



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**WEBOVÁ APLIKACE PRO POŘIZOVÁNÍ NOVÝCH  
ZÁBĚRŮ STARÝCH POHLEDNIC**

WEB APP FOR SHOOTING NEW PHOTOS OF ANCIENT POSTCARDS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MARTIN ŠTAINER**

**VEDOUcí PRÁCE**

SUPERVISOR

**prof. Ing. ADAM HEROUT, Ph.D.**

BRNO 2017

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

**Zadání bakalářské práce**

Řešitel: **Štainer Martin**

Obor: Informační technologie

Téma: **Webová aplikace pro pořizování nových záběrů starých pohlednic  
Web App for Shooting New Photos of Ancient Postcards**

Kategorie: Uživatelská rozhraní

**Pokyny:**

1. Seznamte se s problematikou "re-photography" a pořizování nových záběrů starých pohlednic a fotografií.
2. Definujte požadavky na funkčnost webového systému pro správu starých a nových fotografií.
3. Prototypujte prvky uživatelského rozhraní a funkčnosti řešené webové aplikace. Testujte je na uživatelích.
4. Vytvořte webovou aplikaci pro podporu pořizování nových záběrů starých fotografií.
5. Získejte a vložte do aplikace dostatečné množství relevantních dat.
6. Testujte vytvořenou aplikaci na uživatelích.
7. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

**Literatura:**

- dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2, značné rozpracování bodů 3 až 5.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Herout Adam, prof. Ing., Ph.D.,** UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
612 05 Brno, Božetěchova 2



---

doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Tato práce zabývá problematikou repeat photography. Popisuje návrh a implementaci webové aplikace určenou k pořizování nových záběrů starých pohlednic. V práci jsou popsány případy užití, datový model, jednotlivé problémy řešené při implementaci a iterativní testování s následným vylepšením aplikace.

## Abstract

Thesis is concentrated on the problematics with repeat photography. It describes a suggestion of web implementation and implementation it self. It is determined for new shot of old postcards. The process also describes problems involved with implementation and iterative testing with users and following improvement of app.

## Klíčová slova

PHP, HTML, Bootstrap, CSS, Nette, MySQL, repeat photography, fotografie, fotografování staré pohlednice, GUI

## Keywords

PHP, HTML, Bootstrap, CSS, Nette, MySQL, repeat photography, photography, images, ancient postcards, GUI

## Citace

ŠTAINER, Martin. *Webová aplikace pro pořizování nových záběrů starých pohlednic*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Herout Adam.

# Webová aplikace pro pořizování nových záběrů starých pohlednic

## Prohlášení

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

.....

Martin Štainer  
18. května 2017

## Poděkování

Děkuji svému vedoucímu prof. Ing. Adamu Heroutovi, Ph.D za cenné rady a profesionální vedení, své rodině za nekonečnou podporu, svému kolegovi Martinu Buchtovi za spolupráci při propojování mobilní aplikace s webovou aplikací a panu Ing. Petru Ivovi za poskytnutí velké databáze historických pohlednic.

# Obsah

<b>1 Úvod</b>	<b>2</b>
<b>2 Problematika repeat photography</b>	<b>3</b>
2.1 Faktory ovlivňující kvalitu znovupořizovaných záběrů pohlednic . . . . .	4
2.2 Uplatnění v různých oborech vědy . . . . .	4
2.3 Srovnání existujících aplikací . . . . .	5
<b>3 Použité technologie a nástroje k realizaci práce</b>	<b>8</b>
3.1 Technologie použité pro vývoj . . . . .	8
3.2 Front-end technologie . . . . .	8
3.3 Back-end technologie . . . . .	11
<b>4 Návrh a implementace aplikace</b>	<b>16</b>
4.1 Případy užití . . . . .	16
4.2 Návrh databáze . . . . .	16
4.3 Návrh uživatelského rozhraní . . . . .	18
4.4 Struktura aplikace . . . . .	18
4.5 Implementace databázové vrstvy . . . . .	19
4.6 Implementace mapy . . . . .	19
4.7 Vlastní jednoduchý grafický editor . . . . .	20
4.8 REST API . . . . .	21
4.9 Ukládání obrázků . . . . .	22
4.10 Autorizace a autentizace . . . . .	23
<b>5 Iterativní testování prototypu a postupné vylepšování</b>	<b>24</b>
5.1 Prototyp . . . . .	24
5.2 Verze 1.0 . . . . .	26
5.3 Verze 1.1 . . . . .	28
<b>6 Závěr</b>	<b>29</b>
<b>Přílohy</b>	<b>32</b>
<b>A Obsah příloženého paměťového média</b>	<b>33</b>

# Kapitola 1

## Úvod

Tato práce se zabývá problematikou repeat photography, výsledkem je webová aplikace, která umožňuje uživatelům nahrávat staré pohlednice, ukládat současné snímky pohlednic a srovnávat je. Nechybí uživatelské profily a sdílení na populárních sociálních sítích. Webová aplikace poskytuje rozhraní pro distribuci dat mobilní aplikaci. Tato distribuce je implementována technologií REST.

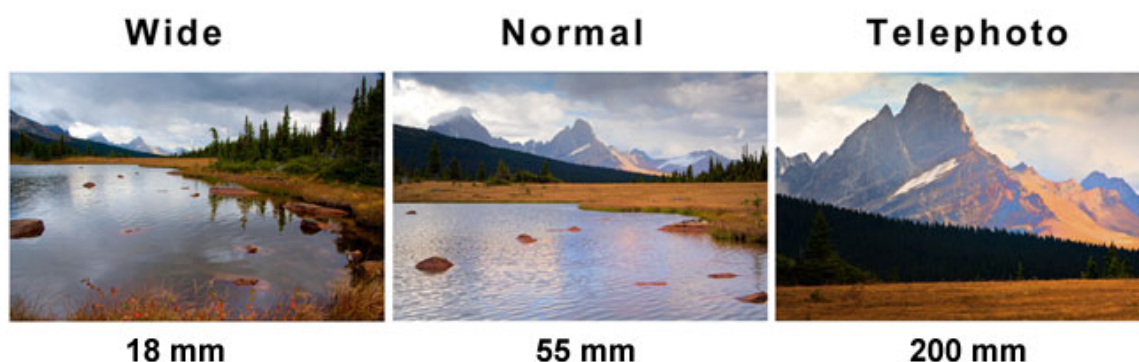
Aplikace zvyšuje zájem o kulturní dění, podporuje turismus, motivuje k poznání okolí a nachází uplatnění i v různých odvětvích vědy, nejen sociálních ale i přírodních. Typickými příklady jsou aplikace v geografii, populační ekologii nebo ve sledování street art. V sociálních vědách mohou fotografie posloužit při studování změn chování v reklamách apod. V přírodních vědách mohou pomoci při sledování změn vegetace, pozorování vlivu globálního oteplování atd.

V tomto dokumentu popisují problematiku repeat photography [2](#), srovnání existujících aplikací [2.3](#), přehled technologií [3](#), návrh a implementace aplikace [4](#), iterativní testování aplikace s následným zlepšením [5](#) a v závěru shrnuji tuto práci s možnostmi dalšího rozvoje a dosažené výsledky.

## Kapitola 2

# Problematika repeat photography

Pojmem repeat photography se rozumí vyfocení místa s časovou mezerou. Tato časová me- zera je obvykle vnímána jako předtím a teď. Výsledné srovnání zobrazuje změnu mezi dvěma snímky. Preciznost a kvalita znovupořádaného snímku je ovlivněna schopností fotografa a vhodně zvoleným aparátem, přesněji objektivem. Aby bylo výsledné srovnání co nejvíce au- tentické, je důležité vybrat vhodnou historickou pohlednici. Ta musí mít dostatečné rozlišení a kvalitu, aby bylo možné rozeznat objekty s pozadím.



Obrázek 2.1: Rozdíl zabírané plochy ze stejného místa při použití tří rozdílných objektivů. Obrázek vlevo ukazuje použití širokoúhlého objektivu. Na prostředním obrázku je záběr při použití normální objektivu. Pravý obrázek ukazuje použití teleobjektivu.

S příchodem digitálních technologií a kvalitních softwarů již není nutné, aby fotografie byla vyfocena naprosto přesně. Díky úpravám pomocí grafických editorů lze vyfocenou fo- tografii vylepšit na požadovaný výsledek. Aby fotografie mohla být upravena, musí program podporovat více vrstev. Novou fotografii je možné překrýt přes pohlednici a provádět s ní operace rotování, ořezání, zvětšení, zmenšení nebo posun.

Uchování fotografií (ať už v digitální nebo papírové podobě) se doporučuje včetně její předlohy, přesné lokace (zeměpisná výška a šířka nebo UTM dat<sup>1</sup>) a dalších metadat, které mohou pomoci při studování vizuálních změn v různých oborech vědy. Oproti tradičnímu uchování dat nepodléhá digitální uložení opotřebení [1].

<sup>1</sup>Universal Transverse Mercator coordinate system <https://www.uwgb.edu/dutchs/FieldMethods/UTMSystem.htm>

## 2.1 Faktory ovlivňující kvalitu znovupořádaných záběrů pohlednic

Nejdůležitějším faktorem při repeat photography je nalezení počátečního bodu, odkud byl záběr pořízen. Výběr bodu je komplikován, pokud je okolí místa nebo samotné místo rekonstruováno. Mnoho historických pohlednic není možné vyfotit, a to z důvodu nepřístupnosti do míst, odkud byly vyfoceny. Především se jedná o fotografie z vyšších míst např. střech nebo balkonů. Počasí, roční období a denní doba taktéž komplikují pořízení snímku.

## 2.2 Uplatnění v různých oborech vědy

Nejjednodušší strategie pro sledování změn v sociálním chování je pomocí aplikací zabývajících se rephotography nebo skrz vlastní dokumentované materiály. Příkladem může být následující jednoduchá studie. Obrázky 2.2 ukazují města s 15 roční mezerou. Na první fotografii lze vidět rozvíjící se malé město. Jsou tu obchody, škola a benzínová pumpa. Na druhé fotografii pořízené o 15 let později si lze všimnout, že obchody jsou zavřeny a ulice vypadá zchátrale. Z toho jednoduchého poznatku lze vyvodit, že město přešlo k úpadku. Důležité je umět změny správně interpretovat – u města mohlo být postaveno obchodní centrum a obchody pouze zbankrotovali. Z tohoto důvodu jsou vizuální studie různě doplňovány o rozhovory apod [2].



Obrázek 2.2: Změna měst s 15 roční mezerou. Levý obrázek ukazuje města v rozvoji, jsou otevřené oba obchody. Na pravém obrázku si lze všimnout města v úpadku, obchody jsou zavřeny.

Dalším příkladem může být projekt repeat photography USGS [3]. Ten začal roku 1997 systematickým vyhledáváním fotografií ledovců z národního parku Glacier. Od roku 1997 bylo vyfoceno více než šedesát fotografií sedmnácti různých ledovců. Třináct z nich vykazovaly výrazný pokles a z intenzivněji studovaných ledovců se ukázalo, že jsou jen třetinové oproti jejich odhadované maximální velikosti. Další fotografie zaznamenávají dramatickou změnu vegetace v reakci na klimatické změny. Fotografie mohou ukázat další jemnější detaily, než je změna velikosti ledovců z důsledku globálního oteplování.

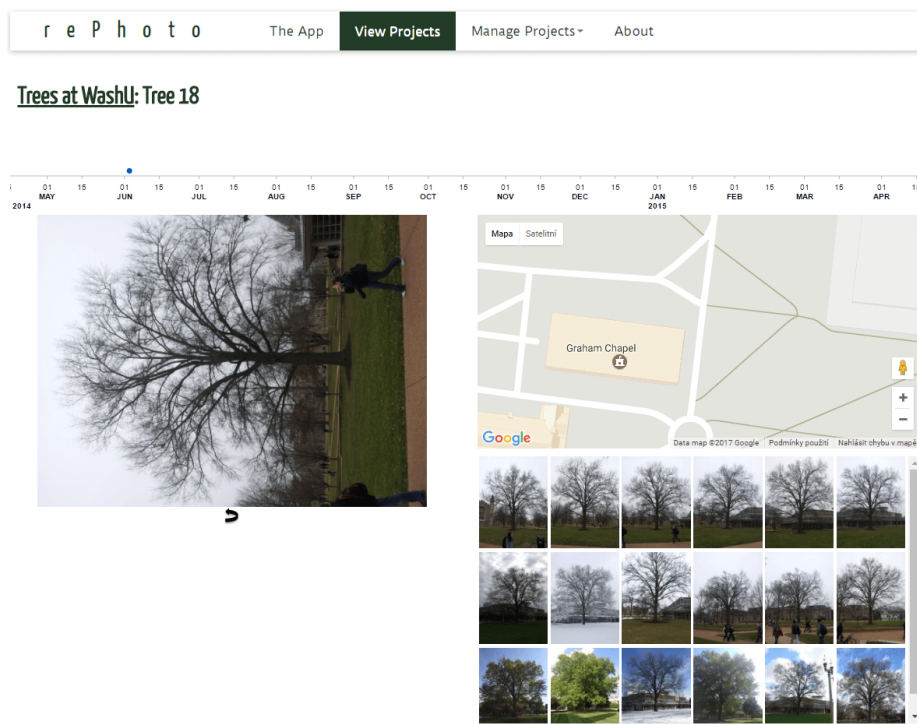


## 2.3 Srovnání existujících aplikací

V dnešní době existuje několik projektů, které se zabývají problematikou repeat photography nebo sbíráním historických fotografií a pohlednic. Všechny se liší svým zpracováním a především oblastí, které zabírají. Mnoho z těchto projektů jsou blogy, webové stránky, které slouží pouze k prezentaci vlastně pořízených snímků. Aplikací, které by umožnily běžnému návštěvníkovi se zapojit a sdílet nové snímky nebo staré fotografie, je méně.

### rePhoto

rePhoto<sup>2</sup> je projektem tří univerzit (Washington University, University of Vermont and the University of California, San Diego), který se skládá z mobilní aplikace pro operační systémy Android, iOS a webové aplikace. Mobilní aplikace slouží k pořizování nových záběrů s kamerou mobilního telefonu, která umožňuje prolnutí zabíraného místa s předlohou. U předlohy dovoluje změnu průhlednosti a obsahuje několik šablon, které různě rozdělují obrazovku pro lepší záběr foceného místa. Webová aplikace slouží k prohlížení fotografií na časové ose a ke správě projektů. Projektem může být cokoliv, např. člověk vyfocený před a po přísné dietě. Uživatelé do těchto projektů mohou libovolně přispívat svými výtvy.

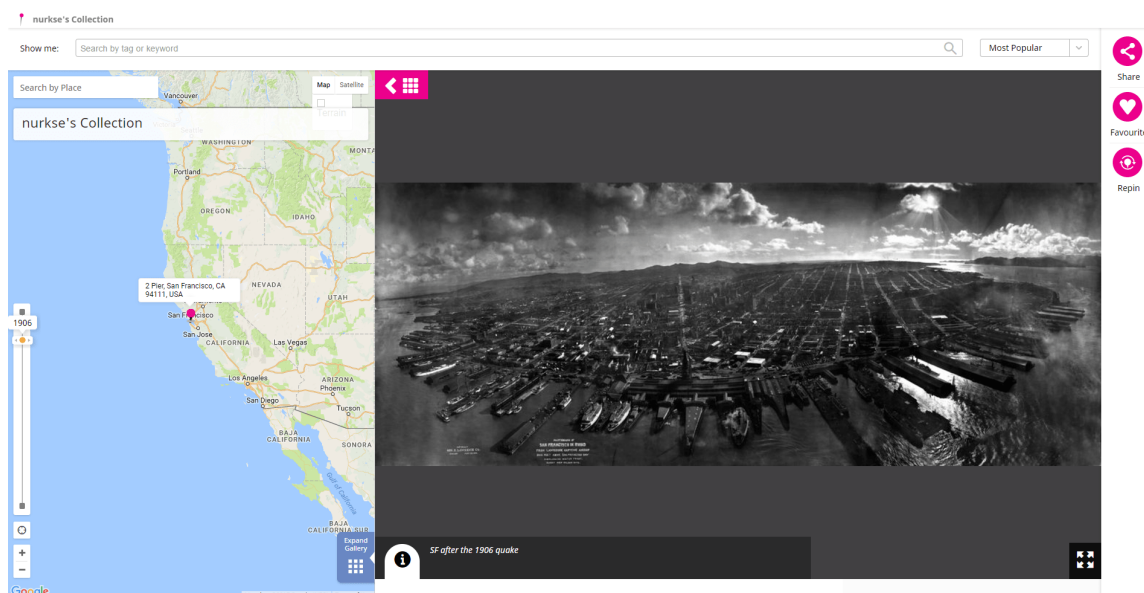


Obrázek 2.3: Ukázka prohlížení fotografií projektu rePhoto. Webová aplikace má jednoduchý design a neumožňuje fotografie srovnávat. Fotografie jsou řazeny v časové ose.

<sup>2</sup>rePhoto – <http://www.projectrephoto.com>

## Historypin

Historypin<sup>3</sup> je způsob shromáždění lidí za účelem oslavení lokální historie. Skládá se z archivu, vzájemně se podporující komunity a společného přístupu k zapojení se do místní historie. [4] Projekt se nezabývá problematikou repeat photography, avšak obsahuje mnoho historických pohlednic a fotografií z celého světa. Základem této aplikace je mapa, která zobrazuje fotografie s jejich přesnou lokací. Při prohlížení fotografií je možné zvolit kolekci, která napomáhá sdružovat snímky se společným tématem.



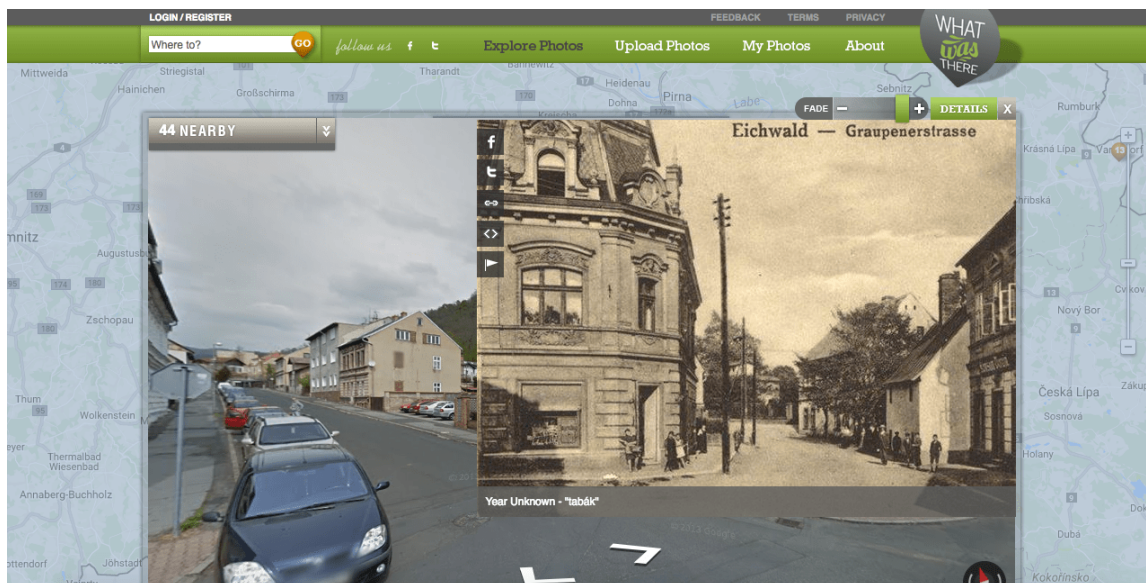
Obrázek 2.4: Ukázka prohlížení fotografií projektu historypin. Na obrázku si lze všimnout zdůraznění polohy na mapě a zobrazení fotografie ve velké velikosti. V horním panelu se vyskytuje možnost vyhledávání a prohlížení kolekcí.

## WhatWasThere

WhatWasThere<sup>4</sup> je projekt zobrazující historické fotografie, které mohou být proluté s fotografiemi z Google Street, ale neumožňuje vyfotit aktuální podobu foceného místa. Je velmi podobný projektu historypin. Také funguje jako online archiv, sdružuje nadšence se zájmem o místní historii, ale liší se možností srovnání dvou fotografií. Shromážděním fotografií z různých míst světa se snaží ukázat historii světa. Poskytuje podobné funkce jako ostatní zmíněné projekty: nahrávání, prohlížení a správu fotografií.

<sup>3</sup>Historypin - <http://www.historypin.com>

<sup>4</sup>WhatWasThere - <http://www.whatwasthere.com>



Obrázek 2.5: Ukázka prohlížení fotografií projektu WhatWasThere. Obrázek ukazuje zobrazení fotografie na Google Street. V horním rohu fotografie je možné měnit průhlednost fotografie a srovnávat s aktuální podobou místa. Veškeré zobrazení probíhá pouze v dialogových oknech.

## Rozdíl mezi existujícími aplikacemi a mnou tvořenou aplikací

Všechny zmíněné aplikace nabízejí podobné funkce, soustředí se na problematiku repeat photography nebo archivují historické fotografie a pohlednice. Žádný z výše zmíněných projektů neslouží k pořizování nových záběrů historických pohlednic. Projekt WhatWasThere zobrazuje srovnání na fotografiích z Google Street View<sup>5</sup>. Často si nelze všimnout změn z důvodu nekvalitních aktuálních snímků. Ty jsou blokovány stromy apod. Projekt rePhoto má pouze webovou aplikaci pro prohlížení pořizených fotografií, ale neposkytuje možnost poskytnutí fotografie z vlastního aparátu. Projekt historypin zobrazuje historické fotografie bez možnosti srovnání v časové ose. Cílem mé aplikace je vytvořit aplikaci bez těchto nedostatků.

<sup>5</sup>Google Street View - <https://www.google.cz/intl/cs/streetview/>

## Kapitola 3

# Použité technologie a nástroje k realizaci práce

V této kapitole jsou popsány technologie, které byly využity při implementaci této práci. V dnešní době je možné pro vývoj webové aplikace využít mnoho jazyků a technologií. Pro práci byly vybrány populární a prověřené technologie, které umožnily rychlý vývoj především díky aktivní komunitě a skvělé dokumentaci.

### 3.1 Technologie použité pro vývoj

#### Composer

Composer je nástroj pro správu balíčků a knihoven v PHP. Podporuje jejich instalace, vytváření závislostí mezi nimi a jejich aktualizace. V dnešní době, kdy programátor využívá mnoho knihoven, které jsou velmi často aktualizovány, je téměř nutností jej používat. Některé knihovny nelze bez Composeru nainstalovat. Velmi výhodná je i kontrola prostředí, při které kontroluje verzi PHP a instalaci požadovaných modulů. To se hodí při přesunu aplikace na jiné prostředí [?].

#### Git

Git je distribuovaný systém pro správu verzí. Je nezávislý na operačním systému a liší se v konkrétních implementacích. Při práci využívá repozitáře. Ty jsou využívány jako lokální nebo vzdálené úložiště. Při inicializaci repozitáře je vytvořena skrytá složka `.git`, ve které jsou uloženy všechny metadata. Verzování (ukládání změn) probíhá pomocí příkazu `commit` v příkazové řádce. Všechny verze jsou ukládány a je možné se k nim během práce vracet. Přenos mezi jednotlivými repozitáři probíhá příkazy `pull` a `push`. Při stahování nebo odesílání změn ze vzdáleného serveru jsou verze kontrolovány a následné konflikty v kódu musí být vyřešeny. Ve své práci jsem ho používal pro přesun kódu mezi zařízeními a pro zálohování [6].

### 3.2 Front-end technologie

V této kapitole jsou vysvětleny technologie pro vývoj uživatelského rozhraní. Při výběru byl kladen důraz na možnost rychlého prototypování, moderní a přehledný design a dostatečné možnosti úprav.

## HTML5 a CSS3

V dnešní době jsou standardní technologie pro popis dokumentu a jeho vzhledu nejnovější verze HTML 5 (Hypertext markup language 5) a CSS 3 (Cascading Style Sheets 3), které přinášejí oproti svým předchůdcům mnohé vylepšení. HTML je popis pro značkovací jazyk a HTML 5 je konkrétní specifikace. CSS 3 je posledním standardem pro popis stylů plně kompatibilní se svými předchůdci. Při vytváření webových stránek je nutné rozlišovat, co je ve specifikaci a co mají prohlížeče implementované [7].

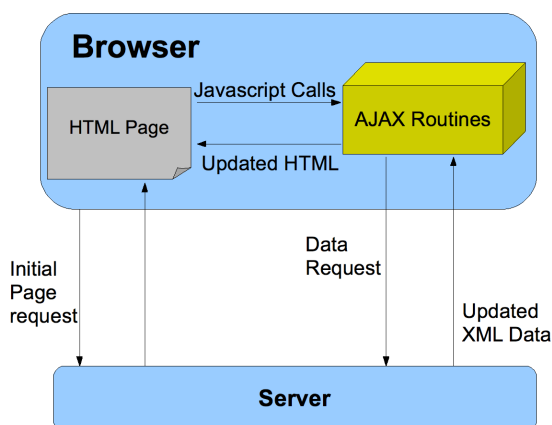
I když nejsou veškeré funkce implementované ve všech prohlížečích, oba jazyky přináší mnohá vylepšení, rychlejší vývoj a především nové funkce. Nový element canvas je použit při implementaci grafického editoru popsáno v kapitole 4.7.

## JavaScript

JavaScript je skriptovací jazyk pro zvýšení interaktivity webu. V poslední době se objevuje nejen ve webových prohlížečích, ale i na serverech a v dalších technologiích. Syntaxe tohoto jazyka vychází z jazyku Java a nevyžaduje překlad do strojového kódu. Podporuje objektově orientované programování a rozlišuje malá a velká písmena. Slouží pro práci s HTML elementy a umožňuje téměř kompletně upravit stránku bez nutnosti využití serverové podpory. Pro psaní kódu je možné využít načtení externího souboru pomocí tagu `<script src='soubor.js'>` umístěného v hlavičce nebo tzv. inline-skriptů, které se vkládají přímo do HTML kódu [8].

## AJAX

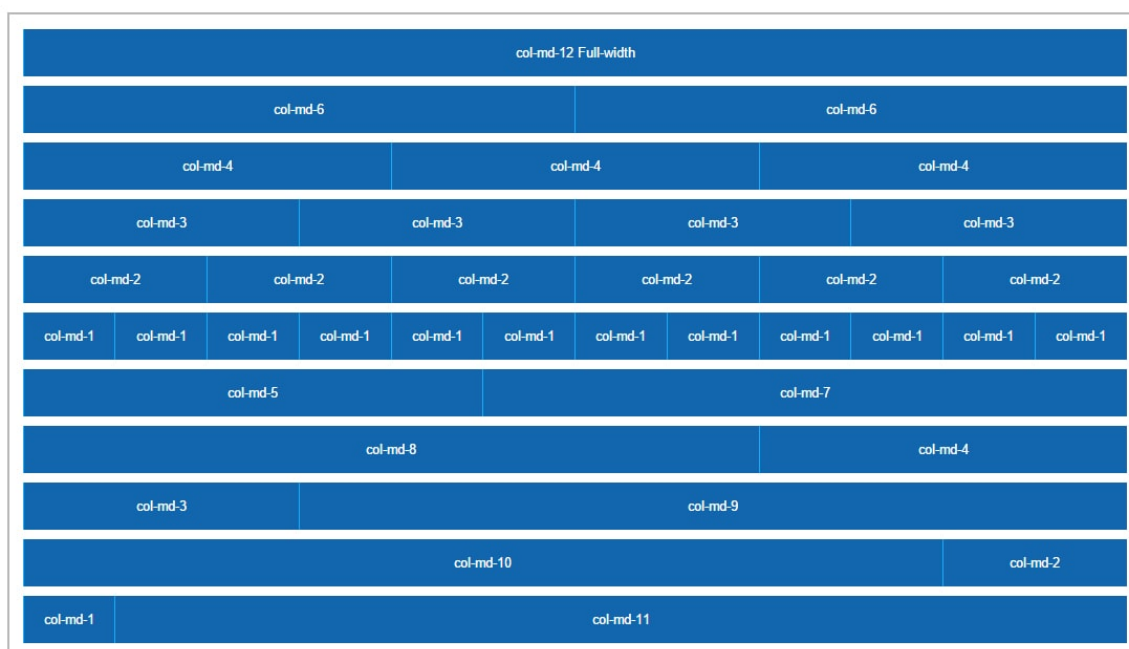
AJAX je spojení technologií za účelem asynchronního komunikování se serverovou částí z klientské strany. Při použití formátu XML je používána značka AJAX (asynchronous javascript and XML) a při použití formátu JSON se uvádí jako AJAJ (asynchronous javascript nad JSON). Při této komunikaci se šetří datový tok a serverové zdroje. Není nutné posílat celý HTTP požadavek a celou stránku vygenerovat znovu, ale stačí pouze data nutná k překreslení části stránky. Tento přístup také zvyšuje pohodlné používání webových aplikací z důvodu rychlé odezvy [9].



Obrázek 3.1: Obrázek ukazuje princip technologie AJAX. Jednotlivé AJAX rutiny jsou volány. Ty následně komunikují se serverem, který odpovídá ve formátu XML a HTML dokument je překreslen.

## Bootstrap

Bootstrap je front-end framework pro tvorbu responzivních webových stránek. Obsahuje kombinaci HTML, CSS a JavaScript kódu, která umožňuje rychlý vývoj uživatelských rozhraní bez nutnosti psaní velkého množství kódu. Se základním stylem pro tabulky, menu, tlačítek a formulářů je vhodný pro okamžité použití. JavaScriptové komponenty umožňují interaktivní chování GUI. [10] Umístění elementů na stránce je zobrazeno pomocí grid systému. Ten obsahuje dvanáct různě velkých mřížek, které je možné různě kombinovat.



Obrázek 3.2: Obrázek ukazuje mřížkové rozdělení. Jednotlivé rozdělení je možné jakkoliv kombinovat a vytvořit tak strukturu HTML dokumentu.

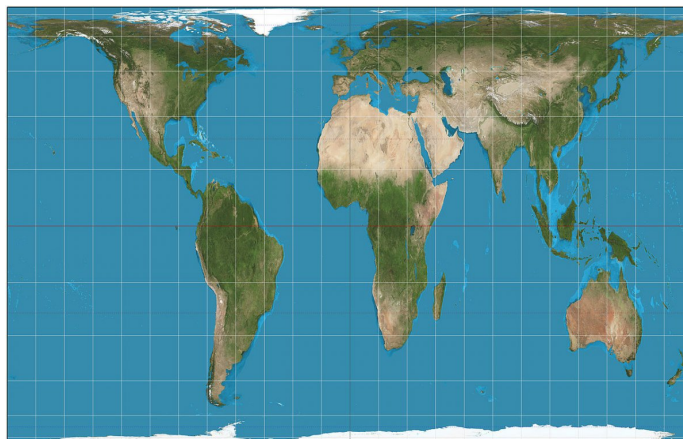
Ačkoliv se může zdát, že předepsaný kód limituje tvůrce GUI, není tomu tak. Bootstrap je plně přizpůsobitelný a jednoduše rozšiřitelný. [11]. Byl vybrán pro jeho nejpropracovanější grafický návrh s mnoha komponenty a z důvodu velkého množství příkladů.

## Google Maps

Pro práci s mapou byly využity mapy od firmy Google. Ty projektují svět na plochém povrchu ve tvaru obdelníku. Tento přístup byl navržen zhruba před třista lety a přináší zkreslení zejména v oblastech obou pólů, ale má výhody, které převažují nad jeho nedostatky [12].

Google Maps poskytují API vhodné pro vlastní webové aplikace. Dovolují upravit vzhled, nechybí podpora pro události na mapě, zvládnou navigaci k místu a označování bodů na mapě. Při práci s lokací využívá body zeměpisné výšky a šířky.





Obrázek 3.3: Obrázek ukazuje Mercatorovo zobrazení. To vykresluje svět na rovné ploše a dochází k mírnému zkreslení v oblasti obou pólů.

## jQuery

jQuery je JavaScriptová knihovna zjednodušující interakci mezi HTML, přesněji DOM<sup>1</sup> a JavaScriptem. Specificky ulehčuje přesouvání a manipulaci HTML dokumentu, zachytávání událostí prohlížeče, použití AJAXových požadavků a sjednocení kódu podporovaného více prohlížeči. Filozofii této knihovny nejlépe vyjadřuje slogan "Napiš méně, udělej více." Ta může být rozdělena do tří skupin:

- Najdi HTML elementy přes CSS selektor a proved s nimi následující operace.  
`jQuery(elementystridou).operace()`
- Na elementy použij více metod, oddělených znakem.  
`jQuery(elementystridou).operace().dalsioperace().dalsioperace()`
- Využívej možnosti jQuery a implicitní iterace.  
`jQuery(elementystridou).operace(callback(), dalsioperace())`

Knihovna během své existence získala velkou popularitu především díky své filozofii. Výrazně ulehčuje interakci webových stránek a zrychluje vývoj. Na stránkách projektu se nachází mnoho doplňků, které je možné bezplatně použít na vlastních projektech. Ty řeší časté problémy při vytváření webových stránek [13].

## 3.3 Back-end technologie

Při výběru technologií pro serverovou část bylo možné vybrat z mnoho jazyků a frameworků. Důraz byl kladen na dostačující výkon, cenovou dostupnost hostingu, podporu pro práci s obrázky a rychlý vývoj. Zvolil jsem jazyk PHP<sup>2</sup>, který využívá celosvětově 82.6% všech webových aplikací[14]. Díky velké popularitě existuje velké množství kvalitních a levných webhostingů. V nové verzi PHP 7 při řešení Mandelbrotovy množiny dosahuje nejlepších výsledků oproti svým konkurenčním jazykům. Kvalitní dokumentace a velké množství frameworků jen potvrdily výběr na základě výše zmíněných požadavků.

<sup>1</sup>DOM - Document object model <https://www.w3.org/DOM/>

<sup>2</sup>PHP -/ Hypertext preprocessor <http://php.net/manual/en/>

Tabulka 3.1: Srovnání doby interpretace vykreslení kódu Mandelbrotovy množiny. I když je jazyk PHP brán jako pomalý, nová verze dosahuje nejlepších výsledků v porovnání s ostatními jazyky v tabulce. Tabulka přebrána z infografiky na straně [https://www.zend.com/en/resources/php7\\_infographic](https://www.zend.com/en/resources/php7_infographic)

Jazyk	PHP 7	RUBY	PYTHON	PERL
Čas [s]	0.281	0.684	1.128	2.083

Při výběru rámce pro jazyk PHP se programátor může rozhodovat mezi velkým množstvím projektů. Mezi nejznámější patří Symfony, Nette, Zend, Yii nebo Laravel. Všechny se liší především svou filozofií a přístupem k určitým problémům. Symfony vyniká svojí knihovnou pro datábazovou vrstvu Doctrine a obsahuje mnoho doplňků. Nette má velmi kvalitní ladící nástroj a silný šablonovací systém. Zend je projekt od tvůrců jazyku PHP a vyniká velmi kvalitní dokumentací [15]. Yii a Laravel byly z výběru vyřazeny pro nedostatečnou podporu práce s obrázky. Pro svoji práci jsem si nakonec vybral český rámec – Nette Framework.

## REST API

REST, celým názvem Representational State Transfer je soubor podmínek, které definují způsob distribuce dat. Ten je implementován jako webové rozhraní komunikující přes protokol HTTP pro distribuci dat. Komunikace probíhá podle modelu klient-server a je bezstavová. Při každém požadavku je potřebné uvést všechny parametry nutné k vykonání akce, server neudržuje žádný kontext komunikace.

Zdrojem v REST API může být cokoliv. Často se jedná o elektronické dokumenty, uživatele v aplikaci, ale mohou to být např. i rodinné vztahy. Omezením je, že každý zdroj musí mít svou unikátní URL<sup>3</sup>. Zdroj může mít více reprezentací, např. vládní API mohou poskytovat dokumenty ve více jazycích nebo ve více formátech. I když zdrojem může být cokoliv, uživatel API může vykonat pouze tyto operace definované sémantikou protokolu HTTP:

**GET** - Získání reprezentace zdroje.

**DELETE** - Smazání zdroje.

**POST** - Vytvoření nového zdroje na základě získané reprezentace.

**PUT** - Nahrazení zdroje za jiný zdroj na základě získané reprezentace.

Klient posílá požadavek na server. Cílem serveru je vyhodnotit reprezentaci zdroje, provést požadovanou operaci a vrátit odpověď. Reprezentace odpovědi může obsahovat data nebo stav ve formátu XML, JSON, HTML nebo v jakémkoliv jiném[16].

REST API není jediný způsob distribuce dat. Oproti RPC (vzdálenému volání procedur) má konečnou množinu operací. Bylo vybráno z důvodu jednoduché rozšiřitelnosti a dostatečného množství knihoven pro realizaci a komunikaci.

<sup>3</sup><https://www.w3.org/Addressing/URL/url-spec.txt>

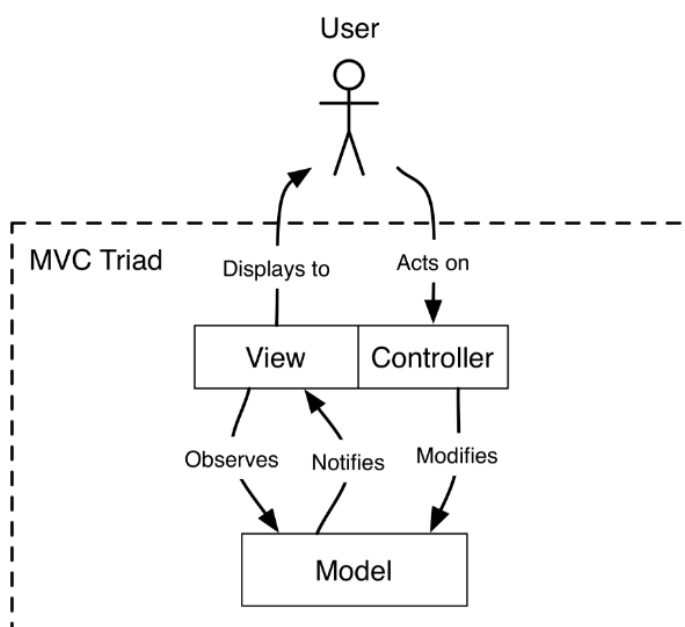


## Nette Framework

Nette Framework je český svobodně vyvíjený PHP framework pro vývoj webových aplikací. Skládá se z mnoha komponent, které mohou být nahrazeny nebo upraveny podle potřeb programátora. Podporuje AJAX, DRY<sup>4</sup>, MVC a znovupoužitelnost kódu. Byl vybrán pro jeho aktivní českou komunitu a silnou orientaci na bezpečnost. Při výzkumu popularity v roce 2015 skončil na třetím místě za celosvětově známými frameworky Zend a Symfony. [17]

## Koncept Model-View-Controller

Koncept Model-View-Controller (MVC) je pravděpodobně nejvíce používaná architektura pro vývoj uživatelských rozhraní a webových aplikací. Aplikace je rozdělena na tři logické celky: Model, View a Controller. Model reprezentuje data a celkovou logiku aplikace. View zobrazuje uživatelské rozhraní. Controller řeší sled událostí a aplikační logiku. Silnou stránkou konceptu je nezávislost vrstev – změny v jakékoliv vrstvě mají minimální dopad na ostatní vrstvy. [17]



Obrázek 3.4: Princip architektury MVC. Uživateli je zobrazeno uživatelské rozhraní z vrstvy view. Akce uživatele zpracovává controller. Ten komunikuje s modelem, který řeší celkovou logiku.

## Nette Sandbox

Sandbox je název pro základní strukturu projektu v Nette rámci. V základu tato struktura obsahuje celý framework s předpřipravenou funkční aplikací. Pro instalaci se doporučuje nástroj Composer.

`/app` - Obsahuje zdrojové kódy aplikace.

<sup>4</sup>DRY - Don't repeat yourself <https://www.zdrojak.cz/clanky/navrhove-principy-dry/>

`/tests` - Složka pro unit testy.

`/temp` - Dočasná složka pro cache.

`/vendor` - Knihovny nainstalované přes Composer.

`/www` - Veřejně přístupný projekt.

## Nette Database

Tato knihovna mapuje databázové tabulky na objektové entity. Tento přístup je vhodnější pro reprezentaci reálného světa. Výhodou při používání této knihovny je nezávislost na konkrétním databázovém serverovém systému. Tato vlastnost je zajištěna využitím PDO<sup>5</sup>. Pro práci s konkrétní tabulkou může programátor psát vlastní dotazy nebo využít objekt `Table`. Ten zjednodušuje a optimalizuje výběr dat z databáze. Hlavní myšlenkou je načítání dat pouze z jedné tabulky a jen jednou. Knihovna sama zjišťuje vazby na ostatní tabulky pomocí cizích klíčů a své dotazy ukládá do cache paměti. Při položení dotazu vrací objekt `ActiveRow` pro jeden řádek a pro více řádků objekt `Selection` [18].

```
1 $books = $context->table('book');
2 foreach ($books as $book) {
3     echo $book->title;
4     echo $book->author_id;
5 }
```

Listing 3.1: Ukázka použití výpisu dat z tabulky `book`.

## Nette Forms

Nette formuláře usnadňují práci s vytvářením a zpracováním formulářů ve webových aplikacích. Automaticky formulář zajišťují proti několika útokům, poskytují validaci i na straně klienta, podporují vícejazyčnost a jsou jednoduše vykreslitelné. Vytváření probíhá instanciováním objektu `Form`, kterému jsou předány jednotlivé prvky. K jednotlivým prvkům lze přidávat různé druhy předepsaných validací. Pokud předepsané validace nestačí, je možné si dopsat vlastní [19].

```
1 $form = new Form();
2 $form->addText('email', 'E-mail')
3     ->addRule(Form::EMAIL)
4     ->setRequired('E-mail is required');
5 $form->addText('given_name', 'Given name')
6     ->setRequired('Given name is required.');
```

Listing 3.2: Ukázka vytvoření formuláře s formulářovými prvky pro e-mail a jméno. Oba prvky mají navázané validace před odesláním. Po odeslání formuláře je volána metoda `processForm`.

## MySQL

MySQL je relační databázový systém šířený pod volnou licencí GPL a dostupný na všech nejčastěji používaných operačních systémech. Tabulky v MySQL mají více typů úložišť.

<sup>5</sup>PDO - <http://php.net/manual/en/book.pdo.php>

Uložiště MyIsam nepodporuje cizí klíče, ale je ze všech nejrychlejší. Oproti tomu InnoDB podporuje cizí klíče, uzamykání na úrovni řádků, ale je o poznání pomalejší. Jak už sám název napovídá, je ovládaný jazykem SQL. Ten se skládá z příkazů pro manipulaci dat, definici dat, řízení přístupu a řízení transakcí.

## Kapitola 4

# Návrh a implementace aplikace

Cílem této kapitoly je vysvětlit návrh a implementaci aplikace. V této kapitole popisují návrh uživatelského rozhraní, návrh databáze, případy užití, jednotlivé problémy řešené při implementaci a samotnou implementaci.

### 4.1 Případy užití

Před samotnou implementací a návrhem aplikace byly určeny případy užití pro běžného uživatele. Běžní uživatelé jsou fotografové, historikové, sběratelé pohlednic apod.

#### *Nahrání historické pohlednice*

Aby mohl být projekt komunitně řízený je důležité, aby každý uživatel mohl nahrávat a spravovat své historické pohlednice. Cílem je vytvořit kvalitní databázi historických pohlednic.

#### *Nahrání aktuálního snímku pohlednice*

Pro fotografy, kteří nechtějí z jakéhokoliv důvodu využívat mobilní aplikaci pro pořizování nových záběrů, musí existovat možnost nahrání fotografie přes webovou aplikaci.

#### *Prohlížení fotografií a pohlednic*

Funkce je pro všechny návštěvníky, kteří si chtějí prohlédnout své okolí a porovnat historickou proměnu.

#### *Hodnocení fotografií*

V rámci zvýšení motivace pro pořizování fotografií je vhodné mít možnost hodnocení.

#### *Prohlížení uživatelských profilů*

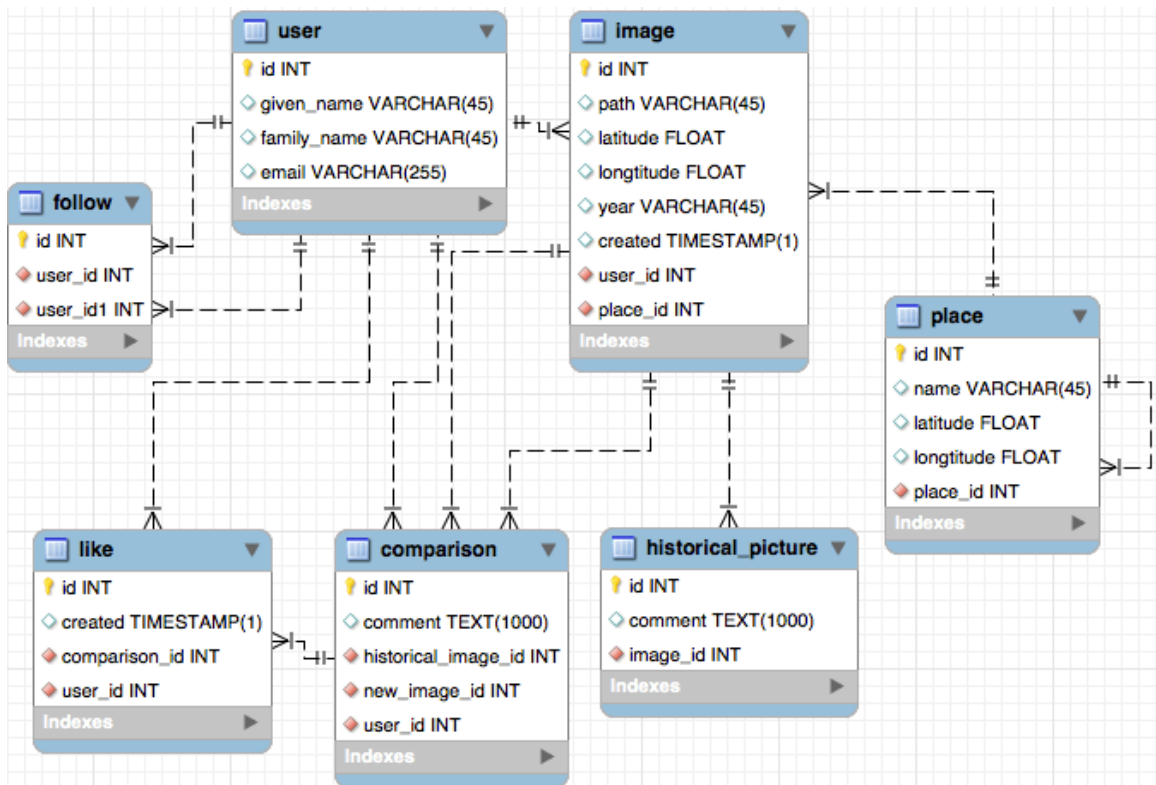
Tato funkce motivuje ostatní fotografy, aby mohli srovnávat své výtvořky s ostatními.

#### *Sdílení na sociální síť*

Sociální síť jsou nedílnou součástí většiny z nás. Sdílení může pomoci jako silný marketingový nástroj a uživatelé mohou prezentovat své výtvořky nejen na svém profilu na stránce projektu.

### 4.2 Návrh databáze

Po analýze případů užití, tj. po stanovení rolí a možností, co může uživatel v aplikaci vykonávat, bylo nutné navrhnout, co bude systém udržovat za data.



Obrázek 4.1: Relační schéma databáze. Schéma znázorňuje strukturu tabulek a jejich relace.

#### *User*

Uchovává všechny důležité informace o uživateli včetně přihlašovacích údajů, facebook token, přezdívky a otisku hesla.

#### *Image*

Entita obsahující všechny obrázky nahrané do systému. Položka path je cesta v souborovém systému. Souřadnice zeměpisné výšky a šířky slouží pro geolokaci a při vykreslování značek na mapě. Každá nahráná fotografie je spjata s uživatelem.

#### *Follow*

Tabulka pro ukládání sledování uživatelů. Každý uživatel může sledovat jiného uživatele.

#### *Like*

Obsahuje "líbí se mi" všech fotografií.

#### *Comparison*

Tabulka obsahující srovnání staré pohlednice a nově nahrané fotografie. Obsahuje také komentář vložený při pořízení nového snímku.

#### *Historical picture*

Slouží pro rozšíření údajů pohlednice o komentář a rozeznání od fotografie.

## 4.3 Návrh uživatelského rozhraní

Pro návrh uživatelského grafického rozhraní (dále GUI) byl použit framework Bootstrap. Díky možnosti rychlého prototypování nebyly tvořeny drátěné modely, které jsou vhodné pro náhled řešení, ale rovnou funkční prototyp. Základní vzhled frameworku byl upraven. I přes fakt, že bootstrap podporuje responzivní design, nebylo uživatelské rozhraní tvořeno pro mobilní zařízení z důvodu existence mobilní aplikace. Při navrhování GUI jsem vycházel z obecných zásad, které zajišťují kvalitu, srozumitelnost a intuitivnost. Uživatel nesmí nad uživatelským rozhraním přemýšlet, ale pohodlně jej používat [21]. Zásady využití při návrhu:

### Zpětná vazba

Po provedení akce je nutné uživatele informovat jak proběhla.

### Nezatěžovat paměť uživatele

Nenutit uživatele nic počítat. Omezit zbytečné zadávání vstupů apod.

### Přehlednost

Rozhraní musí být na první pohled přehledné, uživatel musí tušit, co mu dané zobrazení říká a co může provádět.

### Vizuální nápověda

Změna kurzoru, posuvníky, pozice v dokumentu apod.

### Možnost "zpět"

I přes podporu prohlížečů, které dovolují se vrátit na poslední stranu v historii, je důležité, aby uživatelské rozhraní bylo bez slepého bodu.

### Konzistence

Aplikace by měla zobrazovat stejné hlášky, podobné zobrazení a stejné návyky při podobných akcích.

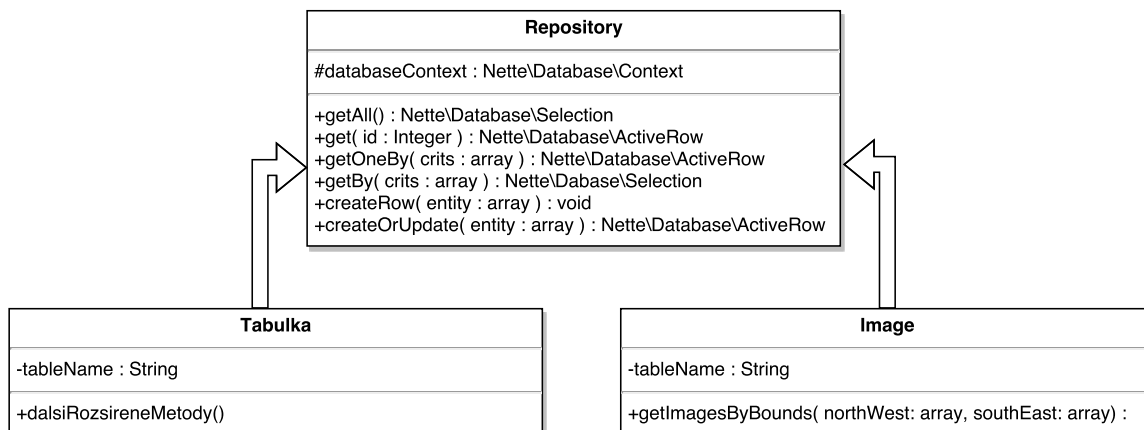
## 4.4 Struktura aplikace

Jako základní kostra struktury byl využit Nette Sandbox popsáný v kapitole 3.3. Protože aplikace slouží i jako server a obsahuje administraci, byla rozdělena na tři základní moduly. Modul API, samotná aplikace a administrace. Rozdělení do modulů poskytuje jednodušší rozšiřitelnost aplikace a zvyšuje přehlednost. Výsledná struktura aplikace vypadá následovně:

```
app
  AdminModule
    model
    presenters
    templates
  ApiModule
  ..
  FrontModule
  ..
```

## 4.5 Implementace databázové vrstvy

Pro zjednodušení a redukování psaní kódu jsem vytvořil obálku, která ulehčuje práci s Nette Database. Všechny tabulky jsou třídy dědící z třídy *Repository*. Při vytvoření objektu *Repository* je vloženo datábázové připojení z kontejneru závislostí. Předek poskytuje všechny základní operace společné pro všechny potomky. Mezi tyto operace patří: získání jednoho záznamu podle id nebo podle kritérií, získání více záznamů podle kritérií, mazání a úpravy záznamů. Pokud je potřeba rozšířit funkčnost jakékoliv třídy např. při složitějších dotazech nebo v případě propojení datábázového záznamu se souborem, stačí dopsat danou metodu v konkrétní třídě.



Obrázek 4.2: Diagram tříd znázorňující popis vytvořené struktury pro práci s jednotlivými tabulkami. Repository (předek) poskytuje základní operace pro komunikaci s databází. Jednotlivé tabulky jsou potomky dědící datábázové připojení.

## 4.6 Implementace mapy

Pro komunikaci s rozhraním Google Maps je nejprve nutné si vygenerovat klíč. Využívání API je limitované na 10000 požadavků denně a klíč se využívá jako identifikátor. V případě nedostačujícího limitu je možné použít placenou verzi. Sestrojení mapy probíhá instanciováním objektu *Google.Maps.Map*, který vyžaduje připnutí na HTML element. Po vytvoření objektu je mapa inicializována a umožňuje procházet svět. Při implementaci byl využit plugin, který zjednodušuje používání rozhraní Google Maps.

V této práci bylo potřebné řešit 3 problémy – označení bodů fotografií na mapě, získání hranic pro zobrazení fotografií v pravém panelu a označování bodů při nahrávání historické pohlednice. Označení bodů fotografií na mapě umožňuje API pomocí metody *Google.Maps.Map.addMarker()*. Procházením záznamů historických pohlednic byly značky vytvářeny a ukládány do pole pro případné další použití. Zjištění zobrazených značek na mapě byla použita metoda *Google.Maps.Map.getBounds()*, která vrací souřadnice levého horního a pravého dolního rohu. Pro označování bodů na mapě byly využity události pro změnu středu, kliknutí a posunutí, při kterých byl bod překreslen.

## Mapa pohlednic

Mapa pohlednic je rozdělená na 2 panely. Vlevo se nachází mapa, vpravo je panel s pohlednicemi a dodatečnými informacemi. Po načtení mapy je zjištěna hranice viditelných bodů na mapě, poslán AJAXový požadavek na server a následně pravý panel překreslen s viditelnými pohlednicemi na mapě. Pro zvýšení přehlednosti a orientaci na mapě je sledována událost najetí myši na pohlednici v pravém panelu, která aktivuje animaci "skákání" bodu na mapě. Při opuštění kurzoru je animace deaktivována.

## Přidání pohlednice s označením lokace na mapě

Pro přidání pohlednice je nutné označit co nejpřesnější lokaci odkud byla pohlednice pořízena. Z tohoto důvodu je formulář rozdělen na dva panely. V levém se nachází mapa s předkresleným bodem, se kterým je možné pohybovat. Posunutí bodu probíhá i jedním kliknutím na lokaci na mapě. Na mapě jsou dále vykresleny vstupy pro GPS souřadnice a adresu. Při změně jednoho vstupu jsou aktualizovány ostatní vstupy a bod je posunutý na konkrétní místo.

```
1 // inicialize mapy
2 var map;
3 map = new GMaps({
4     div: '#map',
5     lat: 49.80,
6     lng: 15.606837,
7     zoom: 7,
8 });
9 marker = map.addMarker( {
10     id: {$picture->id},
11     lat: {$picture->image->latitude},
12     lng: {$picture->image->longtitude},
13     title: {$picture->image->address},
14     animation: google.maps.Animation.DROP
15 });
```

Listing 4.1: Ukázka kódu inicializace mapy a vytváření značek na mapě. Při vytváření značek je nutné zadat souřadnice zeměpisné výšky a šířky.

## 4.7 Vlastní jednoduchý grafický editor

Při implementaci ořezování byly využity tři rozšíření knihovny jQuery: [croppie](https://foliotek.github.io/Croppie/)<sup>1</sup>, [SimpleCropper](https://github.com/tomazdragar/SimpleCropper)<sup>2</sup> a [jQuery Guillotine](https://github.com/matiasgagliano/guillotine)<sup>3</sup>. Všechny byly snadně použitelné s rychlým výsledkem, avšak měly svá omezení. Složitější úpravy komplikovaly chyby a konflikty s ostatními knihovnami. Rozšíření neumožnily náhled na straně klienta. Rozšíření už nejsou podporované a aktivně vyvíjené. Z tohoto důvodu jsem se rozhodl pro vlastní řešení s využitím elementu canvas.

Po nahrání nově pořízené fotografie je zobrazen element canvas s 2D kontextem. Elementu je přiřazeno pozadí historické pohlednice a nahraná fotografie se vykreslí s průhledností v kontextu. Sledováním událostí myši, je fotografie překreslena:

<sup>1</sup>croppie - <https://foliotek.github.io/Croppie/>

<sup>2</sup>SimpleCropper - <https://github.com/tomazdragar/SimpleCropper>

<sup>3</sup>jQuery Guillotine - <https://github.com/matiasgagliano/guillotine>



- Přiblížení fotografie - dvojitě kliknutí levého tlačítka nebo rolování kolečka směrem dolů.
- Oddálení fotografie - rolování kolečka směrem nahoru.
- Posun fotografie - kliknutí a držení levého tlačítka, při puštění je posun zastaven.

Při posunu se vypočte nová relativní pozice myši od počátečního bodu. Kontext je vymazán a fotografie při pohybu překreslena. Pro oddalování a přibližování zobrazované fotografie je použita metoda změny měřítka.

```

1 var ctx = canvas.getContext('2d');
2   trackTransforms(ctx);
3   function redraw() {
4       // Clear the entire canvas
5       ctx.globalAlpha = opacity;
6       var p1 = ctx.transformedPoint(0, 0);
7       var p2 = ctx.transformedPoint(canvas.width, canvas.height);
8       ctx.clearRect(p1.x, p1.y, p2.x - p1.x, p2.y - p1.y);
9       // draw image
10      ctx.drawImage(img, 0, 0, img.width * (canvas.width / img.width), img
11                    .height * (canvas.width / img.width));
    }

```

Listing 4.2: Ukázka překreslení 2D kontextu elementu canvas. Tato metoda je volána při událostech vyvolané myši. Celý kontext je vymazán a obrázek překreslen v nové velikosti nebo s novou pozicí.

## 4.8 REST API

Jedná se o webovou aplikaci sloužící jako server pro mobilní aplikaci. Při komunikaci je použit formát JSON, ale je implementován i formát XML. Aby mohl klient, v našem případě mobilní aplikace, komunikovat s rozhraním, je nutné aby se autentizoval do aplikace. Pro zabezpečení rozhraní je využit privátní klíč, který se přenáší v hlavičce každého požadavku. Tento klíč je šifrován pomocí funkce *sha1()*. Pokud se šifry privátních klíčů rovnají, server odpovídá, jinak nikoliv.

### Struktura API

Tato struktura byla podle požadavků mobilní aplikace upravována a rozšiřována o další funkce. Nebyla porušena zpětná kompatibility, a proto byla zachována první verze.

```

/v1
  /hello
  /users
    / [POST]
    /id [GET]
    /login [POST]
    /logout [DELETE]
  /images

```

```
/ [GET]
/ [POST]
/id [GET]
```

## Vybrané funkce a jejich popis

/hello

Testovací zdroj, který byl zanechán pro kontrolu funkčnosti API.

### Formát odpovědi

```
1 {
2   "hello": "world",
3 }
```

/users

Tento zdroj slouží pro práci s uživateli. Mobilní aplikace může uživatele registrovat, upravovat jejich údaje, přihlásit a nebo odhlásit.

/images/[souradnice]/[souradnice]/[souradnice]/[souradnice]

**GET** - Pro získání jednoho zdroje nebo kolekce historických fotografií. Při zadání souřadnic je kolekce omezena na lokaci, ze které chce klient získat pohlednice.

### Formát odpovědi

```
1 {
2   "historicalPhotoId": "1",
3   "image_url": http://rephoto.cz/api/v1/images/1,
4   "comment": 127,
5   "address": 100,
6   "lat" : 49.774121,
7   "long" : 19.8384893
8 }
```

**POST** - Vytvoření nového zdroje a nahrání fotografie. Obrázek je poslán jako pole bytů v kodování Base64<sup>4</sup>. Ořez obrázku probíhá na straně serveru.

### Formát požadavku

```
1 {
2   "historicalPhotoId": "1",
3   "image": byteArray,
4   "x": 127,
5   "y": 100,
6   "width" : 200,
7   "height" : 200
8 }
```

## 4.9 Ukládání obrázků

V rámci aplikace jsou veškeré obrázky ukládány v souborovém systému. Ten oproti uložení v databázi přináší několik výhod i nevýhod. Je jednodušší pro správu, je méně náročný na zdroje, ale na druhou stranu neumožňuje kontrolu integrity dat a jednoduché zálohování. Datové úložiště je reprezentována třídou *Storage*. Ta poskytuje metody pro uložení,

<sup>4</sup>Base64 – <https://tools.ietf.org/html/rfc4648>

ořezávání, odstranění souboru a dekodování Base64. Každý obrázek je reprezentován svým unikátním jménem. Toto omezení zajišťuje unikátní klíč v databázové tabulce.

```
1  /** @var App/Model/Storage @inject */
2  private $storage;
3
4  /**
5   * Resource method for creating and uploading image
6   * @POST image
7   */
8  public function actionCreate()
9  {
10     $fileName = $this->storage->saveBase64($this->input->image);
11     $this->storage->crop($fileName, $this->input->scale, $this->input->x
12         , $this->input->y, $this->input->width, $this->input->height);
13     ...
14 }
```

Listing 4.3: Ukázka použití úložiště při práci s obrázky. Úryvek ukazuje zpracování REST zdroje.

## 4.10 Autorizace a autentizace

Pro přihlášení do aplikace jsou implementované dvě autentizační třídy. První využívá přihlášení pomocí e-mailu a hesla. Druhá třída autentizuje uživatele pomocí facebookového profilu. Při autorizaci jsou rozlišovány tři role: návštěvník, kurátor a administrátor. Návštěvník může prohlížet celou aplikaci, ale nemá možnost nahrát fotografii či starou pohlednici. Kurátor může nahrávat staré pohlednice a její aktuální podobu. Má svůj veřejně přístupný profil, kde jsou jeho fotky zveřejněny. Administrátor má přístup do admin sekce, kde je možné všechny fotografie mazat a celý projekt spravovat.

## Kapitola 5

# Iterativní testování prototypu a postupné vylepšování

Vybraným uživatelům byla předložena aplikace k vyzkoušení. Vedený dialog pomáhal k získání nových poznatků, jež vedly k následnému vylepšení aplikace.

### Scénáře prováděné uživateli:

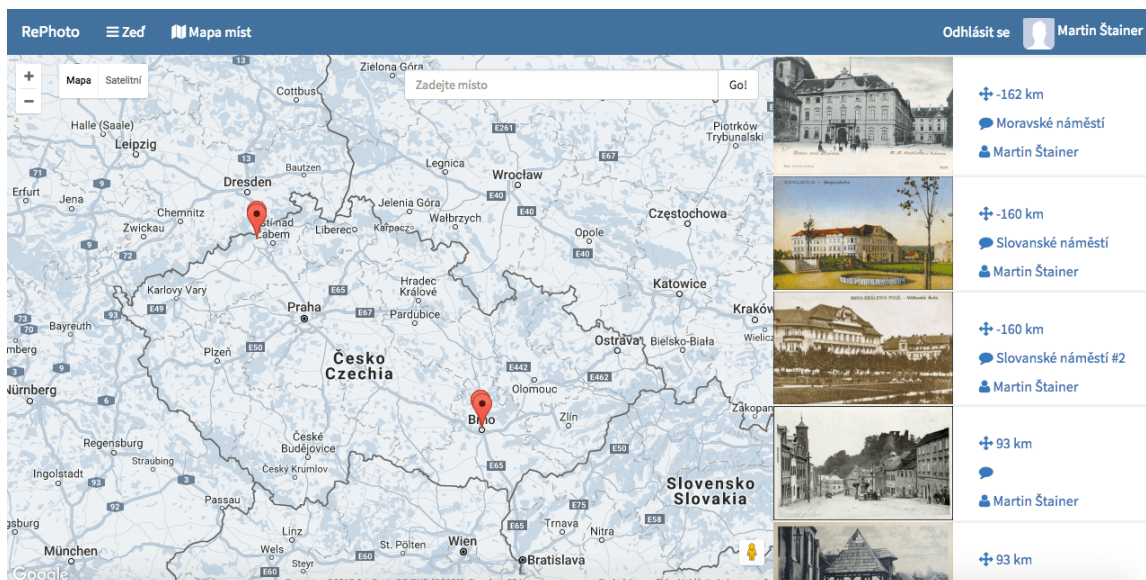
- Nahrání historické pohlednice
- Nahrání aktuálního snímku historické pohlednice
- Registrace, přihlášení a úprava profilu
- Práce s mapou a prohlížení míst, fotografií
- Prohlížení uživatelských profilů

## 5.1 Prototyp

Prototyp obsahoval mnoho chyb, které omezovaly pohodlné používání aplikace. Uživatelé si především stěžovali na detaily v grafickém návrhu, dlouhé trvání načítání a celkové nepochopení určitých funkcí. Nicméně všechny scénáře byly úspěšně dokončeny.

### Práce s mapou

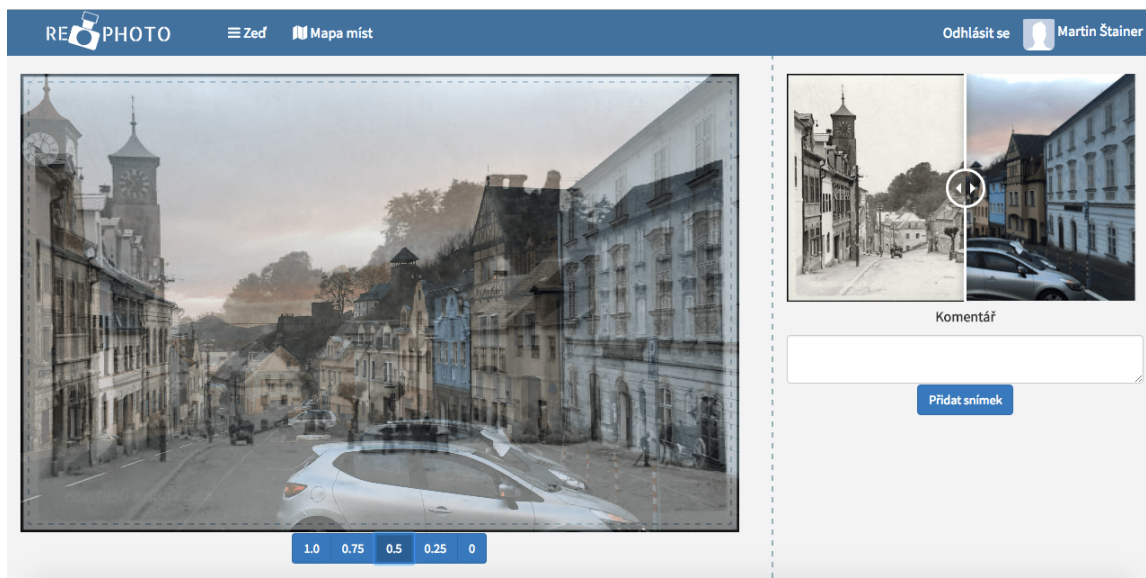
Při zobrazení mapy byl uživatel přiblížen na přesnou lokaci. Uživatelé se shodli, že toto chování není vhodné, protože vidí pouze malou část mapy a nemají přehled, co se na mapě vyskytuje. Většina z nich hned při zobrazení stránky mapu oddalovala. Z tohoto důvodu bylo chování upraveno a mapa oddálena tak, aby byla vidět celá země. Uživatelům rovněž vadil prázdný panel s popisem historických pohlednic. Pokud na mapě nejsou žádné pohlednice, v pravém panelu se zobrazí nejbližší z nich.



Obrázek 5.1: Screenshot provedených úprav na straně s mapou. Obrázek ukazuje pohled po načtení mapy, kdy uživatel již není geolokován na přesnou polohu, ale na zemi. Pravý panel již nikdy není prázdný a ukazuje případně pohlednice v okolí.

## Nahrávání aktuální fotografie

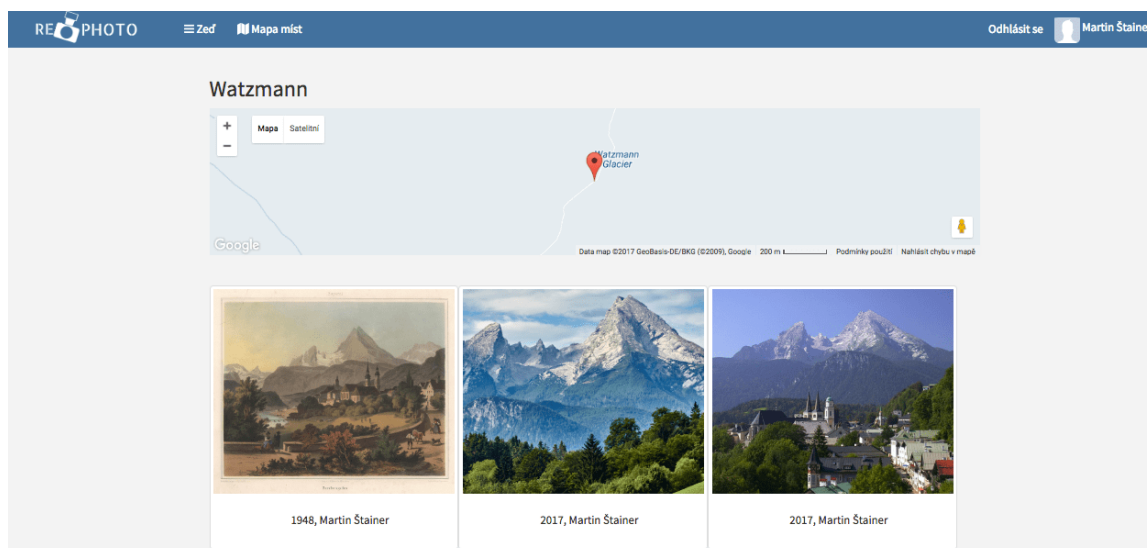
Po nahrání fotografie je zobrazen jednoduchý grafický editor, který umožňuje přibližovat, oddalovat a posouvat prolnuté fotografie. Tyto operace je možné provádět pomocí myši. Tento přístup přišel testerům intuitivní. Na druhou stranu chyběla možnost změny průhlednosti, která komplikovala přesnější shodu s historickou pohlednicí. Další kritikou bylo nevhodné zobrazení nahrané fotografie, která se nepřizpůsobovala velikosti okna.



Obrázek 5.2: Screenshot provedených úprav jednoduchého grafického editoru. Nově přibyly tlačítka pro změnu průhlednosti. Obrázek je po nahrání roztažen na velikost canvasu.

## Profil historické pohlednice

Důležitou částí je profil historické pohlednice, který zobrazuje pořízené snímky v časové ose. Uživatelé nechápali význam vytvořeného zobrazení a došlo k jeho celkovému předělání. Informace, které se zobrazovaly, nepřinášely uživateli nic přínosného. Zobrazení bylo zjednodušeno, informace zobrazeny v dialogovém okně přímo na mapě a ostatní přesunuty do nového profilu historické pohlednice.



Obrázek 5.3: Screenshot nového profilu pohlednice. Nově jsou zobrazeny fotografie vedle sebe bez zbytečných informací a označením lokace. Zobrazení by mělo pomoci k jednoduššímu srovnání.

## 5.2 Verze 1.0

Tato verze obsahovala opravy všech připomínek vytvořeného prototypu. Celkově byli uživatelé s aplikací mnohem spokojenější, ale stále jim vadily určité nedostatky při používání. Byl mírně upravena grafická stránka aplikace a přidány některé funkce.

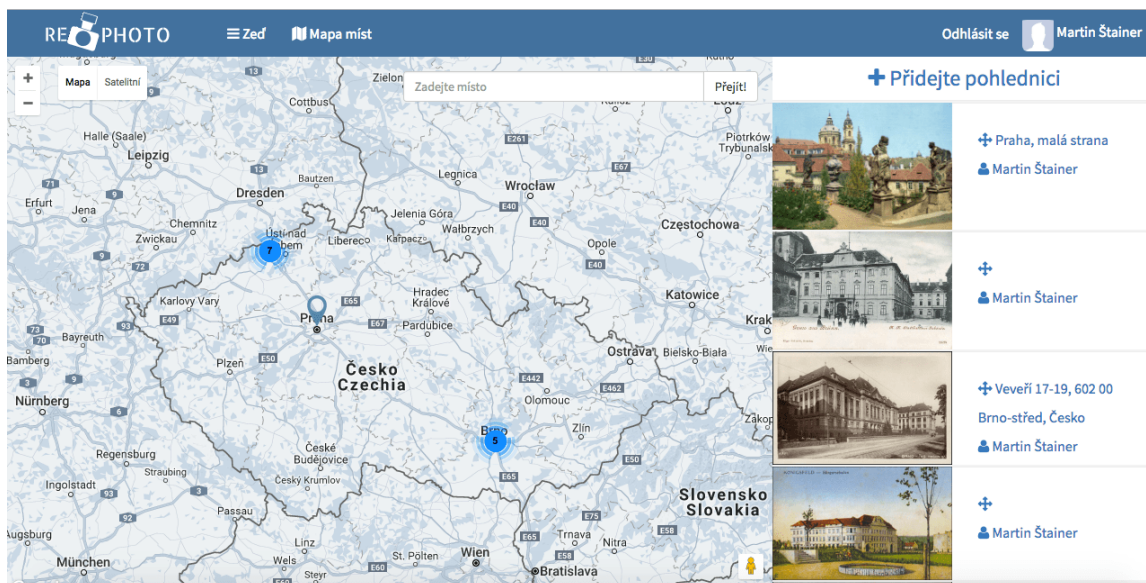
### Další úpravy mapy

Obrázek 5.2 ukazuje několik fotografií na jednom místě. Piny jsou překryté a uživatel nemá přehled o tom, kolik se v místě vyskytuje fotografií. Jako řešení bylo použito spojení bodů. Uživatel vidí počet fotografií a při kliknutí na ikonu je daná oblast přiblížena a fotografie zobrazeny. Uživatelé si rovněž stěžovali na nedostatečnou informovanost o načítání fotografií v pravém panelu. Tento problém byl vyřešen jednoduchou animací, která se spustí při spuštění AJAX požadavku a je ukončena po jeho dokončení.

### Úpravy nahrávání fotografií

Při ořezávání fotografie byla nahraná fotografie neustále průhledná. Většinu uživatelů to přišlo nepřirozené. Bylo změněno chování a fotografie je průhledná pouze při manipulaci s pozicí. Uživatelé navrhovali přidat tlačítka pro manipulaci s obrázkem.

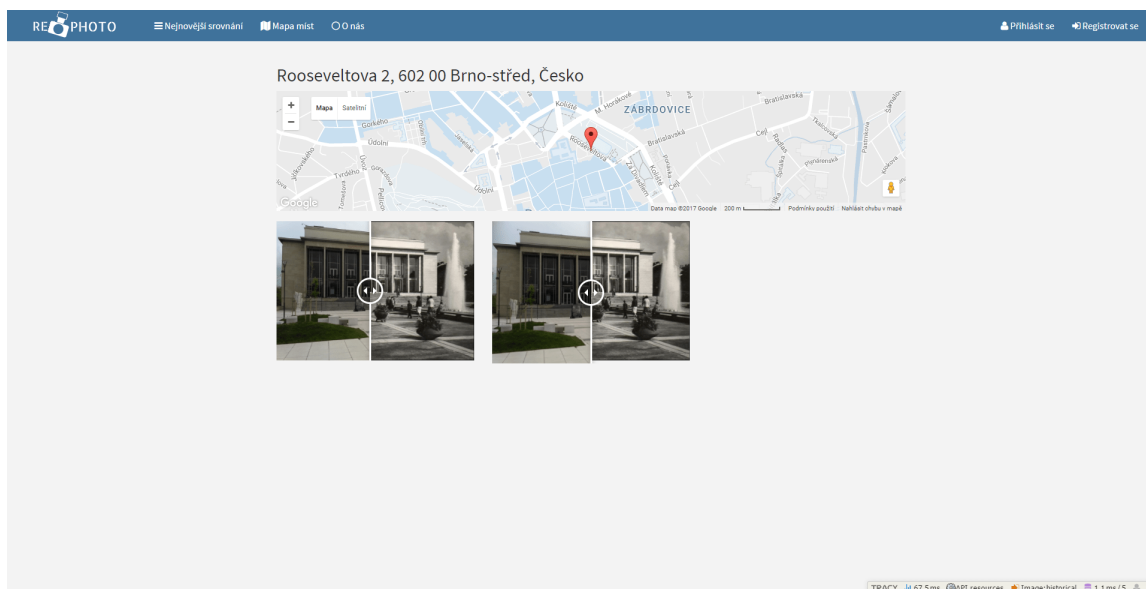




Obrázek 5.4: Screenshot úpravy spojování značek na mapě. Nově jsou fotografie spojovány do bodů a při kliknutí na ně je uživatel přiblížen na danou lokaci.

## Úpravy profilu historické pohlednice

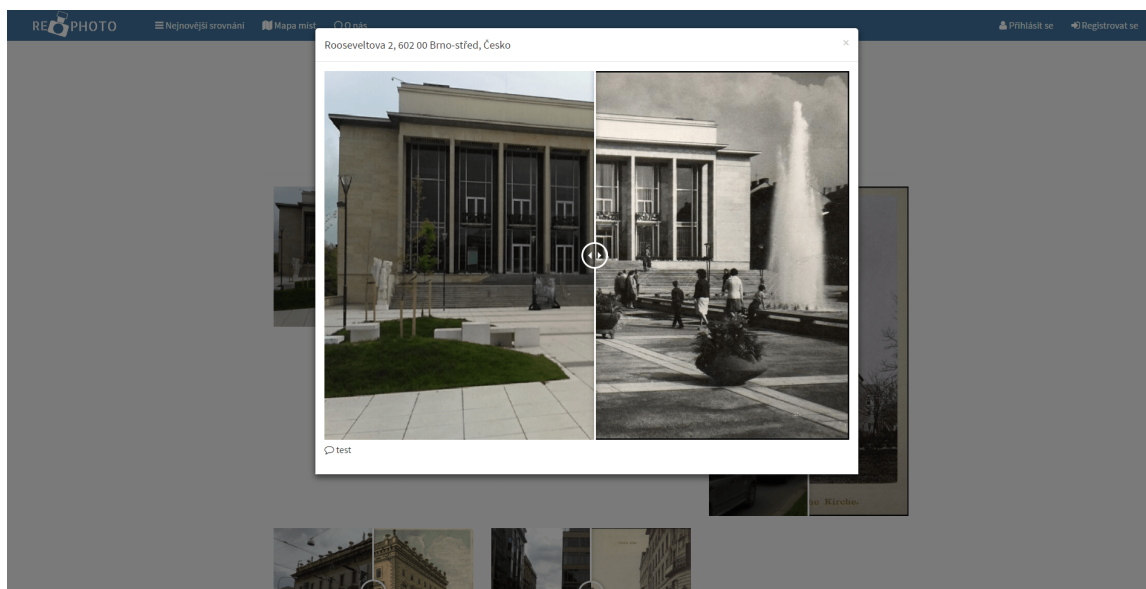
I když byl profil celkově předělán, uživatelům se aktuální podoba stále nelíbila. Srovnat fotografií lze pouze vlastníma očima. Z tohoto důvodu byla na stránku přidána možnost srovnání s historickou pohlednicí. Obrázek ukazuje nově vyfocené snímky seřazené v časové ose s možností překrývání.



Obrázek 5.5: Screenshot úpravy profilu historické pohlednice. Obrázek ukazuje přesnou lokaci a všechny pořízené snímky v daném místě.

### 5.3 Verze 1.1

Při zlepšování této verze byly přidány AJAXové požadavky pro lepší uživatelský zážitek. Na všech stránkovaních bylo upraveno chování jako "nekonečné načítání". Pokud návštěvník dosáhne aktuálního limitu výpisu, jsou mu načteny další příspěvky v AJAXovém volání. Ty se připevní na konec stránky a uživatel pohodlně pokračuje v prohlížení. Na profilu místa a uživatele byla přidána možnost zvětšování fotografií v modálním okně.



Obrázek 5.6: Screenshot úpravy profilu historické pohlednice. Obrázek ukazuje přesnou lokaci a všechny pořízené snímky v daném místě.



## Kapitola 6

# Závěr

Cílem této práce bylo navrhnout a vytvořit webovou aplikaci pro zveřejňování starých pohlednic s možností repeat photography. Aplikace byla zveřejněna na webové adrese <http://www.rephoto.cz> a nic nebrání jejímu používání. Před spuštěním byla nahrána menší databáze starých pohlednic z celé České republiky. Její přínos mohou ocenit např. fotografové, historikové nebo lidi se zájmem o své okolí. Serverová část aplikace byla implementována ve skriptovacím jazyce PHP s využitím rámce Nette. Pro uživatelské rozhraní byly použity jazyky HTML, CSS a JavaScript. Moderní vzhled, přístupnost a podporu všech moderních prohlížečů poskytl balíček Bootstrap.

Zadání této práce bylo splněno. Podařilo se mi seznámit se s problematikou repeat photography. Navrhl jsem a implementoval webovou aplikaci, která pokrývá všechny stanovené případy užití a následně ji testoval s uživateli a iterativně vylepšoval. Přínos této aplikace může být při sledování vizuálních změn v různých oborech vědy nebo může sloužit k pobavení uživatelů.

Tato práce mě naučila nejen používat moderní technologie pro vývoj webových aplikací, ale i aplikaci před implementací rozumně navrhnout. Nově získané poznatky při testování pro mne byly velice přínosné. Jako programátor jsem nad aplikací přemýšlel jiným způsobem než uživatel. Pro vývoj jsem zvolil agilní přístup, který se ukázal jako správný a pomohl k časnému dokončení aplikace.

Do budoucna plánuji aplikaci nadále rozvíjet. Aplikace je technicky připravena na nahrávání fotografií z celého světa, ale chybí jí jazykové mutace. Staré pohlednice bych rád sdružoval do kolekcí, tak aby uživatelé mohli prohlížet různé období v naší historii. Příkladem takové kolekce můžou být např. staré pohlednice z Druhé světové války. Dále bych chtěl zpracovávat další zpětné vazby uživatelů a implementovat jejich požadavky.

# Literatura

- [1] WEBB, Robert H., Diane E. BOYER a R. M. TURNER. *Repeat photography: methods and applications in the natural sciences*. Washington: Island Press, c2010. ISBN 978-159-7267-120.
- [2] MARGOLIS, Eric a L. PAUWELS. *The SAGE handbook of visual research methods*. Los Angeles: SAGE, 2011. ISBN 978-1-847-87556-3.
- [3] *Repeat photography project* [online]. [cit. 2017-05-17]. Dostupné z: <https://www.usgs.gov/centers/norock/science/repeat-photography-project>
- [4] *Historypin* [online]. [cit. 2017-05-14]. Dostupné z: <https://about.historypin.org/>
- [5] *PHP: Composer* [online]. Dayle Rees, 2016 [cit. 2017- 05-02]. Dostupné z: <https://leanpub.com/composer-php>
- [6] *Pět důvodů, proč zvolit Git* [online]. Karel Minařík, 2010 [cit. 2017- 05-02]. Dostupné z: <https://www.zdrojak.cz/serialy/pet-duvodu-proc-zvolit-git/>
- [7] PILGRIM, Mark. *HTML5: up and running*. Sebastopol, CA: O'Reilly, 2010. ISBN 978-059-6806-026.
- [8] *JavaScript* [online]. Dušan Janovský, 2010 [cit. 2017- 05-02]. Dostupné z: <https://www.jakpsatweb.cz/javascript/>
- [9] *Ajax* [online]. [cit. 2017- 05-02]. Dostupné z: [https://www.w3schools.com/xml/ajax\\_intro.asp](https://www.w3schools.com/xml/ajax_intro.asp)
- [10] SPURLOCK, Jake. *Bootstrap*. First edition. ISBN 978-1-4493-4391-0.
- [11] *Bootstrap customize* [online]. Mark Otto and Jacob Thornton and Bootstrap contributors, 2010 [cit. 2017- 05-02]. Dostupné z: <http://getbootstrap.com/customize/>
- [12] PETROUTSOS, Evangelos. *Google Maps: power tools for maximizing the API*. ISBN 978-0-071-82302-9.
- [13] *JQuery cookbook*. Cambridge: O'Reilly, c2010. ISBN 978-0-596-15977-1.
- [14] *PHP usage* [online]. W3Techs survey, 2016 [cit. 2017- 05-02]. Dostupné z: <https://w3techs.com/technologies/details/pl-php/all/all>
- [15] *How to choose PHP framework* [online]. Dr. Michael J. Garbade, 2008 [cit. 2017- 05-02]. Dostupné z: <https://opensource.com/business/16/6/which-php-framework-right-you>

- [16] RICHARDSON, Leonard a Michael AMUNDSEN. *RESTful Web APIs*. ISBN 978-1-449-35806-8.
- [17] *Nette Framework* [online]. Nette Foundation, 2016 [cit. 2017- 05-02]. Dostupné z: <https://nette.org>
- [18] *Nette Database* [online]. Nette Foundation, 2016 [cit. 2017- 05-02]. Dostupné z: <https://doc.nette.org/cs/2.4/database>
- [19] *Nette Forms* [online]. Nette Foundation, 2016 [cit. 2017- 05-02]. Dostupné z: <https://doc.nette.org/cs/2.4/forms>
- [20] *MVC* [online]. Stefano Borini [cit. 2017- 05-02]. Dostupné z: <https://github.com/stefanoborini/modelviewcontroller-src/>
- [21] *Úvod do vytváření GUI* [online]. Vítězslav Beran [cit. 2017- 05-02]. Dostupné z: <https://www.fit.vutbr.cz/study/courses/ITU/private/lectures/Intro/itu-gui-cs.uvod.pdf>

# Přílohy

## Příloha A

# Obsah přiloženého paměťového média

- readme.txt - popis instalace aplikace
- src - zdrojové kódy
- technicka-zprava.pdf - tato technická zpráva
- video.mp4 - krátké prezentační video
- poster.jpg - plakát