4] Mezinárodní kodex ICC/ESOMAR. *ICC/ESOMAR* [PDF] ICC/ESOMAR, 2016 [cit. 2017-10-20]. Dostupné z: https://www.esomar.org/uploads/public/knowledge-and-standards/codes-and-guidelines/ICCESOMAR_Code_Czech_.pdf
- [5] [RICHARDSON, Leonard a Sam. RUBY] *RESTful web services*. Farnham: O'Reilly, 2007. ISBN 978-0596529260.
- [6] [KRAFT, Jakub] Re: Hledání podle PSČ. *Seznam.cz Nápověda* [online] Seznam.cz, a.s., 2017-11-29 [cit. 2018-01-07]. Dostupné z: <https://napoveda.seznam.cz/forum/viewtopic.php?f=31&t=30573>
- [7] [BREITENMOSER, Andreas, Mac SCHWAGER, Jean-Claude METZGER, Roland SIEGWART a Daniela RUS] Voronoi coverage of non-convex environments with a group of networked robots. In: 2010 IEEE International Conference on Robotics and Automation [online]. IEEE, 2010, 2010, s. 4982-4989 [cit. 2018-03-25]. DOI: 10.1109/ROBOT.2010.5509696. ISBN 978-1-4244-5038-1. Dostupné z: <http://ieeexplore.ieee.org/document/5509696/>