

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Novák** Jméno: **Kryštof** Osobní číslo: **459089**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra počítačů**  
Studijní program: **Softwarové inženýrství a technologie**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Aplikace pro vizualizaci pražských opevnění**

Název bakalářské práce anglicky:

**Application for visualization of Prague fortifications**

Pokyny pro vypracování:

Navrhněte a implementujte interaktivní webovou aplikaci pro vizualizaci vývoje pražských opevnění v čase. Aplikace umožní vkládat, doplňovat, editovat použité historické dokumenty, mapy, náčrty s časovou stopou a s důrazem na maximální kartografickou přesnost a vzájemnou souvislost. Významné body opevnění ponесou i vizuální a textovou informaci. Uživatelská část umožní jak statické procházení jednotlivými časovými body, ale i možnost animovaného průchodu vývoje opevnění v určitém časovém úseku s možností parametrizace časových kroků a sledovaného území. Navrhněte jednotlivé kroky implementace, aplikaci implementujte. Navrhněte systém testování a výslednou aplikaci otestujte. Výsledky zhodnoťte.

Seznam doporučené literatury:

Vladimír Kupka, Pražská opevnění, Libri, 2008.  
Vědecká knihovna v Olomouci, Mapy staré Prahy k letům 1200, 1348 a 1419, <http://www.digitalniknihovna.cz/vkol/view/uuid:d1524bea-2ff8-4116-9c26-740392046f00?page=uuid:8aa1f070-f1e9-495e-ac18-f30affc8f2c1>.  
Encyclopædia Britannica, Maps and mapping - map projections, 15 ed., Encyclopædia Britannica Inc., 1984.  
German Carrillo, Web mapping client comparison v.6, <http://geotux.tuxfamily.org/index.php/en/geo-blogs/item/291-comparacion-clientes-web-v6>.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**doc. Ing. Ivan Jelínek, CSc., kabinet výuky informatiky FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **13.02.2020**

Termín odevzdání bakalářské práce: **22.05.2020**

Platnost zadání bakalářské práce: **30.09.2021**

\_\_\_\_\_  
doc. Ing. Ivan Jelínek, CSc.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačů

## Aplikace pro vizualizaci pražských opevnění

**Kryštof Novák**

Vedoucí práce: doc. Ing. Ivan Jelínek, CSc.  
Obor: Softwarové inženýrství a technologie  
Květen 2020



## Poděkování

Děkuji doc. Ing. Ivanu Jelínkovi, CSc. za vedení bakalářské práce. Mé poděkování patří též Bc. Zdeňku Vlachovi za poskytnuté konzultace a VHÚ Praha za spolupráci při získávání podkladů pro tuto práci. Děkuji také své rodině za morální podporu.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 12. května 2020

I declare that this work is all my own work and I have cited all sources I have used in the bibliography.

Prague, May 12, 2020

## Abstrakt

Tato práce se zabývá vytvořením interaktivní webové aplikace pro vizualizaci vývoje pražských opevnění - VPO.

Ve své první části obsahuje rešerše podobných existujících aplikací a rešerši využitelných historických a mapových zdrojů. Dále se věnuje vlastnostem kartografických zobrazení a z nich vyplývajících možností a omezení pro uvažovanou aplikaci. Poznatky jsou následně uplatněny v technickém návrhu aplikace, na který navazuje výběr softwarových programovacích prostředků pro samotnou realizaci. Za využití mapové komponenty LeafletJS a PHP frameworku Laravel je aplikace následně implementována a pokusně veřejně nasazena. Poslední část práce je věnována testování aplikace a z něj vycházejícím úpravám.

**Klíčová slova:** bakalářská práce, pražské opevnění, mapová aplikace, webová aplikace, Laravel, Php, LeafletJS

**Vedoucí práce:** doc. Ing. Ivan Jelínek, CSc.

## Abstract

This bachelor thesis deals with design and implementation of Interactive web application for visualizing of development of Prague fortifications - VPO.

Its first part presents a survey of similar existing applications and available historical and map resources. Further, it focuses on properties of different cartographic projections and the resulting possibilities and limitations for the application in question. These findings are then applied in the technical design of the application, which is followed by the selection of software programming tools for the implementation. The application is subsequently implemented and experimentally deployed using the map component LeafletJS and the PHP framework Laravel. The last part of this work focuses on testing of the application and on the modifications made based on testing results.

**Keywords:** Bachelor Thesis, Prague Fortification, Map Application, Web Application, Laravel, Php, LeafletJS

**Title translation:** Application for visualization of Prague fortifications

## Obsah

|  |           |  |           |
|--|-----------|--|-----------|
| <b>1 Úvod</b>                          | <b>1</b>  | <b>7 Testování</b>                     | <b>35</b> |
| <b>2 Analýza</b>                       | <b>3</b>  | 7.1 Automatické testy .....            | 35        |
| 2.1 Existující aplikace .....          | 3         | 7.2 Manuální testy .....               | 35        |
| 2.2 Historické a mapové zdroje .....   | 6         | 7.2.1 Různé internetové prohlížeče     | 36        |
| 2.2.1 Historické mapy .....            | 6         | 7.2.2 Mobilní zařízení .....           | 36        |
| 2.2.2 Současné mapy .....              | 8         | 7.3 Uživatelské testování .....        | 37        |
| 2.2.3 Významné body .....              | 8         | <b>8 Závěr</b>                         | <b>39</b> |
| 2.3 Kartografická zobrazení a přesnost |           | <b>Literatura</b>                      | <b>41</b> |
| map .....                              | 9         | <b>A Návrhy uživatelského rozhraní</b> | <b>45</b> |
| 2.3.1 Kartografická zobrazení .....    | 9         | <b>B Ukázky výsledné aplikace</b>      | <b>49</b> |
| 2.3.2 Problematika pozicování          |           | <b>C Automatické testy</b>             | <b>53</b> |
| historických map .....                 | 9         |  |           |
| 2.4 Datové formáty map .....           | 10        |  |           |
| <b>3 Návrh aplikace</b>                | <b>13</b> |  |           |
| 3.1 Funkční požadavky .....            | 13        |  |           |
| 3.2 Struktura aplikace .....           | 13        |  |           |
| 3.3 Uživatelské rozhraní .....         | 14        |  |           |
| 3.4 Datový model .....                 | 14        |  |           |
| <b>4 Softwarové prostředky</b>         | <b>19</b> |  |           |
| 4.1 Mapová komponenta .....            | 19        |  |           |
| 4.1.1 Komponenty využívající Adobe     |           |  |           |
| Flash .....                            | 19        |  |           |
| 4.1.2 Komponenty využívající           |           |  |           |
| HTML5 + Java Script .....              | 19        |  |           |
| 4.2 Prototyp .....                     | 21        |  |           |
| 4.3 Technologie frontendu .....        | 21        |  |           |
| 4.4 Technologie backendu .....         | 22        |  |           |
| 4.4.1 PHP frameworky .....             | 22        |  |           |
| 4.5 Technologie databáze a ukládání    |           |  |           |
| dat .....                              | 23        |  |           |
| <b>5 Implementace</b>                  | <b>25</b> |  |           |
| 5.1 Vývojové nástroje .....            | 25        |  |           |
| 5.2 Technologie spojené s Laravelem    | 25        |  |           |
| 5.3 Rozšiřující knihovny .....         | 26        |  |           |
| 5.4 Struktura projektu .....           | 26        |  |           |
| <b>6 Výsledná aplikace</b>             | <b>29</b> |  |           |
| 6.1 Ukázky a popis výsledné aplikace   | 29        |  |           |
| 6.2 Veřejně spuštěná aplikace .....    | 32        |  |           |
| 6.2.1 Technická specifikta zvoleného   |           |  |           |
| hostingu .....                         | 32        |  |           |
| 6.3 Vlastní spuštění aplikace .....    | 33        |  |           |
| 6.3.1 Lokální instalace .....          | 33        |  |           |
| 6.3.2 Konfigurace aplikace .....       | 34        |  |           |

## Obrázky

|  |    |
|--|----|
| 2.1 Aplikace IPR Praha - Archiv územních plánů. Zobrazení za pomoci překrývání map. ....   | 4  |
| 2.2 Interaktivní mapa opevnění ČSR. Zobrazení za pomoci markerů. ....  | 5  |
| 2.3 Aplikace <i>Scroll NewYorkCity</i> . Zobrazení posouváním map. ....  | 6  |
| 2.4 Pokus o překrytí Seutterovy mapy (1650) a současné mapy Prahy s využitím nástroje georeferencer.com  | 7  |
| 2.5 Zkreslení Mercatorova zobrazení. Autor Stefan Kühn, zdroj wikimedia.org .....  | 10 |
| 3.1 Návrh veřejné části uživatelského prostředí aplikace .....   | 14 |
| 3.2 Návrh administrátorské části uživatelského prostředí aplikace - přidání významného bodu .....  | 15 |
| 3.3 Relační model databáze aplikace. (FK = cizí klíč) .....  | 16 |
| 4.1 Vztahy free/opensource webových mapových klientů, 2012. Převzato z <a href="http://geotux.tuxfamily.org">geotux.tuxfamily.org</a> [Car] ....       | 20 |
| 4.2 Prototyp aplikace s využitím LeafletJS. ....   | 22 |
| 5.1 Struktura důležitých souborů projektu. ....  | 27 |
| 6.1 Úvodní obrazovka VPO se základní podkladovou mapou a vybranou celou časovou osou. ....   | 29 |
| 6.2 Úvodní obrazovka VPO se satelitní podkladovou mapou a vybraným úsekem časové osy (1450-1750). Otevřen detail významného bodu "Koňská brána." ..... | 30 |
| 6.3 Úvodní obrazovka VPO se satelitní podkladovou mapou a překryvnou rekonstrukční mapou. ....   | 31 |
| 6.4 Administrátorská sekce VPO - seznam významných bodů. ....  | 31 |
| 6.5 Administrátorská sekce VPO - georeferencování vkládané mapy. . .   | 32 |
| 7.1 VPO na displeji mobilního telefonu. ....   | 37 |
| A.1 Návrh administrátorské části uživatelského prostředí aplikace - seznam map. ....   | 46 |
| A.2 Návrh administrátorské části uživatelského prostředí aplikace - seznam významných bodů. ....   | 46 |
| A.3 Návrh administrátorské části uživatelského prostředí aplikace - přidání nové mapy. ....  | 47 |
| A.4 Návrh administrátorské části uživatelského prostředí aplikace - seznam administrátorů. ....  | 47 |
| B.1 Veřejná část uživatelského prostředí aplikace - detail významného bodu, překryvná mapa - Jüttnerův plán 1816. ....                                 | 50 |
| B.2 Administrátorská část uživatelského prostředí aplikace - přidání významného bodu. ....   | 50 |
| B.3 Administrátorská část uživatelského prostředí aplikace - galerie obrázků k významnému bodu. ....   | 51 |
| B.4 Administrátorská část uživatelského prostředí aplikace - detail významného bodu. ....  | 51 |
| B.5 Veřejná část uživatelského prostředí aplikace - spuštěná animace - rok 1270, překryvná mapa - Přemyslovské opevnění. ....                          | 51 |
| B.6 Veřejná část uživatelského prostředí aplikace - spuštěná animace - rok 1825, překryvná mapa - Barokní opevnění. ....                               | 52 |



## Tabulky

|  |    |
|--|----|
| 2.1 Seznam opatřených map. ....                                    | 8  |
| 3.1 Funkční požadavky. ....  | 17 |
| 7.1 Internetové prohlížeče, ve kterých<br>byla VPO testována. .... | 36 |
| C.1 PHPUnit testy vytvořené pro<br>VPO. ....                       | 54 |



# Kapitola 1

## Úvod

Cílem této práce je vytvoření Interaktivní aplikace pro vizualizaci vývoje pražského opevnění.

Motivací k realizaci tohoto projektu je neexistence veřejně přístupného interaktivního zdroje, který by zájemcům o historii ukázal vývoj pražských opevnění - od opevnění jednotlivých hradišť, přes opevnění Starého a později Nového Města, až po demolice v 19. století - a umožnil jejich zobrazení a lokalizaci v současné podobě moderní Prahy. Tato aplikace bude navíc zobrazovat detailní informace k jednotlivým významným místům opevnění a bude uživatelům umožňovat vkládat další obsah.

V první části této práce (**druhá kapitola**) budou provedeny nezbytné analýzy a rešerše - existujících aplikací, dostupných historických a moderních mapových podkladů a dodatečných informací k pražskému opevnění. Další část analýz bude věnována studiu problematiky mapových zobrazení vzhledem k mojí aplikaci. Důležitá bude také rešerše možností pozicování map a způsobů reprezentace map v počítačích.

Na základě poznatků z předchozí kapitoly bude ve **třetí kapitole** vypracován návrh aplikace - funkční požadavky, databázový model, technická struktura a uživatelské rozhraní.

**Čtvrtá kapitola** této práce je poté věnována výběru vhodných technologií pro samotnou realizaci aplikace - programovacích jazyků, frameworků a mapové komponenty.

V **páté kapitole** budou popsány detaily implementace aplikace a dodatečně použité nástroje a technologie.

Tématem **šesté kapitoly** je popis a ukázky výsledné aplikace, včetně návodu na lokální instalaci a odkazů na veřejně nasazenou verzi i na zdrojové kódy.

Poslední - sedmá kapitola se věnuje testování aplikace - automatickému i manuálnímu a změnám, které budou na základě zjištěných výsledků provedeny.



## Kapitola 2

### Analýza

V této kapitole budou provedeny nezbytné rešerše pro návrh a vývoj aplikace.

Jmenovitě v první podkapitole bude proveden průzkum již existujících aplikací. Cílem je mimo jiné najít inspiraci pro mou aplikaci, zejména po stránce technického řešení a uživatelského rozhraní.

Tématem druhé podkapitoly jsou datové podklady projektu, tj. historické mapy, vhodná mapa moderní Prahy a zdroje dalších informací k pražskému opevnění.

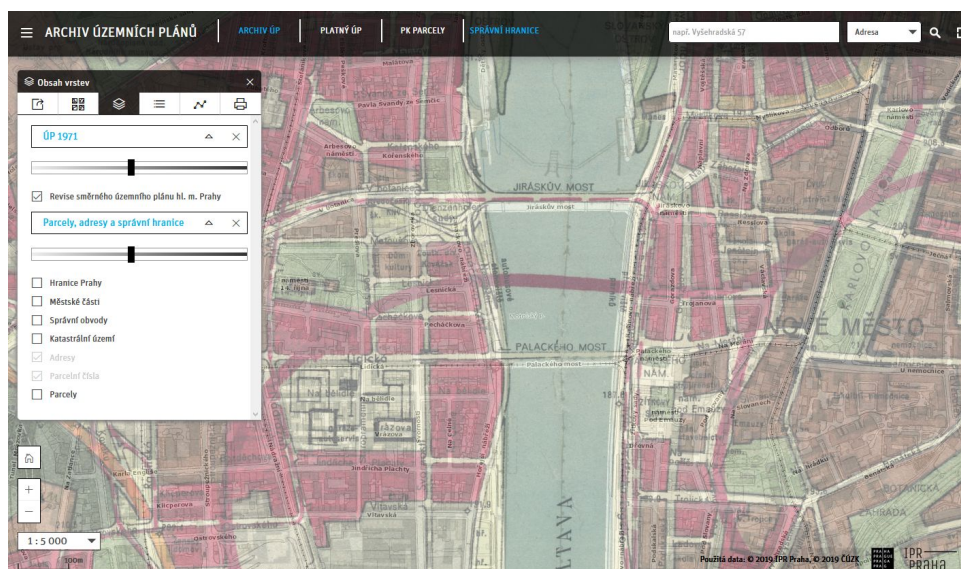
Třetí část je věnována studiu problematiky mapových zobrazení vzhledem k mojí aplikaci. Zabývá se také možnostmi pozicování map, které je důležité pro vzájemné překrývání map.

Poslední podkapitola se zabývá digitálními způsoby reprezentace map.

### 2.1 Existující aplikace

Jak již bylo zmíněno v úvodu, aplikace se stejným zaměřením - tedy umožňující vizualizaci vývoje pražského opevnění - v současné době neexistuje. Existuje však řada více či méně příbuzných aplikací, kterými se lze inspirovat ve způsobu zobrazení map, technickém řešení či uživatelském rozhraní. Zde uvádím několik aplikací, které jsou mému projektu blízké tématem, nebo zastupují konkrétní způsob zobrazení historických dat. Softwarové aspekty jednotlivých aplikací budou rozebrány v kapitole 4.1 Softwarové prostředky - Mapová komponenta.

**IPR Archiv ÚP.** <http://app.iprpraha.cz/apl/app/archivup/>  
Archiv územních plánů (1947-2009) Institutu plánování a rozvoje. Z pohledu zobrazení (více) map se jedná o klasickou aplikaci: v podkladu je (většinou současná) mapa, přes kterou je zobrazena částečně průhledná historická mapa - pro tento způsob je dále v práci používán termín "překrývání map". Aplikace Archiv ÚP tedy uživateli dává na výběr seznam historických územních plánů, které se následně překrývají s mapou současnou. Zároveň si lze volit stupeň průhlednosti historických map. Příklad vzhledu uživatelského rozhraní je na obrázku 2.1

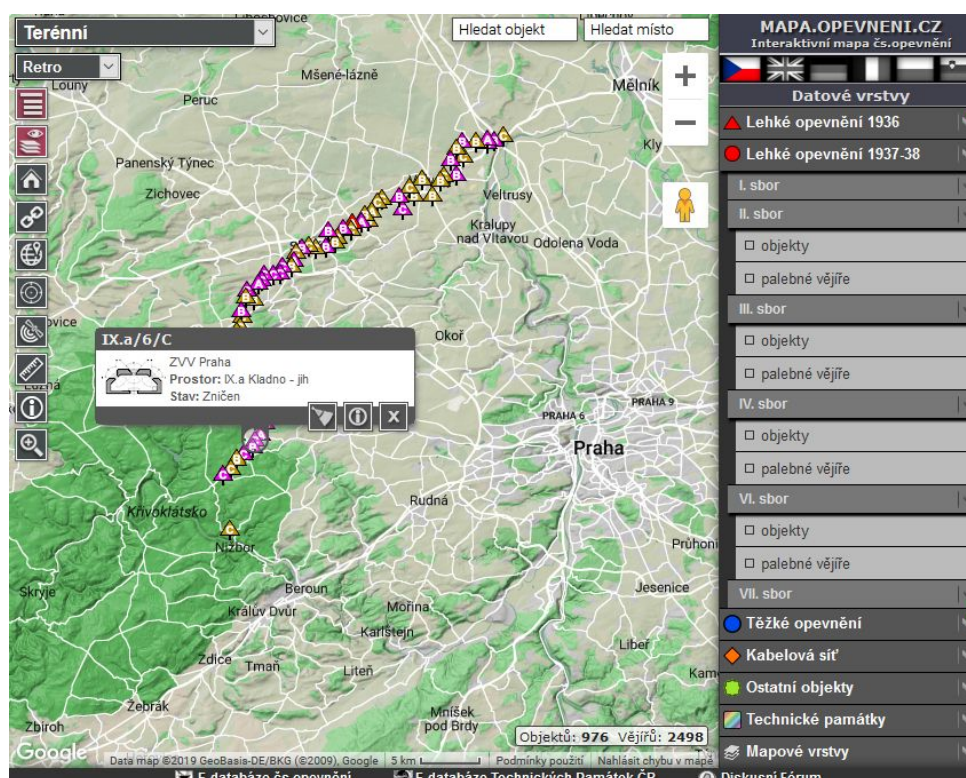


**Obrázek 2.1:** Aplikace IPR Praha - Archiv územních plánů. Zobrazení za pomoci překrývání map.

**Kontaminovaná místa ČR - Cenia.** <https://kontaminace.cenia.cz/>  
 Mapa kontaminovaných míst ČR. Z pohledu zobrazení map opět poměrně klasická aplikace - mapy jsou přes sebe překrývány s volitelnou průhledností. Kontaminovaná místa jsou zobrazena na ortofotomapě ČSR z 50. let. Mimo jiné lze ale v aplikaci například zobrazit III. Vojenské mapování. Uživatelské prostředí je lehce chaotické - některé ovládací prvky překrývají mapu, jiné jsou umístěny vedle ní. Klasické mapy mají každá svůj vlastní posuvník viditelnosti, ortofoto mapy mají jeden společný.

**Interaktivní mapa čs. opevnění.** <https://mapa.opevneni.cz/>  
 Projekt mapa.opevneni.cz si klade za cíl přehledně zobrazit opevnění Československé republiky vystavěné v letech 1935-1938. V některých případech ale nemusí být vhodné prezentovat data historickými mapami - například kvůli jejich nepřehlednosti, nedochování/neexistenci, nebo, jako v tomto případě, požadavku na větší interaktivitu. Autoři proto k zobrazení použili v podkladu Google mapu ČR (ve výchozím nastavení), do které jsou vloženy jednotlivé body opevnění (tzv. řopíky, pěchotní sruby...) jako markery. Uživatel si jednotlivé skupiny markerů může libovolně zapínat a tím například zobrazit jen lehké opevnění postavené v roce 1936. Vizualizace dat za pomoci markerů bude v mém projektu také částečně využito - budou jimi vyznačeny významné body opevnění (kapitola 2.2.3). Zobrazeno na obrázku 2.2

**Dvě Prahy.** <http://www.dveprahy.cz/>  
 Aplikace vyvinutá IPR Praha, umožňující srovnání dnešní a historické podoby města Prahy (od roku 1816 do současnosti). Namísto klasického překrývání (jaké používá IPR Archiv, Cenia...) jsou mapy umístěny vedle sebe. Jedná

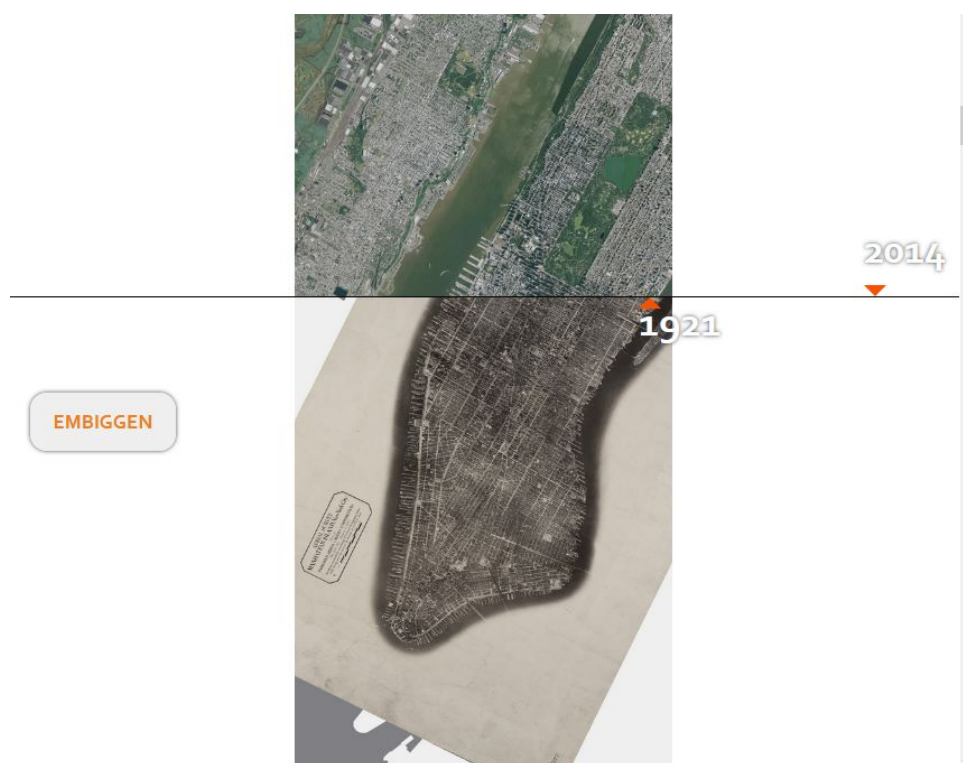


**Obrázek 2.2:** Interaktivní mapa opevnění ČSR. Zobrazení za pomoci markerů.

se o koncept vhodný pro zkoumání a vzájemné porovnávání dvou map, při porovnávání více map může být však ztracena přehlednost a pokud je navíc potřeba v tomto zobrazení používat markery, bude uživatel pravděpodobně zcela dezorientován. Pro naše účely - přímé demonstrování vývoje - bude proto vhodnější použít klasickou metodu překrývání.

**Scroll N.Y.C..** <https://mauriciogirardo.com/lab/scrollnyc/>  
New York City - Manhattan na historických mapách. Tato aplikace používá inovativní způsob zobrazení map - uživatel posouvá kolečkem myši (scroll), čímž se pod sebe podsouvají jednotlivé mapy. Viz Obrázek 2.3.

**Georeferencer.** [www.georeferencer.com](http://www.georeferencer.com)  
Na rozdíl od předchozích příkladů - jednoúčelově zaměřených aplikací - je Georeferencer univerzální systém pro vytváření a zejména pozicování map (=georeferencování). Běžný uživatel si do aplikace může nahrát vlastní mapu (například scan historické mapy) a přesně ji zapozicovat. Velké instituce tuto aplikaci, v různých úpravách, mohou využívat k prezentaci celých svých mapových sbírek. V základu staví na georeferencer.com například [www.staremapy.cz](http://www.staremapy.cz) (projekt Univerzity Karlovy), The British Library Map Collection ([www.bl.uk/georeferencer](http://www.bl.uk/georeferencer)) nebo mapová sbírka University of London. Georeferencer je založen na MapTiler/OpenLayers komponentách a je českého původu.



Obrázek 2.3: Aplikace *Scroll New York City*. Zobrazení posouváním map.

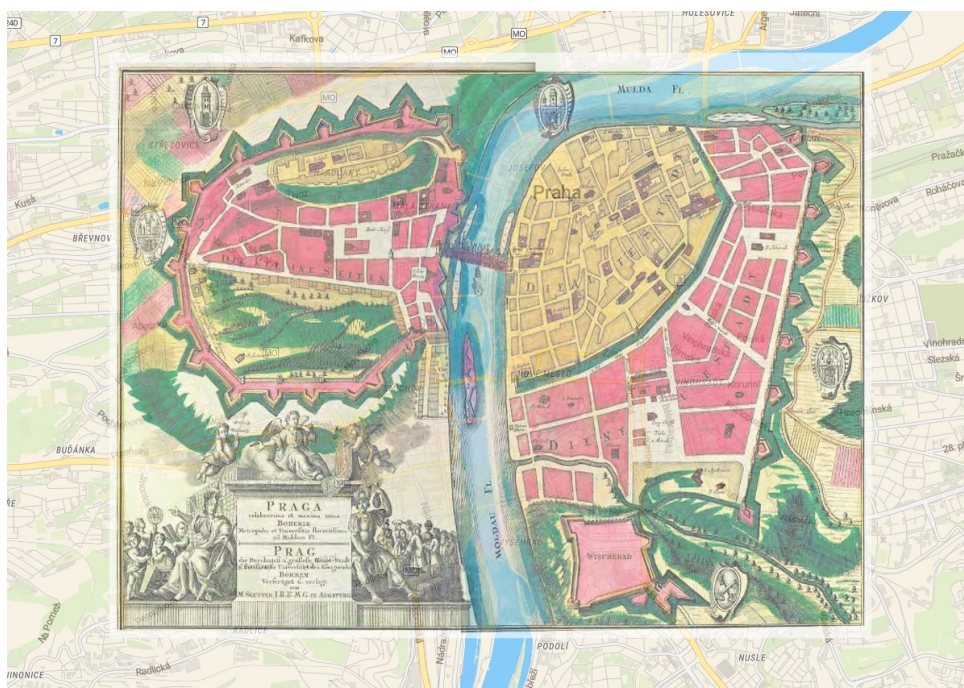
## 2.2 Historické a mapové zdroje

Pro prezentování vývoje pražských opevnění využijeme překrývání historických map s nastavitelnou průsvitností přes podkladovou současnou mapu. Do vizualizace budou navíc vloženy tzv. markery označující významná místa opevnění. Pro demonstrování základních funkcí naší aplikace je tedy nutné zajistit alespoň několik historických map pokrývajících co nejširší časové období, vybrat dostupné současné mapy, najít zdroje informací k významným bodům a shromáždit k nim základní údaje.

### 2.2.1 Historické mapy

Původním záměrem bylo demonstrovat výstavbu a vývoj Pražského opevnění prolínáním historických map. Rešerší literatury [HA17] se však ukázalo, že dostatečně přesné mapy, které by mohly být použity pro vzájemné překrývání a překrývání se současnými mapami, začaly být vytvářeny až po rozvoji trigonometrického mapování v 18. století. Starší mapy jsou většinou kartograficky velmi nepřesné a prolnout je proto se současnými mapami je téměř nemožné, viz Obrázek 2.4. Je třeba mít na paměti, že staré mapy "... mohou poskytovat o dávných městech poněkud zavádějící informace, neboť první kartografové často čerpali spíše z oblíbených legend než z bezprostředního pozorování." [HA17] Ve





**Obrázek 2.4:** Pokus o překrytí Seutterovy mapy (1650) a současné mapy Prahy s využitím nástroje georeferencer.com

většině map, dostupných (tehdejší) veřejnosti, byly navíc vojensky důležité objekty záměrně zobrazovány jen velmi vágně, popřípadě vůbec. V Českých zemích prvním souborem map, vytvořeným (částečně) s využitím "moderních" trigonometrických metod, bylo První vojenské mapování vytvořené v 60. letech 19. století. Nicméně za první kartograficky skutečně přesný plán města Prahy je často označován až Jüttnerův plán, vydaný roku 1820. [HA17] Uvážíme-li, že k poslední rozsáhlejší výstavbě hradeb došlo v první čtvrtině 18. století [Kup08], nelze, pro účely uvažované aplikace, rozvoj hradeb ilustrovat za pomoci historických map.

Z výše uvedených důvodů bylo rozhodnuto použít v této aplikaci historické mapy pouze pro demonstraci stavu hradeb v devatenáctém století. Období starší - tedy nezaznamenané na historických a zároveň přesných mapách - je překlenuto za pomoci tří nově vytvořených skic (13. století, 14. století, 17. století - Archeologická mapa, Ivana Boháčová) a dvou nových rekonstrukčních map (Mapy staré Prahy, Václav Vladivoj Tomek). Seznam map a mapových skic shromážděných a použitých v rámci této práce je uveden v tabulce 2.1.

Jako potenciální zdroj dalších map byl kontaktován Vojenský historický ústav. Ten disponuje sbírkou 507 plánů opevnění Prahy [VHA], bohužel se ale v naprosté většině jedná o velmi detailní plány (stavební výkresy, půdorysy...) nepoužitelné pro účely navrhované aplikace.

| Název mapy                         | Rok zobrazení | Autor                  | Zdroj   |
|------------------------------------|---------------|------------------------|---------|
| <i>historické mapy</i>             |               |                        |         |
| Jüttnerův plán                     | 1816          | Josef Jüttner          | [Wika]  |
| Prague for Travelers               | 1858          | John Murray            | [Vá]    |
| Císařský otisk stabilního katastru | 1840          | Habsburská monarchie   | [Vá]    |
| III. vojenské mapování             | 1880          | Habsburská monarchie   | [VÚ]    |
| <i>rekonstruované mapy</i>         |               |                        |         |
| Archeologická mapa                 | 1200          | Ivana Boháčová         | [Pa]    |
| Archeologická mapa                 | 1300          | Ivana Boháčová         | [Pa]    |
| Archeologická mapa                 | 1650          | Ivana Boháčová         | [Pa]    |
| Mapy staré Prahy                   | 1200          | Václav Vladioj Tomek   | [kvO]   |
| Mapy staré Prahy                   | 1388          | Václav Vladioj Tomek   | [kvO]   |
| Staré město                        | 1250          | Vladimír Kupka         | [Kup08] |
| Gotická Praha pol. 13.stol.        | 1250          | Školní atlas čs. dějin | [UK]    |

**Tabulka 2.1:** Seznam opatřených map.

### 2.2.2 Současné mapy

Aby bylo možné historické mapy přesně pozicovat a hlavně porovnávat se současností, je zapotřebí mít adekvátní referenční mapu. Dobrý přehled největších poskytovatelů současných mapových dat lze nalézt na Wikipedii [Wikb]. Vyplývá z něj, že nejpoblárnější služby - Google Maps a Apple Maps použitelnost a zejména modifikování svých map omezují. Nejlepší variantou se proto jeví použití **OpenStreetMaps**.

OpenStreetMaps jsou šířeny pod Open Database License (dříve Creative Commons), která umožňuje mapová data volně sdílet, používat a modifikovat [Com]. Jejich data používá například Facebook, Craigslist, Seznam a mnoho dalších. Díky volné licenci existuje řada poskytovatelů a různých úprav těchto map. Pro naše účely byla konkrétně vybrána mapa Hydda<sup>1</sup> Osm/SE - obsahující jen obrysy domů a satelitní mapa Esri/World Imagery.

### 2.2.3 Významné body

Aplikace bude, kromě map, zobrazovat i tzv. Významné body (ve webové mapové terminologii se většinou používá pojem Point of Interest). Například budou takto označeny všechny bastiony, brány, hrady nebo například důležité části hradeb. Texty a fotografie budou čerpány zejména ze článku na serveru Válka.cz - Pevnost Praha [Vá] Karla Oktábce, který přehledně zpracoval informace ke všem bastionům i ke většině bran Pražského opevnění. Dodatečné informace a obrázky budou použity z publikace Vladimíra Kupky Pražská opevnění [Kup08] a z encyklopedie Wikipedia.org. Aby mohly být Významné

<sup>1</sup>Mapa Hydda musela být, kvůli dlouhodobému výpadku serverů v květnu 2020, dočasně nahrazena mapou Carto.Positron.

body použity pro animaci vývoje, bude u nich muset být, kromě popisku a obrázku uveden i rok vzniku, případně rok zániku.

## 2.3 Kartografická zobrazení a přesnost map

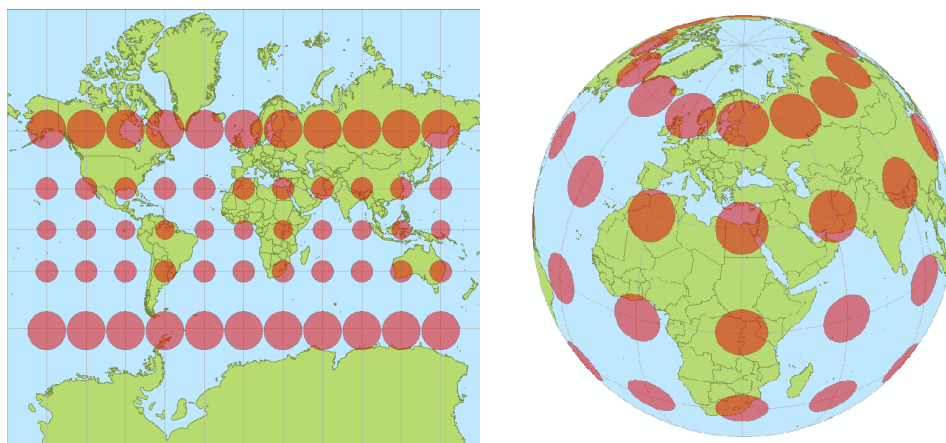
Aby mohly být mapy správně zobrazovány a aby bylo možné s nimi v digitální podobě pracovat, je potřeba nejdříve pochopit jejich základní technické principy - co jsou to kartografická zobrazení, jaké jsou mezi nimi rozdíly a jakým způsobem je třeba naši aplikaci různým zobrazením přizpůsobit. Dále vyvstává otázka jak (historické) mapy pozicovat, aby následně mohly být použity k vzájemnému překrývání. Těmito problémy, ve vztahu k naší aplikaci, se zabývá následující podkapitola.

### 2.3.1 Kartografická zobrazení

Základní otázkou při vytváření jakékoliv mapy (i jen části) Země je, jak přenést povrch Země (geoidu) do roviny. I pokud je totiž geoid zjednodušen na elipsoid, případně kouli, jejich povrch není rozvinutelný do roviny. Problematika tedy spočívá v převedení tohoto povrchu na těleso, jehož povrch rozvinutelný do roviny je. Dle způsobu řešení tohoto problému lze členit kartografická zobrazení například na: Azimutální - které zobrazuje povrch koule (a jemu příbuzné) přímo na rovinu, příkladem je Gnómonická projekce; Kuželové - zobrazení na plášť kužele; Válcové - zobrazení na plášť válce, například Mercatorovo; a různá speciální další [Bri84]. Jednoznačně nejrozšířenějším zobrazením, zejména pro použití na webu, je v dnešní době WebMercator (také známo jako Pseudo-Mercator nebo Google Mercator). Jedná se o úpravu Mercatorova zobrazení, které však Zemi nepovažuje za elipsoid, ale za kouli (na rozdíl od původního Mercatorova zobrazení) [Opea]. Tím bylo dosaženo zjednodušení výpočtů, ale i snížení přesnosti. Povrch koule je následně přenesen na povrch válce, čímž vznikne pravoúhlá síť poledníků a rovnoběžek, která má poměrně dobrou přesnost kolem rovníku, ale zhoršuje se při přibližování k pólům - viz obrázek 2.5. Toto zobrazení nicméně využívají téměř všechny webové služby poskytující mapová data, jako například Google Maps, OpenStreetMaps, Bing Maps, ČÚZK Ortofoto ČR a další. Podkladová mapa v tomto projektu tedy bude také využívat WebMercator zobrazení. Nutno ale podotknout, že při tak "malých" rozměrech (z hlediska mapování Země) jaké má město Praha, lze pro naše účely zanedbat, v jakém zobrazení byla mapa vytvořena [Wick].

### 2.3.2 Problematika pozicování historických map

Zatímco pro správné překrývání moderních map stačí znát zeměpisné souřadnice dvou protilehlých rohů mapy, u historických map se situace komplikuje. Za prvé zeměpisné souřadnice map většinou neznáme a za druhé - staré mapy často nejsou příliš přesné. První problém (nalezení souřadnic mapy)



**Obrázek 2.5:** Zkreslení Mercatorova zobrazení. Autor Stefan Kühn, zdroj wiki-media.org

lze vyřešit implementací grafického nástroje, který umožní překrýt vkládanou mapu s mapou, jejíž souřadnice známe. Prostým posouváním, lineární změnou měřítka a případnou rotací lze poté mapy na sebe zarovnat a zjistit tak zeměpisné souřadnice vkládané mapy. Druhý problém (nepřesnosti) se snaží různé aplikace řešit zpětnou "rekonstrukcí" starých map. Například server [georeferencer.com](http://georeferencer.com) nabízí nástroj, kterým lze postupně vybírat stejné body na současné (přesné) a historické (nepřesné) mapě. Dle těchto bodů se postupně stará mapa deformuje tak, aby odpovídala mapě současné (jedná se o Helmertovu podobnostní transformaci [Bal]). Pro víceméně přesné mapy ale není takto komplikovaného pozicování potřeba a stačí pouze metoda dvou bodů, která může být případně doplněná o operaci otočení. Naopak u silně nepřesných map lze tímto způsobem docílit spíše jen orientačních výsledků, nehledě navíc na problematický obsah těchto map, jak již bylo zmíněno v kapitole 2.2.1. Pro účely naší aplikace tedy bude použito pouze základního pozicování souřadnicemi dvou protilehlých rohů, doplněného o možnost zkosnění (tedy ve výsledku budou uloženy souřadnice tří bodů). Takto umístěné mapy bude následně možné překrývat se současnými mapami.

## 2.4 Datové formáty map

Pro návrh aplikace pracující s mapami je nezbytné vědět, jakým způsobem se mapová data v počítačích ukládají, zobrazují a jak se případně přenášejí po internetu.

Všechna počítači zpracovávaná obrazová data se v základu dělí na:

- vektorová data, popisující obrázek za pomoci křivek. Přináší větší škálovatelnost a menší velikost souborů (formáty např. `.svg`, `.ai`).
- rastrová data, kde obrázek je složen z mřížky (rastru) bodů (pixelů). Přináší větší jednoduchost při zpracování, ale větší velikost souborů a

omezenou škálovatelnost (formáty např. .bmp, .jpg, .png) [Arn12].

V naší aplikaci bude potřeba pracovat s historickými či rekonstrukčními mapami (tj. se scany reálných předloh) a se současnými mapami (mimo jiné ortofotomapami) - v obou kategoriích se tedy pravděpodobně výlučně bude jednat o rastrově reprezentované mapy. Jako takové, zejména pokud mají obsahovat podrobné detaily, ale představují poměrně veliký objem dat, těžko přenositelný po internetu.

Pro ilustraci uvedme, že základní mapy OpenStreetMap [Opeb] mají, v podrobnějším rozlišení, ve kterém jsou k rozpoznání jednotlivé části ulic, průměrně 15KB na 10000m<sup>2</sup> (PNG formát). Pokud bychom tedy chtěli vytvořit takovouto mapu širšího centra Prahy (řekněme 10x10km), měl by výsledný soubor ~160MB. Při průměrné rychlosti download internetového připojení v ČR 15.9Mbps [Fas] by se takováto mapa nahrávala déle než minutu (v ideálním případě).

Z tohoto důvodu se mapy dělí na malé čtvercové obrázky, tzv. mapové dlaždice. *“Cílem je vytvořit vjem souvislé mapy, která se však nenačítá celá najednou, ale po malých mapových dlaždicích pouze v místě, které uživatel právě zobrazuje. Tento způsob výrazně urychluje zobrazování dat, pohyb po mapě i přechod mezi přednastavenými měřítky.”* [av15] Dlaždice jsou navíc rozděleny do jednotlivých výškových vrstev (odpovídajících různým měřítkům, neboli *zoomu*), což opět šetří datový tok.

Výše uvedený příklad mapy Prahy se tedy, namísto z jednoho 160MB velkého souboru v případě již zmíněné základní mapy OpenStreetMaps načítá ze čtverců o velikosti 256x256px (v detailním přiblížení to odpovídá zhruba 80x80m), které používají PNG kompresi a mají každý přibližně 10KB.

V případě této aplikace se však ukázalo, že dostupné historické mapy nemají zdaleka tak vysoké rozlišení jako mapy současné. Jejich originály buď mají příliš malé rozměry nebo je jejich vlastníci záměrně poskytují jen v nižších rozlišeních a tak se velikost komprimovaných scanů map vyjmenovaných v tabulce 2.1 pohybuje pouze v řádech jednotek megabajtů. Díky tomu bude možné historické mapy přenášet bez dělení na jednotlivé dlaždice.

Pro současné podkladové mapy - mnohem detailnější a tedy větší než ty historické - bude ale načítání z mapových dlaždic nezbytné.



## Kapitola 3

### Návrh aplikace

V této kapitole budou formulovány funkční požadavky výsledné aplikace, popsána její struktura, navrženo uživatelské rozhraní a navržen datový model.

#### 3.1 Funkční požadavky

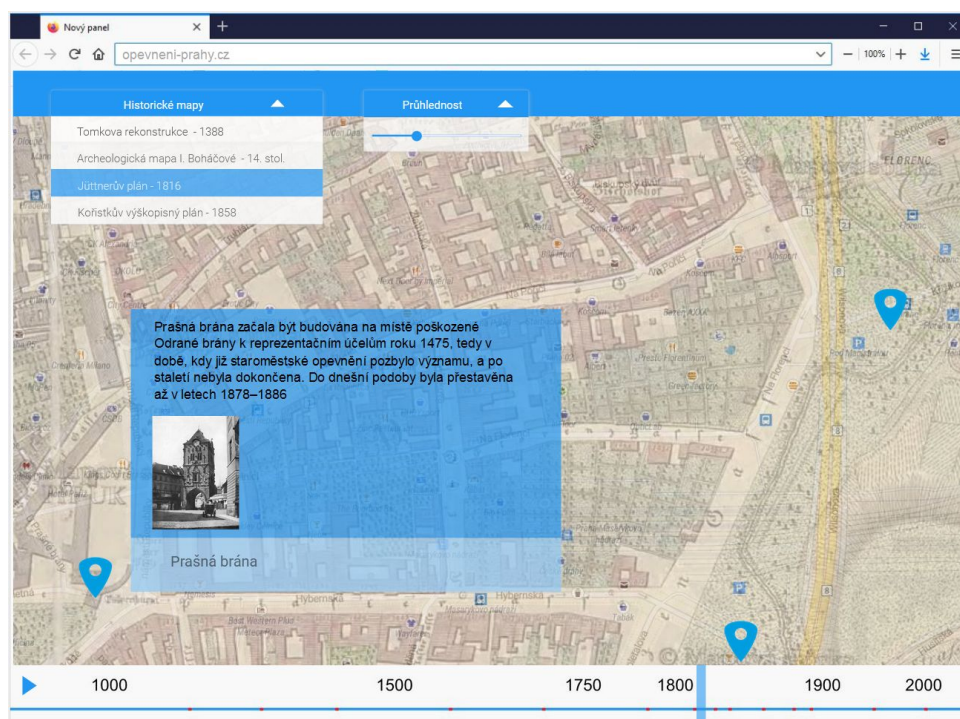
Z analýzy zadání a konzultací s vedoucím práce vyplynuly funkční požadavky na výslednou aplikaci, které jsou shrnuty v Tabulce 3.1. Každý požadavek má určenou prioritu - *Must* nebo *Should*. Pokud by byly splněny jen *Must*-požadavky, nebude splněno celé zadání, ale aplikace bude v základních funkcích použitelná. Pro splnění celého zadání musí být funkční *Must*- i *Should*-požadavky.

#### 3.2 Struktura aplikace

Vzhledem k požadavkům na veřejnou přístupnost aplikace musí mít výsledný projekt webové uživatelské a webové administrátorské rozhraní. Nejjednodušší volbou proto bude celou aplikaci realizovat jako online webovou aplikaci. Vhodné bude dodržet i obvyklou strukturu těchto aplikací a tedy rozdělit ji na:

- **Backend** - část běžící na serveru, starající se o ukládání, spravování, zabezpečení a poskytování dat, k čemuž bude využita **databáze**.
- **Frontend** - část spouštěnou v prohlížeči uživatele (klienta). S využitím **mapové komponenty** bude zajišťovat zobrazování map a významných bodů, dále pak animovat vývoj opevnění a administrátorům nabízet možnosti editace dat.

Jednotlivé technologie - jmenovitě tedy jazyk backendu (případně framework), jazyk frontendu (případně framework) a samotná mapová komponenta - budou vybrány v kapitole 4.



**Obrázek 3.1:** Návrh veřejné části uživatelského prostředí aplikace

### 3.3 Uživatelské rozhraní

Cílem návrhu uživatelského rozhraní (UI) je dosáhnout maximální přehlednosti a logického uspořádání všech prvků aplikace tak, aby její používání bylo co nejvíce intuitivní a pokud možno nevyžadovalo studium manuálu. Na základě zadání, funkčních požadavků, poznatků z rešerší uživatelských rozhraní existujících aplikací (Kapitola 2.1) a průzkumu designu současných webových mapových aplikací byly v grafickém editoru vytvořeny návrhy všech obrazovek UI. Příklady těchto návrhů jsou vyobrazeny na Obrázku 3.1 (příklad veřejné části UI) a na Obrázku 3.2 (příklad administrátorské části UI). Návrhy ostatních obrazovek aplikace jsou shrnuty v příloze této práce. Tyto návrhy představují vizuální formulaci funkcí výsledné aplikace, její finální podoba se však v detailech liší podle softwarové technologie použité pro její realizaci.

### 3.4 Datový model

Protože se nejedná o aplikaci, jejímž hlavním úkolem by byla manipulace s často se měnícími a provázanými daty velkého množství uživatelů, bude databázové schéma poměrně jednoduché. Pro účely této aplikace je potřeba uchovávat následující entity:



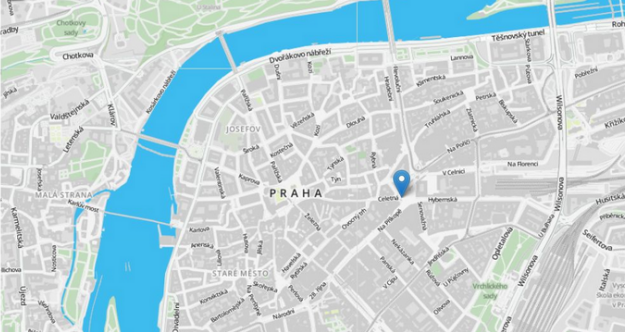
Nový panel x +

opevneni-prahy.cz 100%

Mapy Významné body Administrátorské účty Odhlásit

**Přidat nový významný bod**

Název

Pozice 

Popisek

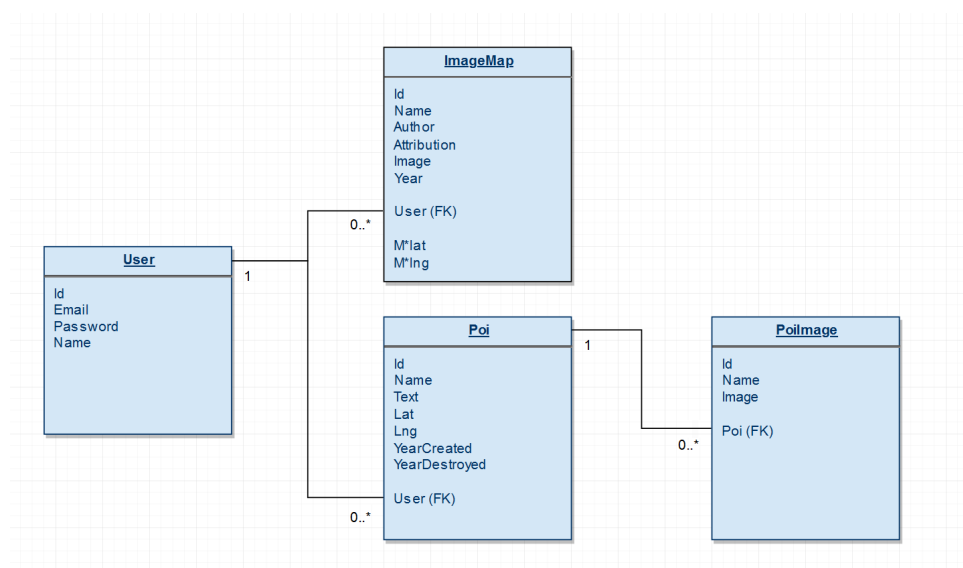
Rok vzniku

Rok zániku   Bod nezaniknul

**Obrázek 3.2:** Návrh administrátorské části uživatelského prostředí aplikace - přidání významného bodu

- **ImageMap** - Překryvné (historické) mapy. Název entity byl zvolen tak, aby nedošlo k záměně s terminologií, kterou používá mapová knihovna (viz kapitola 4.1).
- **Poi** - Významné body (z anglického Point of Interest). Obsahují textové údaje (souřadnice, popisy) k vybraným lokalitám.
- **PoiImage** - Obrazové přílohy k významným bodům.
- **User** - Registrovaní uživatelé (administrátoři). Mají právo editovat obsah.

Relační model databáze s jednotlivými entitami a všemi jejich atributy je zobrazen na obrázku 3.3.



Obrázek 3.3: Relační model databáze aplikace. (FK = cizí klíč)

| Č. požadavku | Popis požadavku   | Priorita |
|--------------|---|----------|
| 1            | Zobrazení map   |          |
| 1.1          | Zobrazit podkladovou (současnou) mapu                   | Must     |
| 1.2          | Zobrazit překryvnou (historickou) mapu                  | Must     |
| 1.3          | Nastavit průhlednost překryvné mapy                     | Should   |
| 2            | Zobrazení významných bodů                               |          |
| 2.1          | Zobrazit Významný bod na mapě                           | Must     |
| 2.2          | Zobrazit detail Významného bodu                         | Must     |
| 2.2.1        | Zobrazit text v detailu Významného bodu                 | Must     |
| 2.2.2        | Zobrazit obrázky v detailu Významného bodu              | Should   |
| 3            | Časová osa  |          |
| 3.1          | Zobrazit významné body z vybraného roku                 | Must     |
| 3.2          | Zobrazit významné body z vybraného intervalu roků       | Should   |
| 3.3          | Přehrát animaci   | Should   |
| 4            | Administrátorské účty                                   |          |
| 4.1          | Přihlásit administrátora                                | Must     |
| 4.2          | Změnit heslo přihlášeného administrátora                | Should   |
| 4.3          | Vytvořit účet nového administrátora                     | Should   |
| 5            | Mapy  |          |
| 5.1          | Nahrát soubor s novou mapu                              | Must     |
| 5.2          | Zapozicovat novou mapu                                  | Must     |
| 5.3          | Dodat údaje k nové mapě                                 | Must     |
| 5.4          | Editovat mapu (změnit pozicování, změnit údaje)         | Should   |
| 5.5          | Smazat mapu   | Must     |
| 6            | Významné body   |          |
| 6.1          | Zapozicovat významný bod                                | Must     |
| 6.2          | Vytvořit textový popis k Významnému bodu                | Must     |
| 6.3          | Vytvořit rok vzniku/zániku u Významného bodu            | Should   |
| 6.4          | Nahrát obrázky k Významnému bodu                        | Should   |
| 6.5          | Editovat Významný bod (změnit pozicování, změnit údaje) | Should   |
| 6.6          | Smazat nahraný obrázek                                  | Should   |
| 6.7          | Smazat Významný bod                                     | Must     |

**Tabulka 3.1:** Funkční požadavky.



## Kapitola 4

### Softwarové prostředky

Tato kapitola se zabývá řešenými, porovnáním a výběrem vhodných technologií k realizaci aplikace.

#### 4.1 Mapová komponenta

Jednou ze stěžejních částí celé aplikace bude frontendová komponenta na zobrazování a překrývání map. Takových komponent je dostupné velké množství, proto zde uvádím základní výčet těch nejpoužívanějších (i nejrepresentativnějších).

##### 4.1.1 Komponenty využívající Adobe Flash

Při řešení existujících aplikací bylo zjištěno, že velké množství z nich je založeno na komponentách využívajících Adobe Flash. Tato technologie ale obecně přestává být masově využívána - kvůli bezpečnostním problémům, výkonnostním problémům a vzrůstající oblibě HTML5, WebGL a dalších. Adobe navíc od roku 2020 přestane vydávat nové verze FlashPlayer a postupně dokonce bude ukončena i podpora celé této platformy v prohlížečích. Za několik let tedy nebude mít běžný uživatel možnost Flash aplikaci vůbec spustit. [JC] Řešení využívající Adobe Flash se proto, díky zastaralosti, jeví jako nevhodné.

##### 4.1.2 Komponenty využívající HTML5 + Java Script

Jako nejaktuálnější se tedy jeví využití komponenty postavené na JavaScriptu/HTML5. Jak ale praví známé přísloví “Každý den vzniknou alespoň dva JavaScriptové frameworky a minimálně jeden zanikne,” které, zdá se, platí ve světě JavaScript komponent minimálně stejnou měrou. Na Obrázku 4.1 je hezké shrnutí vztahů nejvýznamnějších free/opensource webových mapových klientů v roce 2012 (!). Od té doby se situace (dle onoho přísloví) ještě vyvinula. Zde je přehled současných, významně využívaných JS mapových komponent:

- **Google Maps JavaScript API**

Komponenta vyvíjená firmou Google. Slouží zejména pro přístup ke Google Mapám, lze ji ale pro své potřeby částečně upravit. Má však uzavřený



**Obrázek 4.1:** Vztahy free/opensource webových mapových klientů, 2012. Převzato z [geotux.tuxfamily.org](http://geotux.tuxfamily.org) [Car]

zdrojový kód a licenční podmínky vyžadují každé její použití spojit s Google API klíčem. Používá ji např. Interaktivní mapa čs. opevnění.

#### ■ OpenLayers

Rozsáhlá open-source knihovna s pokročilými funkcemi. Umí například zobrazit desítky vektorových mapových formátů, využívat WebGL nebo pracovat s různými kartografickými zobrazeními. Využívá ji například Moravský zemský archiv.

#### ■ LeafletJS

Moderní, flexibilní, kompaktní (kód má pětinouhou velikost oproti OpenLayers) open-source javascript knihovna, která využívá HTML5 a CSS3. Používá ji například Foursquare, Pinterest, Scroll NYC nebo OpenStreetMap.cz (OpenStreetMap.cz navíc poskytují i zdrojové kódy k celé svojí aplikaci v GIT repozitáři [oG].) Dle jejího hesla *"Leaflet doesn't try to do everything for everyone. Instead it focuses on making the basic things work perfectly"* [Lea] postrádá některé pokročilejší funkce (napří-

klad nepodporuje jiná kartografická zobrazení než Mercatorovo). Práce s ní je však jednoduchá, má přizpůsobení pro mobilní zařízení a poskytované funkce jsou dostačující (zejména umožňuje vzájemné překrývání map).

- **CesiumJS**

Zajímavý projekt, zaměřený však zejména na 3D vizualizaci (obdoba GoogleEarth). Vyžaduje prohlížeč s WebGL podporou.

- **ArcGIS**

Velmi pokročilá avšak placená online obdoba populárního GIS systému. Ve Flash verzi ji používá Mapa Kontaminace Cenia, JS verzi pak například IPR Praha.

Seznam dalších komponent na wiki projektu OpenStreetMaps: [https://wiki.openstreetmap.org/wiki/Deploying\\_your\\_own\\_Slippy\\_Map](https://wiki.openstreetmap.org/wiki/Deploying_your_own_Slippy_Map)

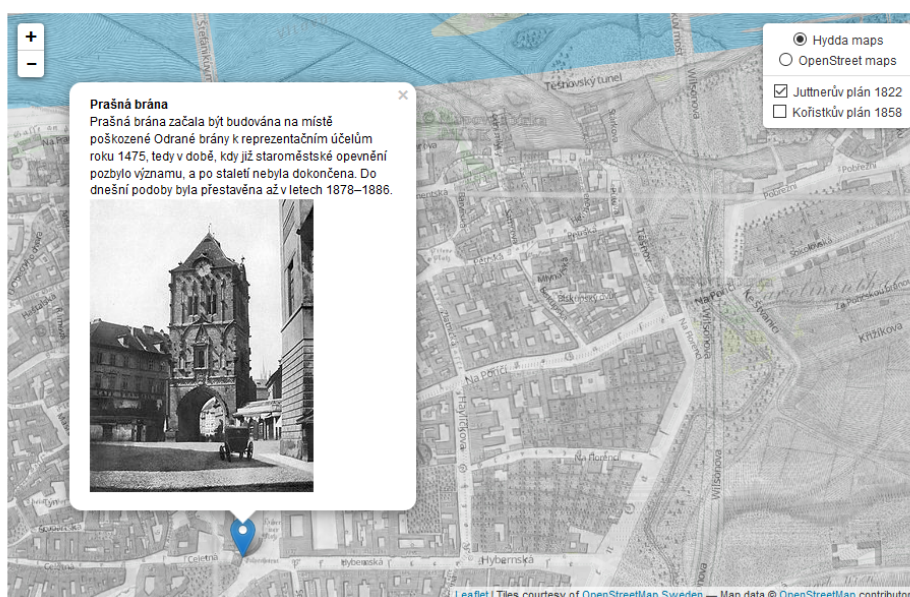
Po porovnání jednotlivých komponent byla, pro vývoj tohoto projektu, vybrána komponenta **LeafletJS**.

## ■ 4.2 Prototyp

V rámci rešerší mapových komponent vznikl jednoduchý prototyp aplikace - zejména pro ověření možnosti použít vybranou komponentu pro požadované účely. Prototyp sestává z jednoduché statické HTML stránky, která obsahuje komponentu LeafletJS. Při otevření stránky je do LeafletJS načtena variace OpenStreetMap Hydda, která je následně překryta 40% průhledným Jüttnerovým plánem (viz Tabulka opatřených map 2.1). Samotný historický plán je načten z jedné 4MB mapové dlaždice. V pravém horním rohu lze přepínat mezi dvěma verzemi OSM map a mezi dvěma historickými mapami. Kromě toho je v mapě zobrazen marker (modrý "špendlík") - Prašná brána. Po kliknutí na něj se zobrazí okno s informacemi a fotografií Prašné brány. Prototyp je zobrazen na obrázku 4.2. Navíc je dostupný online, na <http://matlafous.xf.cz/TRY/bakala/try.html>.

## ■ 4.3 Technologie frontendu

Kvůli jednoduchosti frontendu – který nevyžaduje například asynchronní volání nebo jiné komplikovanější operace a kromě LeafletJS ani větší komponenty – bylo rozhodnuto nepoužít žádný rozsáhlý frontendový framework (jakým je například ReactJS či Angular). Kromě klasického **HTML**, **JavaScriptu** a **CSS** bude však využito knihovny **jQuery** a pro zjednodušení vytváření UI sady stylovacích CSS a JS nástrojů **Bootstrap**.



V pravém horním rohu lze zapínat/vypínat dvě historické mapy a dvě současné mapy Prahy.  
Po kliknutí na modrý "špendlík" se otevře popup okénko s informacemi o Prašné bráně.

Toto je pokusný prototyp pro Semestrální projekt - Fortifikace Prahy. (FEL ČVUT, Kryštof Novák, 2019)

Obrázek 4.2: Prototyp aplikace s využitím LeafletJS.

## 4.4 Technologie backendu

S enormním rozvojem internetu došlo v poslední době i k rozvoji možností, jak internetové aplikace vytvářet. Téměř každý programovací jazyk lze dnes nějakým způsobem použít pro vývoj backendu a tak by samotná rešerše možných technologií vydala na celou bakalářskou práci. Proto jsem svou volbu založil na vlastních zkušenostech a doporučení pedagogů, a z nejvýznamnějších technologií, kterými dnes jsou Spring (Java), PHP, Node.js (JavaScript), Asp.net, Ruby on Rails, Perl případně Python (Django), jsem se rozhodl pro stále ještě nejtradičnější přístup - vývoj v **PHP**.

### 4.4.1 PHP frameworky

Ačkoliv by bylo možné tuto aplikaci napsat v čistém PHP bez externích knihoven, rozhodl jsem se pro využití frameworku. To by například mělo vést k větší bezpečnosti, zrychlení programování rutinních úloh nebo k dodržení standardní kostry aplikace. Vzhledem k vysoké poptávce po programátorech se znalostmi PHP frameworků nelze také zanedbat přínos v osvojení si některé z těchto technologií.

PHP frameworků existuje velké množství a rozhodnout, který je pro můj projekt nejlepší - bez jejich detailní znalosti - se jeví jako téměř nemožné. Své rozhodnutí jsem tedy založil na základě doporučení pedagogů a na statistikách



používanosti jednotlivých frameworků [Goo] (statistiky Google). Výběr jsem tak zúžil na tři frameworky:

#### ■ **Laravel**

Dnes celosvětově nejpoužívanější PHP framework [Goo]. Je poměrně nový (první vydání 2011), open source a šířen pod MIT licencí. Laravel částečně vychází z frameworku Symfony, staví na MVC architektuře a využívá zavedené podpůrné systémy jako Composer, Eloquent ORM a Blade. Má velmi širokou komunitu uživatelů, existuje k němu velké množství návodů. Byly v něm vytvořeny například weby 9GAG, eshop AboutYou nebo část webu BBC. [mwL]

#### ■ **Nette**

Nejpoužívanější framework v ČR [Goo]. Používá jej například ČSFD, ROOT.CZ, Uložto, nebo Slevomat (tedy opravdu zejména tuzemské projekty). V České republice má silnou komunitu, v celosvětovém měřítku je ale téměř neznámý. Přináší vlastní debugovací nástroj Tracy, šablonovací systém Latte a dle slov vývojářů *"Filosofie Nette klade mimořádný důraz na produktivitu, nejlepší postupy a bezpečnost."* [Net]

#### ■ **SlimPHP**

Jedná se o takzvaný microframework - obsahuje jen naprosto základní funkce, což ho ale dělá velmi malým a rychlým. Je tedy vhodný pro malé aplikace, pro které by používání plnohodnotných frameworků znamenalo spíše zpomalení - vývoje i provozu. Slim je jedním z nejpoužívanějších PHP microframeworků a proto často figuruje i v žebříčcích nejpoužívanějších PHP frameworků (včetně těch plnohodnotných).

Jelikož backend bude vyžadovat spíše specifický kód pro tuto aplikaci, než například pokročilejší session management, možnost kombinování různých návrhových vzorů nebo třeba manipulaci s emaily, zůstala by velká část funkcí rozsáhlejších frameworků nevyužita. Mohl by tak v tomto případě dobře posloužit i Slim framework - jehož nasazení bude rychlé a ponechá volbu struktury aplikace přímo na programátorovi. Vzhledem k častým pracovním požadavkům na znalost sofistikovanějších frameworků jsem se ale nakonec rozhodl Slim nevyužít a díky světovému rozšíření (které se pravděpodobně dá očekávat i v ČR) byl nakonec, pro tento projekt, vybrán **Laravel framework**.

## ■ 4.5 Technologie databáze a ukládání dat

Jak bylo ukázáno v kapitole 3.4, databázový model této aplikace je poměrně jednoduchý a zároveň se nepředpokládají časté změny dat (minimálně ne takové, které by bylo nutné ihned reflektovat na straně uživatelů). Přesto, že by bylo možné databázové schéma aplikace (Obrázek 3.3) převést i do nerelačního modelu, výhod nerelačních databází - tedy hlavně rychlého dodávání často se měnících dat (jaké nabízí třeba Firebase Realtime DB) - by v této

aplikaci nemohlo být využito a použití takovéto databáze tedy, pro tento případ, postrádá smysl. Bylo proto zvoleno použití relační **SQL** databáze.

Díky přednostem zvoleného PHP frameworku Laravel bude aplikace navíc v podstatě nezávislá na konkrétní SQL implementaci a tedy schopná pracovat s MySQL, PostgreSQL, SQLite nebo SQL Server 2017+ databázemi bez výraznějších změn [Lara].

Soubory map budou ukládány do adresářové struktury serveru (tedy ne do databáze). Za pomoci .htaccess konfigurací pak bude možné snadno nastavit cache přístup k těmto datům. Pro ukládání map bude použit PNG formát, popřípadě JPEG (v závislosti na formátu, ve kterém mapu nahraje přímo uživatel).

V průběhu pokusného nasazení aplikace musela být navíc do aplikace přidána možnost ukládat data i do úložiště **Amazon S3** (více o nastavení této možnosti v kapitole 6.3.2. Konfigurace ).

## Kapitola 5

### Implementace

Aplikace byla implementována dle návrhu z kapitoly 3. Návrh a s využitím technologií vybraných v kapitole 4. Softwarové prostředky. Tato kapitola se zabývá podpůrnými nástroji, které výrazně usnadnily vývoj, a dále uvádí dodatečně použité technologie. Uzavírá ji část věnovaná struktuře aplikace. Zdrojové kódy jsou k dispozici v GIT repozitáři [Nov] nebo na příloženém CD.

#### 5.1 Vývojové nástroje

Vývoj probíhal v IDE **PhpStorm** z balíku společnosti JetBrains. Toto vývojové prostředí se, zejména po doinstalování Laravel pluginu, velmi osvědčilo a po předchozích zkušenostech se zdá být lepší volbou, než původně zamýšlené IDE NetBeans.

Část nástrojů s sebou přinesl přímo framework **Laravel 6.0** a to zejména: **Artisan CLI** - nástroj zjednodušující (mimo jiné) provádění často se opakujících rutinních úloh, dále **Lokální webový server**, který je rozšířením zabudovaného PHP serveru a **Tinker** - nástroj umožňující přímý přístup k běžícímu programu.

Dále byly využity klasické nástroje pro správu knihoven - **Composer**, starající se o backendové knihovny a **Npm** (Node Package Manager), zajišťující frontendové balíčky.

Pro verzování zdrojových kódů byl použit systém **GIT**, repozitář je dostupný online [Nov].

#### 5.2 Technologie spojené s Laravelem

Přestože hlavním programovacím jazykem aplikace je PHP, bylo by správnější prohlásit, že aplikace byla naprogramována v Laravelu. Tento framework samozřejmě programátorovi nechává možnost volby, pokud se ale drží jeho zvyklostí, přichází Laravel nejen s výchozí strukturou projektu, ale i s doporučením na množství předem určených dalších technologií. K výčtu vybraných technologií z předchozí kapitoly je tedy ještě potřeba přidat:

- Šablonovací systém **Blade**, který do tradičních HTML/JS Views přináší v podstatě vlastní programovací jazyk, velmi usnadňující například zobrazování dat, strukturování Views nebo třeba zabezpečení. Blade šablony jsou automaticky kompilovány do PHP.

- **SASS** – Syntactically Awesome Style Sheets, starající se o stylování Views. Ty do tradičního CSS zavádí např. proměnné, funkce nebo zjednodušující selektory. Při sestavení aplikace jsou kompilovány do čistého CSS [SAS].

- Množství frameworkových knihoven na straně backendu, starajících se například o autentizaci/autorizaci uživatelů, úložiště nebo odstínění programátora od komunikace s databází.

### 5.3 Rozšiřující knihovny

Kromě stěžejních technologií, vybraných v kapitole 4. Softwarové prostředky a technologií spojených s Laravel frameworkem, byly do projektu postupně přidány ještě následující knihovny třetích stran:

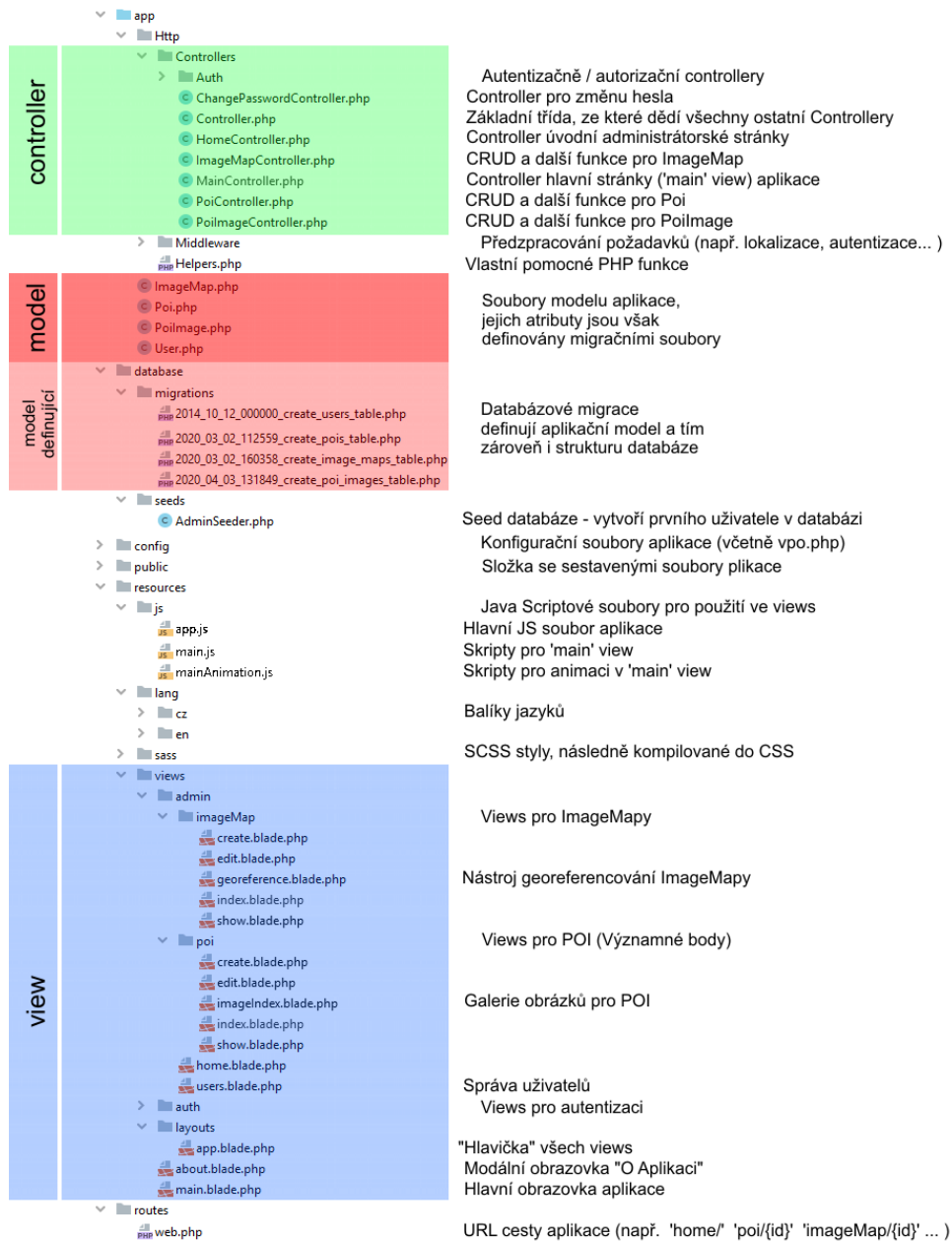
- Frontendové, spravované npm:
  - Leaflet.ImageOverlay.Rotated** [San] Rozšíření komponenty Leaflet o možnost otáčet a deformovat překryvné vrstvy.
  - Seiyria/bootstrap-slider** [Kem] Posuvníková komponenta, umožňující výběr úseku a zvýraznění částí dat. Kromě těchto výhod je navíc, na rozdíl od výchozího bootstrap/custom-slider, kompatibilní s MS Edge.
  - DataTables** [Dat] jQuery plugin umožňující snadnější zacházení s tabulkami: vyhledávání v nich, řazení nebo stránkování.
- Backendové, spravované Composerem:
  - flysystem-aws-s3-v3** Knihovny pro komunikaci s Amazon S3 úložištěm.

### 5.4 Struktura projektu

Aplikace se drží architektury Model-View-Controller (MVC), preferované Laravel frameworkem. Tato struktura se odráží i v adresářové hierarchii projektu, kde jsou:

- Objekty **Modelu** umístěny ve složce `/Http/`  
(Jejich atributy definovány za pomoci Schema Builders v databázových migracích [Larb] `/database/migrations/`)
- Třídy **Controllerů** v `/Http/Controllers/`
- A **Views** ve složce `/resources/views/`

Struktura (vybraných) důležitých souborů projektu, včetně stručného popisu jejich významu, je znázorněna na Obrázku 5.1.



Obrázek 5.1: Struktura důležitých souborů projektu.



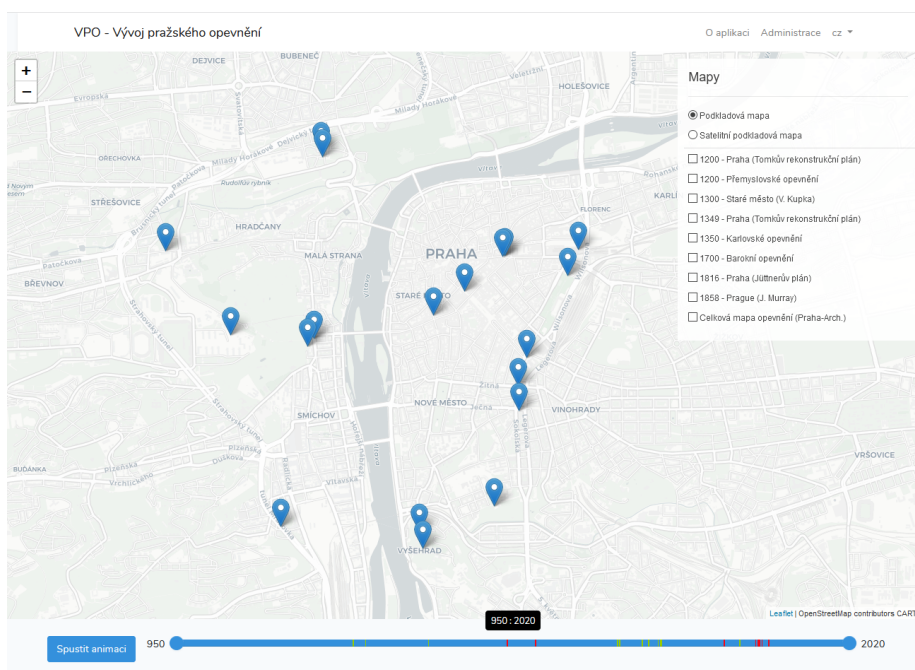
## Kapitola 6

### Výsledná aplikace

Výsledná aplikace, vzniklá v rámci práce popsané v předchozí kapitole, byla pojmenována **VPO – Vývoj pražského opevnění**. Jedná se o funkční a prakticky použitelnou aplikaci, která byla navíc pokusně veřejně nasazena. V první části této kapitoly jsou uvedeny ukázky fungování VPO, druhá část se věnuje veřejně nasazené verzi. Třetí podkapitola je věnována postupu pro lokální spuštění aplikace a její konfiguraci.

#### 6.1 Ukázky a popis výsledné aplikace

**Veřejná část** Hlavní obrazovka výsledné aplikace VPO - Vývoj pražského opevnění je ukázána na obrázku 6.1.



**Obrázek 6.1:** Úvodní obrazovka VPO se základní podkladovou mapou a vybranou celou časovou osou.

Většinu plochy zabírá komponenta *Leaflet* zobrazující mapu s významnými body. Po stranách se nachází ovládací prvky:

- V pravém horním rohu menu *Mapy* - v jeho horní části je přepínač dvou podkladových map, v jeho dolní části pak seznam historických překryvných map. Při vybraní historické mapy se navíc objeví posuvník *průhlednosti*.

- V dolní části obrazovky je umístěna časová osa - lze na ní vybrat úsek historie, který má být zobrazen - na mapě poté zůstanou jen body, které ve vybraných letech existovaly. Tlačítkem *Spustit animaci* se začne úsek na časové ose animovaně posouvat.

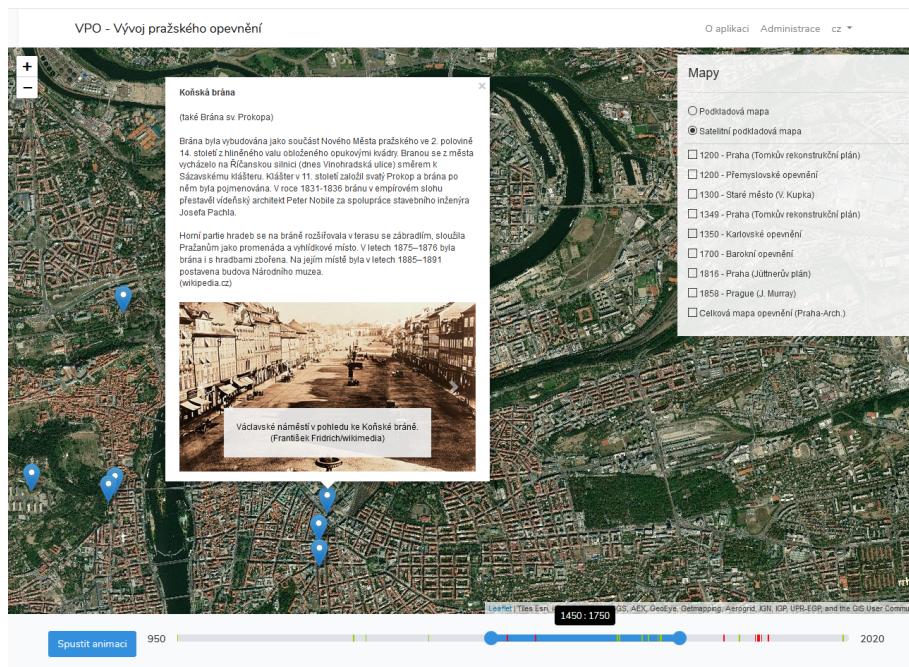
- V horní části okna se nachází menu, umožňující: změnit jazyk aplikace, vstoupit do administrátorské části a otevřít informační obrazovku *O Aplikaci*.

Příklady různých stavů hlavní obrazovky jsou uvedeny na obrázku 6.2 (detail významného bodu na satelitní mapě s vybraným časovým úsekem) a na obrázku 6.3 (překryvná mapa s nastavenou průhledností).

**Administrátorská část** aplikace je rozdělena na:

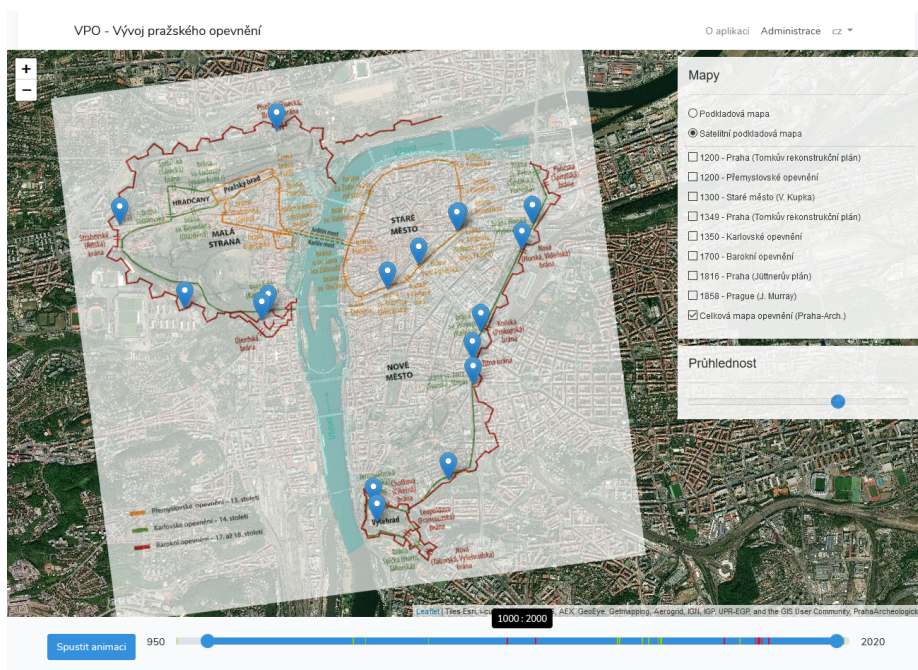
- (Významné) Body - seznam, detail, vytvoření, editace, galerie obrázků
- Mapy - seznam, detail, vytvoření, editace, georeferencování
- Uživatelé - detail přihlášeného, změna hesla, vytvoření

Ukázky prostředí administrátorské části jsou uvedeny na obrázku 6.4 - seznam významných bodů a na 6.5 - georeferencování vkládané mapy. Screenshoty dalších obrazovek aplikace jsou k nalezení v příloze této práce.

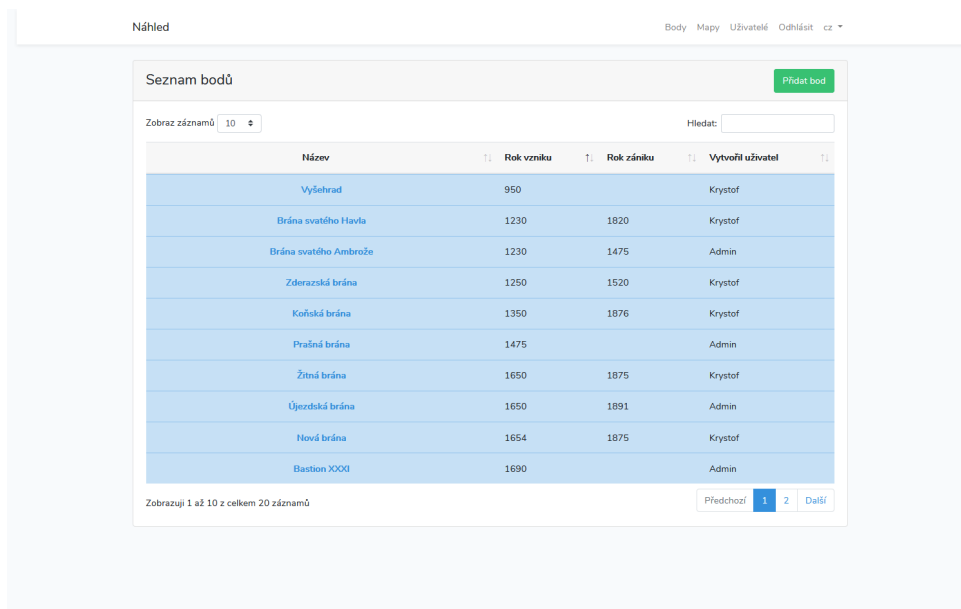


**Obrázek 6.2:** Úvodní obrazovka VPO se satelitní podkladovou mapou a vybraným úsekem časové osy (1450-1750). Otevřen detail významného bodu "Koňská brána."

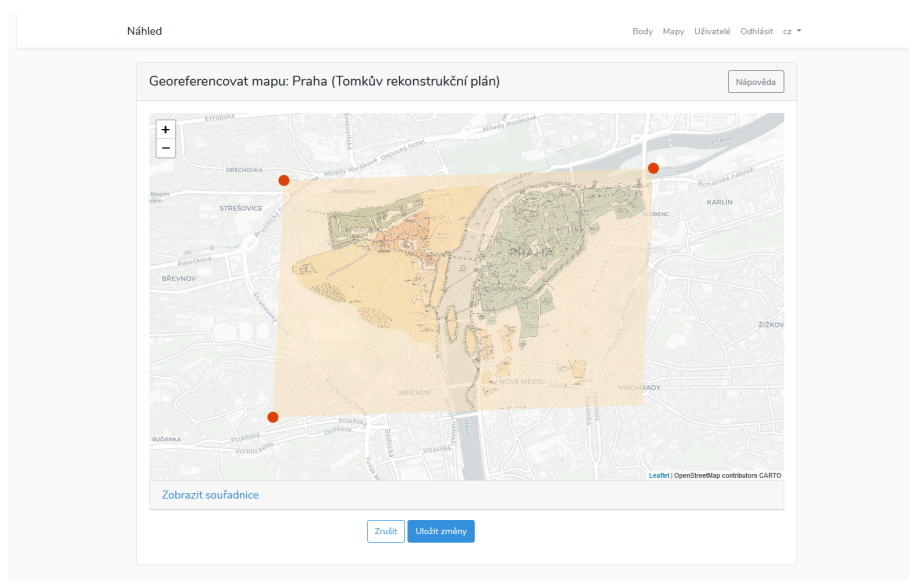




**Obrázek 6.3:** Úvodní obrazovka VPO se satelitní podkladovou mapou a překryvnou rekonstrukční mapou.



**Obrázek 6.4:** Administrátorská sekce VPO - seznam významných bodů.



**Obrázek 6.5:** Administrátorská sekce VPO - georeferencování vkládané mapy.

## 6.2 Veřejně spuštěná aplikace

Aby mohla být aplikace, zejména během vývoje, prezentována vzdáleně a bez nutnosti případného zájemce instalovat na svém počítači celý server, byla aplikace pokusně nasazena (s jistými omezeními, viz dále) na veřejně přístupném bezplatném serveru, konkrétně na:

<http://vpo.herokuapp.com>

Pro přístup do administrátorské části použijte:

Email: *test@test.com*

Heslo: *novak2020bp*

Pro demonstrační účely technického fungování aplikace bylo do veřejně nasazené verze vloženo několik významných bodů i map (hrubě pozicovaných). Nejedná se o kompletní množinu, která by demonstrovala úplnou historii vývoje pražských opevnění; více dat bude do aplikace dodáno v budoucnu (viz kapitola 8. Závěr)

### 6.2.1 Technická specifika zvoleného hostingu

Pro veřejné hostování aplikace byla zvolena služba Heroku, a to díky předinstalovaným knihovnám pro Laravel a možnosti přímého propojení s GitHubem, což velmi usnadnilo vydávání nových verzí.

Jeho bezplatný “Hobby” program ale bohužel přináší i jistá omezení:

- Aplikace se vždy po půl hodině neaktivity sama “hibernuje.” Návštěva stránek po uplynutí této doby tedy znamená opětovný start celé aplikace, který může trvat i 10 vteřin.

- Pro tento projekt původně zamýšlená databáze MySQL je k dispozici pouze v placeném programu. V Hobby programu je však dostupná Postgres databáze, čímž se opět projevila výhoda použití Laravel frameworku. V ENV konfiguraci stačilo jen změnit DB\_CONNECTION z “mysql” na “pgsql” a aplikace bez problému nyní funguje s jiným typem databáze.
- Šifrované HTTPS je dostupné opět jen v placeném programu.

Heroku hosting navíc nenabízí žádné datové úložiště, v našem případě potřebné pro ukládání ImageMap.image a PoiImage.image. Bylo proto využito Amazon S3 úložiště, které však (v neplacené verzi) přináší některé další nevýhody:

- Pomalé načítání dat (které překvapivě téměř nezávisí na velikosti načítaného souboru)
- Omezení velikosti souboru na 2MB

## 6.3 Vlastní spuštění aplikace

Pro plnou funkcionalitu aplikace, nezatíženou problémy konkrétního hostingu, jako je tomu v případě veřejně nasazené verze, je nutné spustit si aplikaci na vlastním serveru. Podkapitola 6.3.1. Lokální instalace obsahuje návod, dle kterého je možné si VPO spustit pro vlastní lokální (testovací) použití. Druhá podkapitola - 6.3.2. Konfigurace - popisuje konfigurační soubory, kterými lze nastavit přístup aplikace k dalším službám, upravit její chování nebo ji dokonce i přizpůsobit pro zobrazování historie jiného města.

### 6.3.1 Lokální instalace

#### Prerekvizity.

Pro lokální spuštění aplikace je potřebné mít nainstalované:

- PHP (může být i součástí nějakého balíku jako např. XAMPP)
- Composer

Dále jsou potřeba již jen zdrojové kódy VPO, které je možné stáhnout z GIT repozitáře [Nov], případně je zkopírovat z příloženého CD.

#### Instalace.

Ve složce s aplikací VPO spusťte příkaz:

```
composer install
```

čímž dojde ke stažení všech potřebných knihoven. Následně přejmenujte soubor ‘env.example’ na ‘env’. V tomto souboru jsou zapsány základní konfigurace pro aplikaci VPO (více v kapitole 6.3.2).

#### Příprava databáze.

Pokud hodláte používat databázi SQLite, není potřeba v ‘env’ souboru provádět žádné změny a stačí jen ve složce ‘database/’ vytvořit prázdný soubor ‘database.sqlite’. Posledním krokem je příprava databázové struktury, která se, díky migračním souborům, dá vytvořit pouhým zavoláním příkazu:



# Kapitola 7

## Testování

Nejen po dokončení aplikace ale i během jejího vytváření (při použití TDD dokonce i před ním) je nedílnou součástí každého projektu testování. V případě VPO byly použity dva typy testování: automatické (programovací) testy, které pomáhaly během vývoje udržet celou aplikaci funkční a praktické (ruční) testování, díky kterému byly odhaleny specifické chyby a byla vylepšena celková použitelnost výsledné aplikace.

### 7.1 Automatické testy

Ve výchozí konfiguraci přichází Laravel s integrací PHPUnit frameworku, postaveném na xUnit architektuře [PHP]. I přes svůj název Laravel využívá PHPUnit nejen pro

- Unit testy - *jednotkové* testy zaměřené jen na malé úseky kódu, většinou jen jednu metodu  
ale zejména pro tzv.
- Feature testy - testující (většinou) rozsáhlejší část aplikace, která může obsahovat například několik spolu interagujících objektů nebo i vytvářet kompletní HTTP požadavky.

Velikou předností testů typu Feature je možnost využívat v nich rozšiřujících funkcí Laravelu. Všechny testy v tomto projektu jsou z toho důvodu typu Feature.

Testy vytvořené pro VPO jsou vyjmenovány v příloze této práce, tabulka C.1. Tyto testy byly používány pravidelně během vývoje a několikrát odhalily problémy na první pohled zcela nesouvisející.

Testy lze spustit zavoláním příkazu  
`vendor/bin/phpunit`

### 7.2 Manuální testy

Ačkoliv technickou funkčnost aplikace hlídají automatické testy, její praktickou použitelnost je stále nutné kontrolovat a zkoušet manuálně. V následující

| Název              | Verze               | Problémy                                     |
|--------------------|---------------------|--|
| Mozilla Firefox    | 72.0.2              | -  |
| Chromium           | 82.0.4081.0         | -  |
| Opera              | 67.0.3575.137       | -  |
| MS Edge / EdgeHTML | 44.18362 / 18.18363 | custom-slider se při každém kliknutí rozšíří |

**Tabulka 7.1:** Internetové prohlížeče, ve kterých byla VPO testována.

podkapitole proto byla VPO testována v různých prohlížečích a prostředích.

### 7.2.1 Různé internetové prohlížeče

VPO byla otestována v internetových prohlížečích vyjmenovaných v tabulce 7.1. Ve všech testovaných prohlížečích, s výjimkou Microsoft Edge (s jádrem EdgeHtml), fungovala aplikace bez problému a až na drobné rozdíly ve vykreslování i vypadá stejně.

Posuvníková komponenta bootstrap/custom-range, která se v Microsoft Edge chová velmi podivně, byla nahrazena komponentou seyria/bootstrap-slider. Aplikace byla po této změně znovu otestována ve všech zmíněných prohlížečích. Nesprávné zobrazování posuvníku v Microsoft Edge tím bylo vyřešeno a nevyskytly se žádné další problémy.

### 7.2.2 Mobilní zařízení

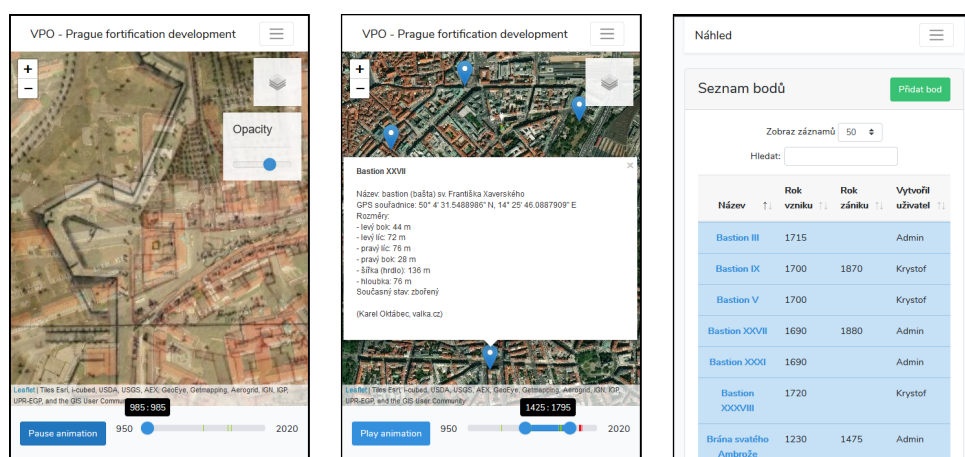
Při testování VPO na mobilních telefonech vyšlo najevo, že i přes použití komponenty LeafletJS, která by měla být pro mobilní zařízení přizpůsobena, mohou vyvstat problémy při používání hlavní obrazovky aplikace.

Přizpůsobení tohoto typu aplikací - tedy webových mapových aplikací zobrazujících nejen čistou mapu - se obecně ukázalo být problematické na tak malých displejích, jaké mají mobilní telefony.

Dokládají to i aplikace popsané v kapitole 2.1. Zhruba jen polovina z nich je nějakým způsobem přizpůsobena pro použití na mobilních zařízeních a i jejich používání je velmi nepohodlné. Zbytek je na mobilních zařízeních zcela nepoužitelný. Aplikace DvěPrahy od IPR Praha dokonce působí dojmem, že je spíše uzpůsobená pro použití na 15 metrů široké projekční ploše IPR CAMP Praha (kde se skutečně také používá), než na displejích mobilních zařízení.

Aby však byla VPO základně použitelná i na mobilní zařízení, bylo provedeno testování na různých velikých emulovaných i reálných mobilních obrazovkách, které přineslo tyto úpravy:

- Automatické skrývání nabídky Map.
- Zavedena maximální šířka *Leaflet.Tooltip* na 90vw.
- Povolení automatického skrývání horního menu do “Hamburger” ikony.



**Obrázek 7.1:** VPO na displeji mobilního telefonu.

- Minimální možná šířka *opacitySlideru*.

Všechny tyto změny musely být provedeny jen kvůli komplikované a “ne-tradiční” hlavní obrazovce aplikace. Administrátorská část – s tradičnějším rozložením na stránky, formuláře, tabulky a jednotlivé komponenty – je díky Bootstrapu responzivní automaticky.

Snímky výsledků přizpůsobení mobilním zařízením jsou zobrazeny na obrázku 7.1.

## 7.3 Uživatelské testování

Uživatelské prostředí VPO bylo, základním způsobem, otestováno několika různými uživateli. Dle požadavků (zobrazte překryvnou mapu, snižte průhlednost historické mapy, přihlaste se, přidejte významný bod apod.) byla aplikace postupně procházena a byly zaznamenány tyto problémy a nejasnosti, s následujícím řešením:

- *Není jasné, jak v dialogu "Přidání Bodu" přesunout přidávaný bod.*  
Přidán informační tooltip "Kliknutím do mapy posunete bod."
- *Není jasné, jak se z administrátorské části vrátit do veřejné části.*  
Titulek aplikace, kterým se lze vrátit do veřejné části, byl - pro administrátory - změněn na nadpis "Náhled."
- *Při prvním používání aplikace nemusí být jasné, k čemu slouží které ovládací prvky.*  
Přidána úvodní informační obrazovka, která se zobrazí při prvním otevření aplikace.
- *Není-li zaškrtnutá žádná mapa, přítomnost posuvníku průhlednosti mate uživatele.*

Zavedeno automatické skrývání posuvníku průhlednosti, pokud není vybrána žádná mapa.

- *Pokud přidám do galerie obrázků obrázek bez popisku, aplikace mne nijak nevaruje.*

Aplikace nyní varuje, pokud přidáváte obrázek bez popisku.

- *Na podkladové mapě není Karlův most (resp. téměř není vidět).*  
Původní podkladová mapa - OSM/SE-Hydda musela být, kvůli dlouhodobému výpadku serverů v dubnu 2020, nahrazena jinou mapou. Jedinou vhodnou nalezenou náhradou byla mapa OSM-Carto, která však má několik takovýchto nedostatků. Jakmile budou servery OSM/SE opět v provozu, bude podkladová mapa zpět nahrazena mapou Hydda.

Kromě těchto změn bylo do každého administrátorského formuláře přidáno tlačítko "Nápověda", které zobrazí informační text ke konkrétnímu formuláři.



## Kapitola 8

### Závěr

V rámci této práce byly provedeny rešerše, připraven návrh a následně implementována, otestována a pokusně nasazena Interaktivní webová aplikace pro vizualizaci vývoje pražského opevnění - VPO.

Nejdříve byly provedeny rešerše již existujících aplikací - na jejich základě jsem dospěl k rozhodnutí vizualizovat vývoj opevnění za pomoci kombinace překrývání částečně průsvitných historických map se současnými mapami a zobrazování tzv. významných bodů v těchto mapách. V další části rešerší jsem se zabýval dostupností vhodných historických podkladů. Původní záměr – demonstrovat vývoj pražského opevnění na historických mapách – se ukázal jako v podstatě nerealizovatelný. K poslední výstavbě opevnění totiž došlo ve 20. letech 18. století a nejstarší přesné mapy Prahy pochází až ze začátku 19. století. V naší aplikaci jsme se tedy přidrželi alternativního řešení, popsaného v podkapitole Historické mapy, totiž překlenout období nezachycené na přesných historických mapách nově vytvořenými rekonstrukčními mapami a historické mapy využít jen k vizualizaci období devatenáctého století. Byly opatřeny celkem 4 historické mapy a 7 rekonstrukčních map. Pro překrývání se současnou mapou byl zvolen mapový zdroj OpenStreetMap/SE a byly nalezeny dodatečné zdroje historických informací k pražskému opevnění. Dále byla s ohledem na naši aplikaci zkoumána problematika mapových zobrazení. Bylo zjištěno, že na území o velikosti většího města (v našem případě Prahy) jsou rozdíly mezi jednotlivými kartografickými zobrazeními zanedbatelné, zejména v případě nepříliš přesných historických map. Byl také zvolen optimální způsob, jakým má aplikace umožňovat pozicování historických map, jaké mapové formáty mají být podporovány a kdy je vhodné využítí tzv. mapových dlaždic.

Na základě výsledků všech těchto analýz byl vypracován návrh aplikace: funkční požadavky, databázový model, technická struktura a uživatelské rozhraní.

Následně byly provedeny rešerše a výběr technologií pro realizaci aplikace – zejména frontendových mapových komponent, z nichž byla následně vybrána komponenta LeafletJS. Po průzkumech používanosti a konzultaci s pedagogy byl dále zvolen backendový jazyk PHP, framework Laravel a frontendový balík Bootstrap.

Aplikace byla úspěšně implementována a poté pokusně veřejně nasazena na

webové adrese <http://vpo.herokuapp.com>. Zdrojové kódy byly zveřejněny v GitLab repozitáři. Aplikace byla podrobena testování, a to jak průběžnému (automatizovanému), tak i ručnímu, obnášejícímu testování v různých prostředích i různými uživateli. Na základě výsledků testování bylo opraveno několik technických chyb a aplikace byla upravena pro snadnější používání.

Výslednou aplikaci by samozřejmě bylo možné dále zdokonalovat - například rozšířit možnosti nástroje pro pozicování map, zajistit lepší hosting, nebo přidat více úrovní uživatelů. Stěžejní pro úspěch aplikace však nyní bude zejména dotvořit její obsah a naplnit ji relevantními informacemi. V době dokončování tohoto textu probíhá jednání se serverem valka.cz o možném částečném převzetí záštity nad aplikací. Mohlo by tak dojít nejen ke vstupu historiků do procesu plnění aplikace informacemi a do otázek jejího dalšího rozvoje, ale také k propagaci VPO mezi laickými zájemci o historii. Domnívám se, že aplikace VPO tedy nejen splňuje požadované zadání bakalářské práce, ale má i reálný potenciál být užitečná a zajímavá pro širokou veřejnost.



## Literatura

- [Arn12] Amy E. Arntson, *Graphic design basics*, Wadsworth, 2012.
- [av15] Jáchym Kellar ZČU / Fakulta aplikovaných věd, *Integrace dat pro účely cykloturistické aplikace*, [https://otik.uk.zcu.cz/bitstream/11025/17987/1/kellar\\_bp.pdf](https://otik.uk.zcu.cz/bitstream/11025/17987/1/kellar_bp.pdf), 2015.
- [Bal] Caterina Balletti, *Analytical and quantitative methods for the analysis of the geometrical content of historical cartography*, [https://www.isprs.org/proceedings/xxxiii/congress/part5/30\\_XXXIII-part5.pdf](https://www.isprs.org/proceedings/xxxiii/congress/part5/30_XXXIII-part5.pdf).
- [Bri84] Encyclopædia Britannica, *Maps and mapping - map projections*, 15 ed., Encyclopædia Britannica Inc., 1984.
- [Car] German Carrillo, *Web mapping client comparison v.6*, <http://geotux.tuxfamily.org/index.php/en/geo-blogs/item/291-comparacion-clientes-web-v6>.
- [Com] Open Data Commons, *Odc open database license*, <https://opendatacommons.org/licenses/odbl/summary/>.
- [Dat] DataTables, *Datatables*, <https://datatables.net/>.
- [Fas] FastMetrics, *Internet speeds by country*, <https://www.fastmetrics.com/internet-connection-speed-by-country.php>.
- [Goo] GoogleTrends, *Trends - laravel vs. symfony vs. slim php vs. nette*, <https://trends.google.com/trends/explore?q=laravel,Symfony,Slim%20PHP,Nette>.
- [HA17] Jeremy Black Heritage Andrew, *Mapy měst*, Universum, 2017.
- [JC] Zive.cz Jakub Cizek, *Legendární flash player končí. z webu definitivně zmizí v roce 2020*, <https://www.zive.cz/clanky/a-je-to-tady-legendarni-flash-player-konci-z-webu-definitivne-zmizi-v-roce-2020-sc-3-a-188749/default.aspx>.
- [Kem] Kyle J. Kemp, *Github seiyria/bootstrap-slider*, <https://github.com/seiyria/bootstrap-slider>.

- [Kup08] Vladimír Kupka, *Pražská opevnění*, Libri, 2008.
- [kvO] Vědecká knihovna v Olomouci, *Mapy staré prahy k letům 1200, 1348 a 1419*, <http://www.digitalniknihovna.cz/vkol/view/uuid:d1524bea-2ff8-4116-9c26-740392046f00?page=uuid:8aa1f070-f1e9-495e-ac18-f30affc8f2c1>.
- [Lara] Laravel, *Database - documentation 6x*, <https://laravel.com/docs/6.x/database>.
- [Larb] ———, *Migrations - documentation 6x*, <https://laravel.com/docs/6.x/migrations>.
- [Lea] Docs Leaflet.com, *Leafletjs*, <https://leafletjs.com/>.
- [mwL] made with Laravel, *Madewithlaravel.com*, <https://madewithlaravel.com/>.
- [Net] Nette, *Nette docs*, <https://nette.org/cs/>.
- [Nov] Kryštof Novák, *Gitlab repozitář - vpo*, <https://gitlab.fel.cvut.cz/novakkr3/vpo>.
- [oG] osmap.cz GitHub, *osmcz-app*, <https://github.com/osmcz/osmcz>.
- [Opea] Docs OpenLayers, *Spherical mercator*, [http://docs.openlayers.org/library/spherical\\_mercator.html](http://docs.openlayers.org/library/spherical_mercator.html).
- [Opeb] OpenStreetMaps, *Openstreetmaps - tiles*, <https://www.openstreetmap.org>.
- [Pa] Ivana Boháčová Praha archeologická, *Hradby středověké prahy*, <http://www.praha-archeologicka.cz/p/367?tgo=82>.
- [PHP] PHPUnit, *Phpunit - documentation*, <https://phpunit.de/>.
- [PK] QuickAdminPanel Povilas Korop, *How to use laravel .env and .env.example files?*, <https://quickadminpanel.com/blog/how-to-use-laravel-env-example-files/>.
- [San] Ivan Sanchez, *Github leaflet imageoverlay rotated*, <https://github.com/IvanSanchez/Leaflet.ImageOverlay.Rotated>.
- [SAS] SASS, *Sass: Documentation*, <https://sass-lang.com/documentation>.
- [UK] Jan Royt Univerzita Karlova, *Nové město pražské*, <https://karel700.cuni.cz/KAREL-57.html>.
- [VHA] VHA, *Sbírka plánů opevnění praha - 1680-1954*, [http://www.vuapraha.cz/sites/default/files/inventare/1\\_tzv\\_rakouske\\_fondy\\_do\\_roku\\_1918/plany\\_praha\\_-\\_1680-1954.pdf](http://www.vuapraha.cz/sites/default/files/inventare/1_tzv_rakouske_fondy_do_roku_1918/plany_praha_-_1680-1954.pdf).

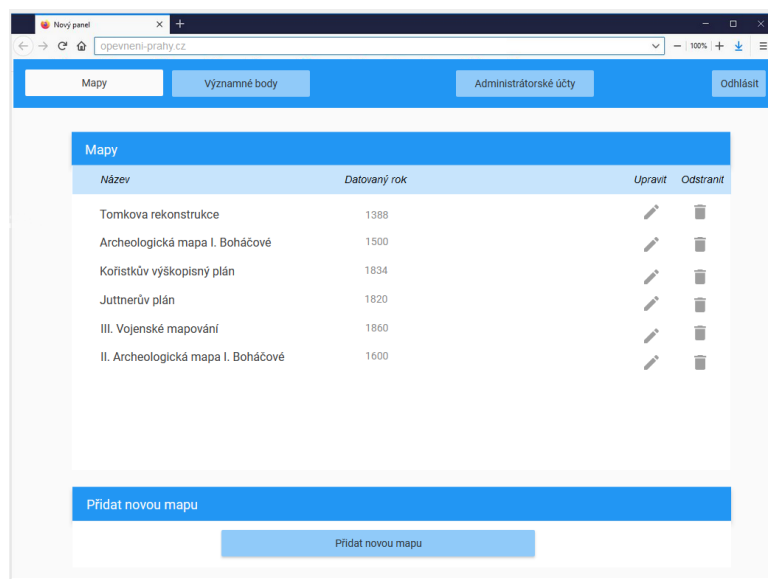
- [VÚ] v.v.i. VÚGTK, *Virtualní mapová sbírka*, <http://chartae-antiquae.cz/cs/mapsets/0/>.
- [Vá] Karel Oktábec Válka.cz, *Pevnost praha*, <https://www.valka.cz/Pevnost-Praha-t120329>.
- [Wika] Wikimedia, *Plán prahy vytvořený josefem jüttnerem*, [https://commons.wikimedia.org/wiki/File:Juttneruv\\_plan\\_Prahy.jpg](https://commons.wikimedia.org/wiki/File:Juttneruv_plan_Prahy.jpg).
- [Wikb] Wikipedia, *Comparison of web map services*, [https://en.wikipedia.org/wiki/Comparison\\_of\\_web\\_map\\_services](https://en.wikipedia.org/wiki/Comparison_of_web_map_services).
- [Wike] ———, *Mapové zobrazení*, [https://cs.wikipedia.org/wiki/Mapov%C3%A9\\_zobrazen%C3%AD](https://cs.wikipedia.org/wiki/Mapov%C3%A9_zobrazen%C3%AD).



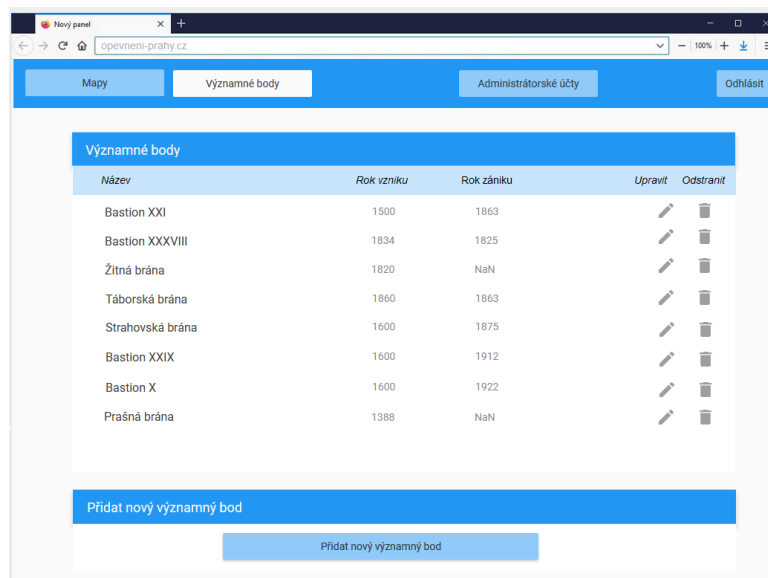


## **Příloha A**

### **Návrhy uživatelského rozhraní**

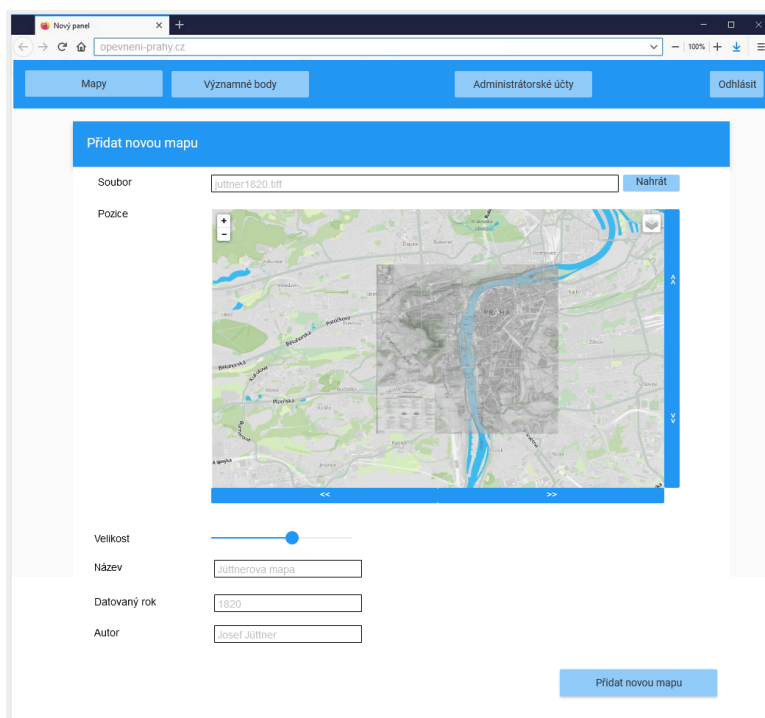


**Obrázek A.1:** Návrh administrátorské části uživatelského prostředí aplikace - seznam map.

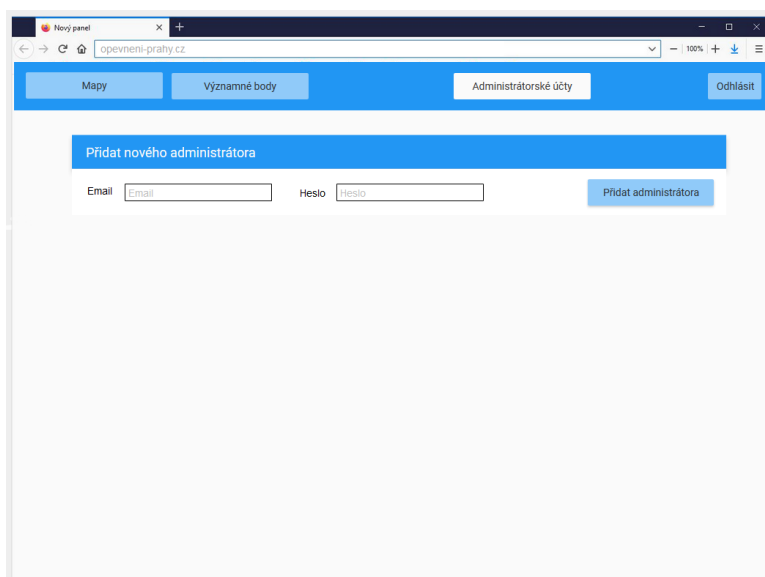


**Obrázek A.2:** Návrh administrátorské části uživatelského prostředí aplikace - seznam významných bodů.





**Obrázek A.3:** Návrh administrátorské části uživatelského prostředí aplikace - přidání nové mapy.



**Obrázek A.4:** Návrh administrátorské části uživatelského prostředí aplikace - seznam administrátorů.





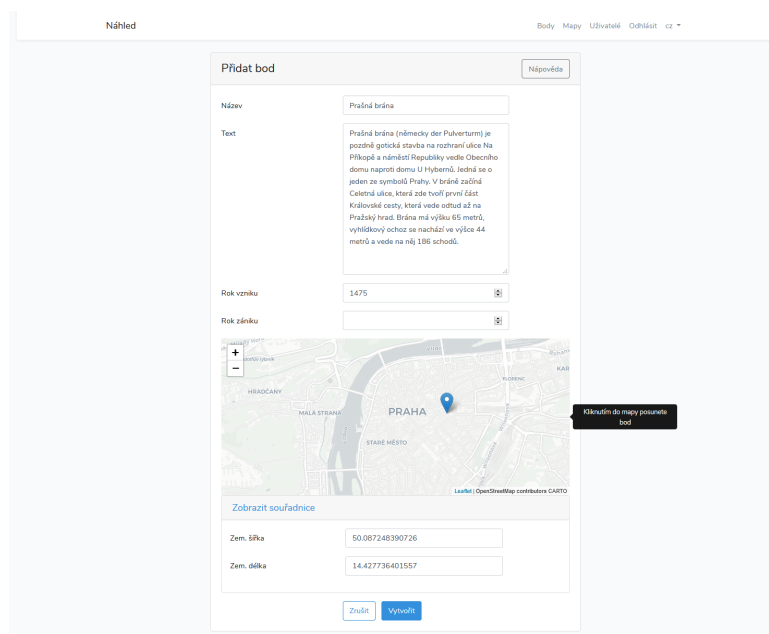
## **Příloha B**

### **Ukázky výsledné aplikace**

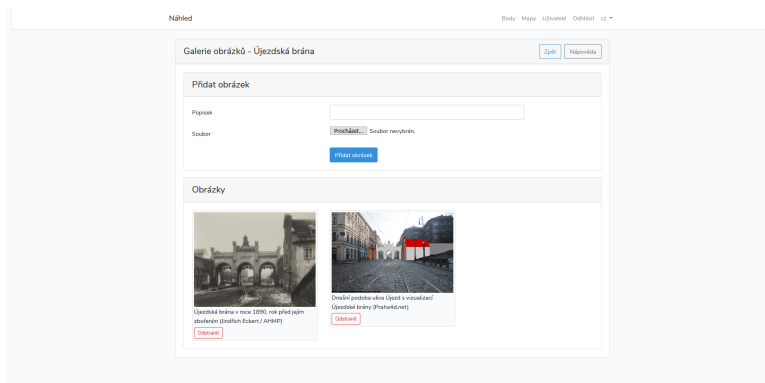
## B. Ukázky výsledné aplikace



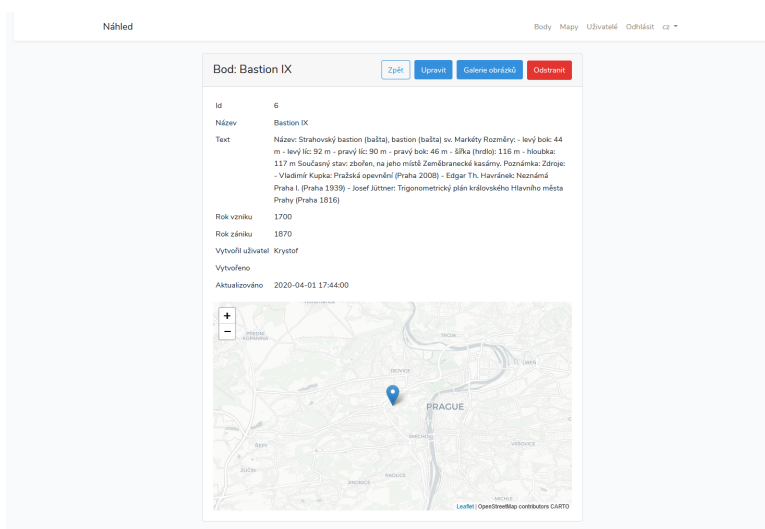
**Obrázek B.1:** Veřejná část uživatelského prostředí aplikace - detail významného bodu, překryvná mapa - Jüttnerův plán 1816.



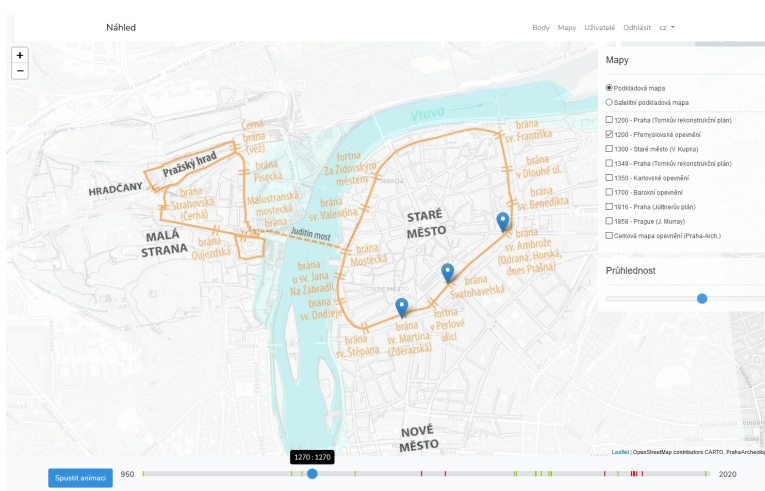
**Obrázek B.2:** Administrátorská část uživatelského prostředí aplikace - přidání významného bodu.



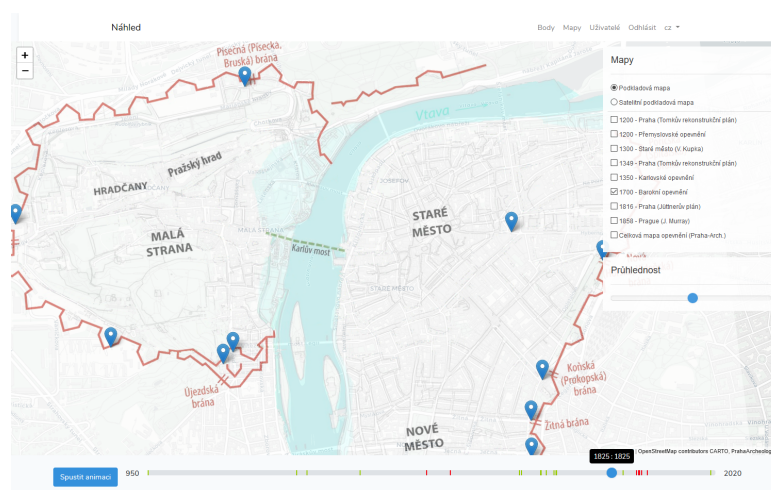
**Obrázek B.3:** Administrátorská část uživatelského prostředí aplikace - galerie obrázků k významnému bodu.



**Obrázek B.4:** Administrátorská část uživatelského prostředí aplikace - detail významného bodu.



**Obrázek B.5:** Veřejná část uživatelského prostředí aplikace - spuštěná animace - rok 1270, překryvná mapa - Přemyslovské opevnění.



**Obrázek B.6:** Veřejná část uživatelského prostředí aplikace - spuštěná animace - rok 1825, překryvná mapa - Barokní opevnění.



## **Příloha C**

### **Automatické testy**

| Název testu   |
|---|
| testBasicAccess   |
| testBackendHomeAccess_authorized                        |
| testBackendHomeAccess_unauthorized                      |
| testImageMapCreate_Valid_authorized                     |
| testImageMapCreate_Valid_unauthorized                   |
| testImageMapCreate_MissingFile_authorized               |
| testImageMapCreate_MultipleValid_NamesNotUnique         |
| testImageMapCreate_DefaultMlat                          |
| testImageMapUpdate_Valid_authorized                     |
| testImageMapUpdate_Valid_unauthorized                   |
| testImageMapDelete_Valid_authorized                     |
| testImageMapDelete_Valid_unauthorized                   |
| testImageMap_HasUser                                    |
| testImageMapCreate_ImageNotImage_authorized             |
| testGetImageMapImage_NotExisting_unauthorized           |
| testGetImageMapImage_NotExisting_authorized             |
| testPoiImageCreate_Valid_authorized                     |
| testPoiImageCreate_Valid_unauthorized                   |
| testPoiImage_HasPoi                                     |
| testGetPoiImage_NotExisting_NotExistingPoi_unauthorized |
| testGetPoiImage_NotExisting_ExistingPoi_unauthorized    |
| testGetPoiImage_NotExisting_NotExistingPoi_authorized   |
| testGetPoiImage_NotExisting_ExistingPoi_authorized      |
| specificPoiImageData                                    |
| testPoiCreate_Valid_unauthorized                        |
| testPoiCreate_MissingLatLng_authorized                  |
| testPoiCreate_MultipleValid_NamesNotUnique              |
| testPoiCreate_Default0Category                          |
| testPoiUpdate_Valid_authorized                          |
| testPoiUpdate_Valid_unauthorized                        |
| testPoiDelete_Valid_authorized                          |
| testPoiDelete_Valid_unauthorized                        |
| testPoi_HasUser   |
| testUser_HasPoi   |
| testUser_HasImageMap                                    |

**Tabulka C.1:** PHPUnit testy vytvořené pro VPO.