|  |  |  |
| --- | --- | --- |
|  | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Marauder’s Map** | | |
| Antonín Dulava | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2017/2018 | |

***Poděkování***

*Tímto bych chtěl poděkovat všem, kteří mě podporovali a motivovali při práci na projektu. Dále bych chtěl poděkovat svým spolužákům za případné postřehy, co zlepšit, zvláště bych chtěl poděkovat Lukášovi Moravcovi, který mi pomohl s projektem po serverové části a doporučil mi vhodnější technologie pro mou práci, v neposlední řadě bych chtěl poděkovat panu učiteli Mgr. Marku Lučnému za nápady, co dále přidat a vylepšit, paní učitelce Ing. Haně Rohanové a panu učiteli Ing. Petru Grusmannovi.*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2017

*podpis autora práce*

**ANOTACE**

Cílem projektu bylo vytvořit webovou aplikaci, která by umožňovala sledování pozic dalších uživatelů nebo zařízení, označení zajímavých míst, označit navštívená místa a možnost ke každému takovému místu přidat hodnocení. Název aplikace je z filmu Harry Potter, kde je speciální mapa pro sledování jiných lidí (Marauder’s Map – Pobertův plánek).

Webová aplikace byla vytvořena ve frameworku Nette a s použitím jazyka PHP, ovšem využívá i další technologie jako Javascript, AJAX a další. Uživatel si při první návštěvě webové stránky může založit nový účet. Údaje o uživatelích se nachází v databázovém systému MySQL, takže jsou z dosahu nějakého nebezpečí. Přístup k datům o uživateli (údaje, souřadnice, kde se nachází, apod.) má jen administrátor.

Uživatel se může přihlásit do aplikace, kde vidí mapu s body, které značí jednotlivé pozice každého uživatele (včetně svojí pozice). Pomocí dalších funkcí může zaznačit oblíbené místo na mapě, které se ukáže všem ostatním uživatelům. Oblíbené místo obsahuje hodnocení uživatele, který tento bod přidal.

**Klíčová slova**

Webová aplikace, mapa, Marauder’s Map

**OBSAH**

[Úvod 7](#_Toc502414873)

[1 Teoretická a metodická východiska 8](#_Toc502414874)

[1.1 Mapa 8](#_Toc502414875)

[1.2 Webová aplikace 9](#_Toc502414876)

[1.3 Databáze 10](#_Toc502414877)

[1.4 Komunikace 10](#_Toc502414878)

[1.5 Uživatel 11](#_Toc502414879)

[1.6 Administrátor 12](#_Toc502414880)

[1.7 Oblíbené místa 13](#_Toc502414881)

[2 Využité technologie 14](#_Toc502414882)

[2.1 Nette Framework 14](#_Toc502414883)

[2.2 PHP 14](#_Toc502414884)

[2.3 Javascript 14](#_Toc502414885)

[2.4 Open Street Map 14](#_Toc502414886)

[2.5 Google Maps API 15](#_Toc502414887)

[2.6 MySQL 15](#_Toc502414888)

[2.7 AJAX 15](#_Toc502414889)

[2.8 Bootstrap 15](#_Toc502414890)

[2.9 Použité programy 15](#_Toc502414891)

[2.9.1 Sublime text 3 15](#_Toc502414892)

[2.9.2 NetBeans IDE 8.2 16](#_Toc502414893)

[2.9.3 Xampp 16](#_Toc502414894)

[2.9.4 MySQL Workbench 16](#_Toc502414895)

[3 řešená Problematika 17](#_Toc502414896)

[3.1 Přesnost lokalizace 17](#_Toc502414897)

[3.2 Připojení dalších uživatelů 17](#_Toc502414898)

[4 Výsledek 18](#_Toc502414899)

[4.1 Registrace 18](#_Toc502414900)

[4.2 Přihlášení 18](#_Toc502414901)

[4.3 Administrační funkce 19](#_Toc502414902)

[4.4 Práce s mapou 19](#_Toc502414903)

[4.4.1 Dva uživatelé 19](#_Toc502414904)

[4.4.2 Oblíbené místo 20](#_Toc502414905)

[4.4.3 Náhled 20](#_Toc502414906)

[Závěr 21](#_Toc502414907)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 22](#_Toc502414908)

Úvod

První myšlenkou bylo vytvořit aplikaci, která by zjednodušila dnešní život. Inspirací pro mě byla aplikace Facebooku sdílení polohy. Tento nápad se mi zalíbil a došlo mi, že by taková aplikace mohla přinést do života přehled. Pro příklad si představme, že jste podnikatel a chcete vědět, kde se váš dopravce se zbožím právě nachází. Můžete proto lépe plánovat do budoucna, protože víte, za jak dlouho se k vám zboží dostane. Můžete vidět, jestli stíhá dopravce zboží dopravit včas nebo se někde zpozdil. Dalším příkladem je, že jste klient, který očekává poštovního kurýra s balíkem. Možnost sledování, kde se právě kurýr s vaším balíkem nachází, vám dává možnost jej odchytit někde v terénu, lépe naplánovat čas, abyste stihli dorazit domů, než přijede kurýr a v neposlední řadě taky vám dává představu, v kolik hodin přibližně dorazí kurýr k vám.

Takový byl prvotní nápad a cíl mého projektu. Bohužel vzhledem k několika vzniklým problémům jsem musel projekt soustředit také i jiným směrem, proto jsem se rozhodl přidat do webové aplikace další funkce např. označení navštíveného místa s možností hodnocení (jako ve škole), které se zobrazí všem uživatelům.

V dalších kapitolách se budu věnovat postupům práce, proč a jaké technologie jsem zvolil a nahlédnu do problematiky a kódu tohoto projektu.

# Teoretická a metodická východiska

Na začátku mé práce jsem se rozhodl rozdělit projekt do několika dílu. Začal jsem základem, který byl samotná mapa.

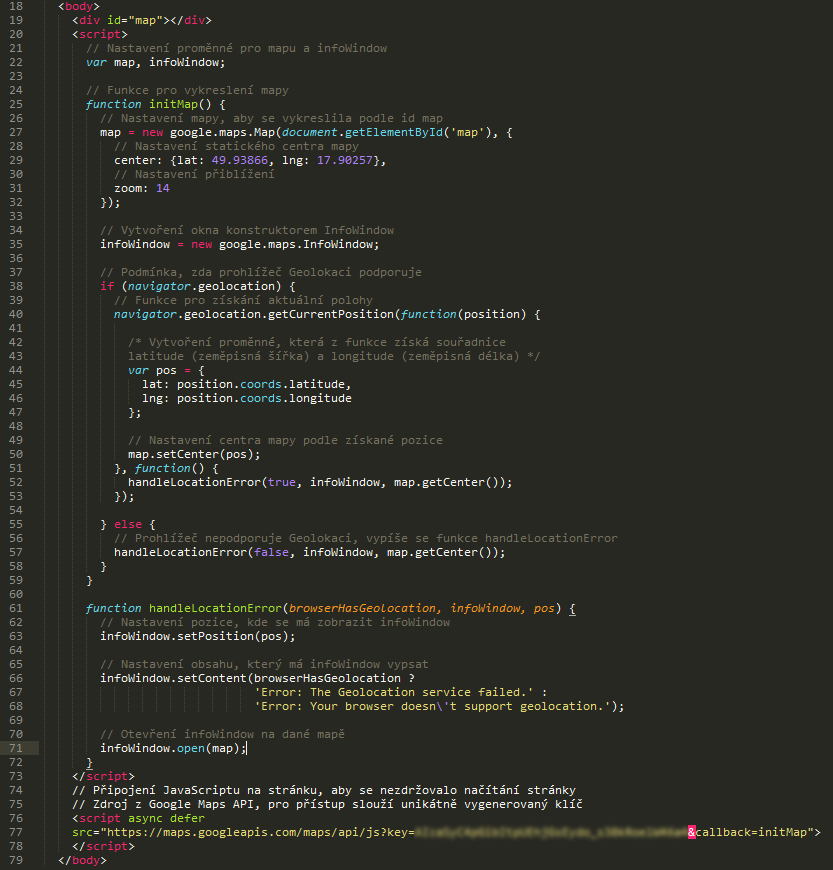
## Mapa

Když jsem poprvé začal řešit mapu, tak jsem se dočetl o Open Street Map. Z oficiálních stránek jsem získal soubory k mapám a začal jsem pracovat s mapou. Zprovoznil jsem vykreslení mapy v HTML stránce. Dalším krokem byla práce s daty a mapou. Konkrétně získání polohy, vycentrování mapy a nastavení dalších vlastností. Tady jsem se setkal s prvním problémem, protože mapy vykreslovaly data pomaleji, než jsem očekával, nenabízely žádnou změnu vzhledu a funkce pro určení polohy pomocí prohlížeče fungovala s velkou odchylkou.

Více jsem si zjišťoval o Open Street Map a na diskusi na jednom fóru jsem se dočetl o Google Maps API. Zjistil jsem si více informací o Google Maps API a spolupráci s Googlem, až jsem narazil na celý vývojářský web, který se soustředí jen na práci s mapami ve webových stránkách. To mě přimělo ještě více Google Maps API vyzkoušet, a proto jsem si zažádal o kód pro přístup právě k Google Maps API. Během několika minut jsem se zaregistroval a získal klíč. Výhodu Google Maps API jsem zaregistroval během prvních momentů a rozhodl jsem pro svůj projekt použít právě mapy od Googlu.

Open Street Map pro vykreslení mapy potřeboval spoustu externích souborů, takže složka s těmito soubory obsahovala 92 MB, na druhé straně Google Maps API bere zdroj mapy ze stránek Google, do kterých se přistupuje pomocí klíče, který mi byl přidělen. Další výhodou je přehlednost a lepší práce s kódem. Ačkoliv se stále jedná o Javascript, tak rozumím tomuto kódu více než u předchozí verze od Open Street Map. Google Maps API umožňují již předdefinovaný výběr rozhraní mapy mezi základním mapou a satelitní mapou. Problém s funkcí na získání polohy pomocí prohlížeče je sice redukovaný, ale stále obsahuje odchylku.

Neukladat screenshot kodu ale kod a barevne tagy



*Obrázek č. 1 Ukázka kódu - mapa*

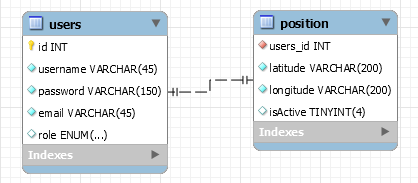
## Webová aplikace

Dalším mým úkolem, bylo postarat se o webovou aplikaci. Vybral jsem si framework od Nette, jelikož jsem již kdysi ve škole pracoval s Nette a umím se v něm trochu orientovat.

Na začátek jsem přizpůsobil kostru Nette mé představě a připojil jsem mapu. Pomocí softwaru Xampp jsem zprovoznil webovou aplikaci na adrese lokálního počítače.

## Databáze

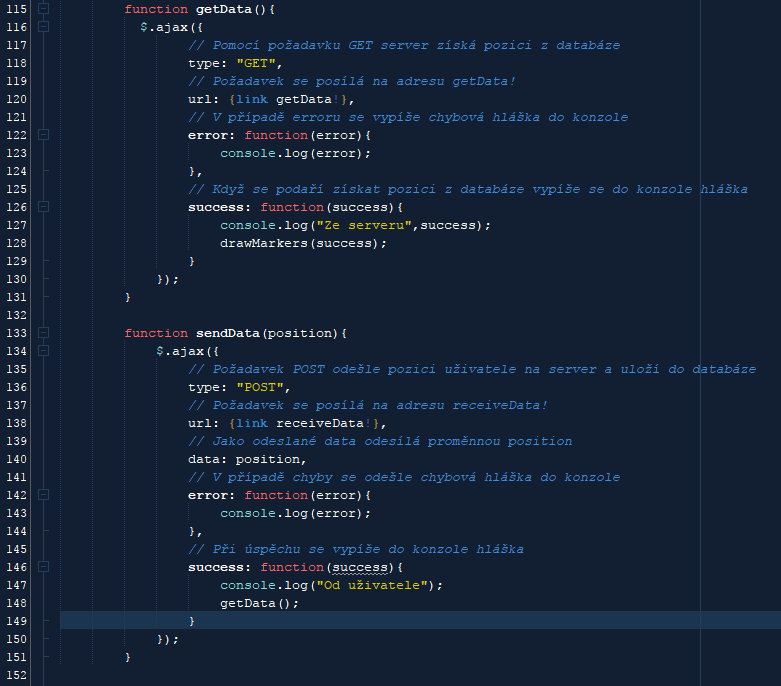
Dále jsem se zabýval, aby se pozice získaná pomocí funkce v Javascriptu ukládala do databáze. Znovu jsem využil znalostí ze školy, a proto jsem vytvořil databázi v programu MySQL Workbench. Databáze je tvořena ze dvou tabulek, tabulky uživatele a tabulky pozice. V tabulce uživatele jsou údaje o uživateli (id, jméno, zašifrované heslo, email) a v tabulce pozice se nachází cizí klíč, který je odkazem na id od uživatele, aby byla uložena pozice každého uživatele. Další pole v tabulce je zeměpisná délka, zeměpisná šířka a nakonec pomocné pole pro kontrolu, zda je uživatel přihlášený nebo ne. Nechceme přeci vykreslovat na mapě někoho, kdo je do aplikace jen zaregistrovaný, nikoliv však přihlášený.



*Obrázek č. 2 ER Diagram*

## Komunikace

Pomocí AJAXu (plugin v Nette) můžeme vytvořit komunikaci mezi klientem a serverem. Pomocí požadavků GET se pošle požadavek na server, který vrátí pozici z databáze, následně pomocí druhého požadavku POST se pošle pozice aktuálního uživatele na server a uloží se do databáze. Server čerpá novou pozici z databáze a vykresluje ji jako bod na mapě.



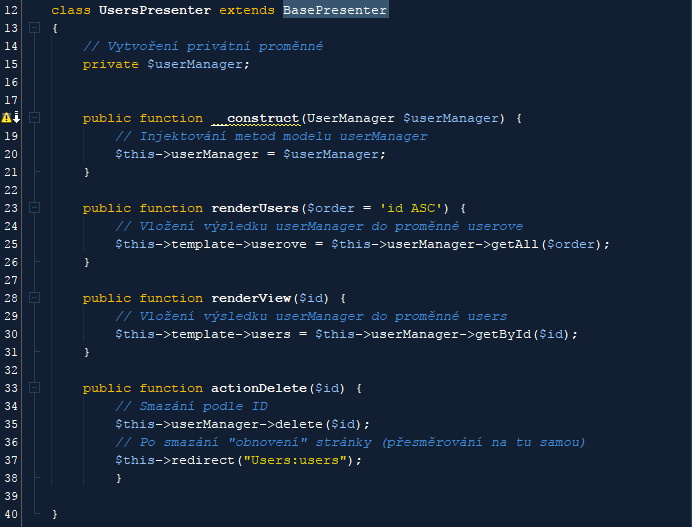
*Obrázek 3. Ukázka kódu - Komunikace*

## Uživatel

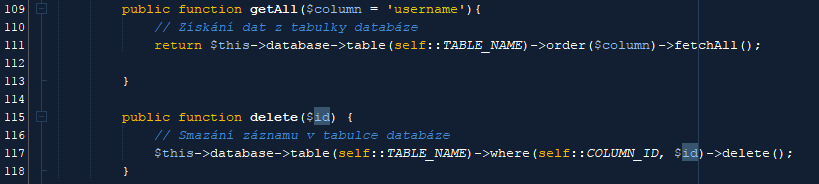
Do webové aplikace jsem přidal možnosti registrace, přihlášení a odhlášení. Napadlo mě, že by se uživatel mohl přihlásit buď pomocí přihlašovacího jména, nebo pomocí emailu. Bohužel přihlašování pomocí emailu zlobilo. Při získávání informací ohledně problému jsem se opět dočetl o nové věci, která se jmenovala Kdyby/Facebook. Jedná se o vytvořenou dokumentaci od pana Filipa Procházky, o tom jak v Nette vytvořit přihlašování pomocí účtu na sociální stránce Facebook. Podle této dokumentace jsem se snažil zprovoznit v mé aplikaci přihlášení pomocí Facebooku.

## Administrátor

Přidal jsem funkci pro administrátora, abych mohl hlídat počet registrovaných uživatelů. Kdyby nějakého záškodníka napadlo vytvořit hodně zbytečných účtů, tak ať je možnost nějak tyto účty smazat přímo přes webovou aplikaci. V budoucnu bude mít funkce administrátora větší použití.



*Obrázek č. 4 Ukázka kódu – Administrátor - Presenter*



*Obrázek č. 5 Ukázka kódu – Administrátor - Manager*

## Oblíbené místa

Funkce oblíbeného místa funguje obdobně jako u uživatele. Oblíbené místo je značeno jiných barevným bodem, než jsou uživatele a obsahuje událost, při které je možno zobrazit hodnocení onoho místa. Do budoucna bych chtěl přidat jméno uživatele, který daný bod přidal a samozřejmě také funkci pro administrátora pro smazání přebytečných bodů.

# Využité technologie

## Nette Framework

Jedná se o český framework, který je open source, jeho využití je tvorba webových aplikací pomocí jazyku PHP verze 5 a 7. Výhoda je, že podporuje AJAX a MVC (model-view-controller). Využívá událostmi řízené programování a z velké části je založen na použití komponent. Přednostmi Nette jsou, že obsahuje silný validační jazyk, automaticky generuje validační JavaScript a má silnou ochranu proti SQL injection. K nevýhodám patří občasné mazání cache paměti a slabou dokumentaci funkcí frameworku. Nette Framework jsem si vybral jako základ pro mou webovou aplikaci, protože jsem se s tímto frameworkem již setkal a pracoval jsem v něm ve škole.

## PHP

Je skriptovací programovací jazyk určený především pro programování internetových stránek a webových aplikací ve formátu HTML, XHTML nebo WML. Použití PHP pro stránky jsou skripty prováděny na straně serveru. Kvůli frameworku Nette jsem psal kód v PHP. Jednalo se hlavně o nastavení serverové části.

## Javascript

Multiplatformní objektově orientovaný skriptovací jazyk. Používá se jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu. Byl potřeba pro úpravu vlastností mapy a pro manipulaci s mapou.

## Open Street Map

Je to projekt, který má za cíl tvořit geografické data a starat se o jejich vizualizaci. Jako podklad slouží záznamy z přijímačů GPS nebo jiných digitalizovaných map. Open Street Map má koncepci Open source, proto je vhodný pro využití ve webové stránce nebo aplikaci. Open Street Map mi sloužilo jako základ v první verzi.

## Google Maps API

Internetová mapová aplikace poskytována zdarma (pro nekomerční použití) od společnosti Google. Slouží jako zdroj pro Map Google, Google Transit a mapy vložené na internetové stránky. S mapami lze pracovat pomocí Javascriptu. Google Maps API je základní kámen mého projektu.

## MySQL

Databázový systém, který slouží pro tvorbu multiplatformních databází. Komunikace probíhá pomoci jazyka SQL. Uplatňuje také relační databázový model. Potřebnou databázi pro ukládání dat jsem vytvořil v jazyce MySQL a pracuju s ní pomocí dotazů MySQL

## AJAX

Jedná se o způsob, kterým můžeme vyměňovat data v aplikaci s databází, serverovými skripty (PHP, Java, apod.) aniž bychom museli aktualizovat celou stránku. Technologii AJAX jsem využil při komunikaci se serverem pomocí požadavků.

## Bootstrap

Sada nástrojů pro tvorbu webu, která je volně ke stažení. Práce s Bootstrapem je jednoduchá, obsahuje předpřipravené šablony založené na HTML a CSS. Slouží pro úpravu typografie, formulářů, tlačítek, navigace, apod. Bootstrap jsem použil pro úpravu vzhledu, a také proto, aby byla webová aplikace responzivní.

## Použité programy

### Sublime text 3

Jde o multiplatformní editor textových souborů a zdrojového kódu psaný v C++ s programovým API v Pythonu. Je rozšiřitelný díky pluginům vytvořeným komunikou Sublime Textu. Samotný editor není open source ani zdarma, ale většina pluginů ano. Mezi hlavní vlastnosti patří přívětivé prostředí, užitečné funkce, rychlost a možnost rozšíření. V Sublime textu jsem pracoval s externím souborem, který obsahoval mapu.



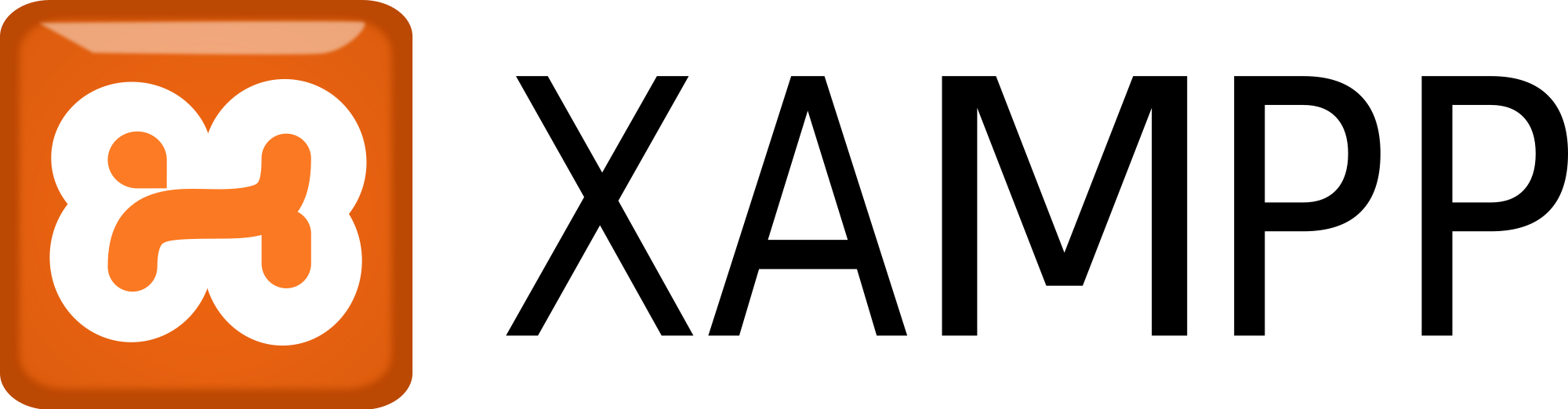
### NetBeans IDE 8.2

Je svobodné, zdarma distribuované integrované vývojové prostředí (IDE) od firmy Oracle Corporation. Primárně určený pro vývoj v programovacím jazyce Java, ale umožňuje vývoj i v jiných jazycích (PHP, C/C++, JavaScript a další). Prostředí je naprogramováno v jazyce Java, a proto lze spustit i na dalších operačních systémech. Osobně jsem si NetBeans vybral, protože se mi toto prostředí zamlouvá a líbí se mi možnost rozšíření a přizpůsobení pomocí pluginů. Většina kódu byla napsaná právě v NetBeans.



### Xampp

Multiplatformní softwarový balíček vyvinutý firmou Apache Friends, obsahuje volně dostupný otevřený software (webový server Apache, databáze MariaDB, interpret PHP a Perlu). Opět jsem využil znalosti ze školy a použil jsem Xampp, abych zprovoznil server.



### MySQL Workbench

Je to nástroj pro vizuální modelování MySQL databází. Umožňuje tvorbu databáze, připojení k jednotlivým databázím a jejich přímé editování, spouštění SQL skriptů a spravování objektů databáze. Patří mezi open source software. Nejprve jsem vytvořil databázi přímo v Xamppu, ale byla potřeba komplexnější databáze s vazbami, proto jsem sáhnul po MySQL Workbench, kde se dá přehledně a jednoduše vytvořit databáze i s vazbami.



Zkratit na max pul strany

# řešená Problematika

## Přesnost lokalizace

Při vývoji webové aplikace jsem se setkal s řadou problémů. Velký problém jsem zaregistroval od prvního vykreslení polohy, protože zobrazený bod na mapě, který měl ukazovat mou polohu, sice fungoval, ale s velkou odchylkou. Tuto odchylku jsem redukoval změnou technologie, ale nepodařilo se mi ji plně potlačit. Když jsem vykreslování polohy zkoušel na více různých místech, tak výsledky byly zcela jiné. V některé síti byla odchylka několik desítek metrů a v jiných sítích byla odchylka v desítkách kilometrech. Zjistil jsem, že když se jedná o stolní počítač připojený kabelem, tak poloha bude velice rozdílná. Je to způsobeno přesměrováním routeru, a také tím, že samotný princip je na základě zjištění pozice podle IP adresy. Na mobilních zařízeních by se tento problém neměl objevovat. To jsem ošetřil nastavením vysoké přesnosti ve funkci pro získání pozice, na úkor však větší náročnosti webové aplikace.

## Připojení dalších uživatelů

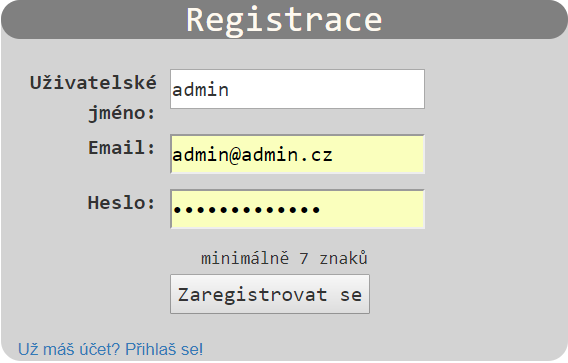
Tento problém nastal, když jsem testoval připojení do webové aplikace prostřednictvím několik dalších zařízení. Při připojení jsem se nedostal ani do možnosti přihlášení uživatele a místo domovské stránky se mi zobrazila mapa s hláškou, která vzniká, když prohlížeč nepovolí přihlašování polohy. Konzole mi psala, že není možné mít podporu funkce pro získání polohy, protože nepoužívám https protokol. Když jsem se více zabýval touto chybou, dostal jsem se k článku o tom, že je potřeba mít SSL certifikát. K webové aplikaci jsem se připojoval pomocí adresy lokálního počítače, který tento SSL certifikát nemá, a proto nemohla fungovat funkce na získávání polohy fungovat.

Prectete si o tomto vice problem vase aplikace musí mit Public SSL auth

# Výsledek

V této kapitole se nachází ukázka, jak vypadá prostředí webové stránky. V budoucnosti mám v plánu styly ještě trochu upravit a zlepšit přehlednost. Stránky jsem se snažil upravovat použitím technologie Bootstrap pro jednoduchost, přehlednost a responzivitu.

## Registrace

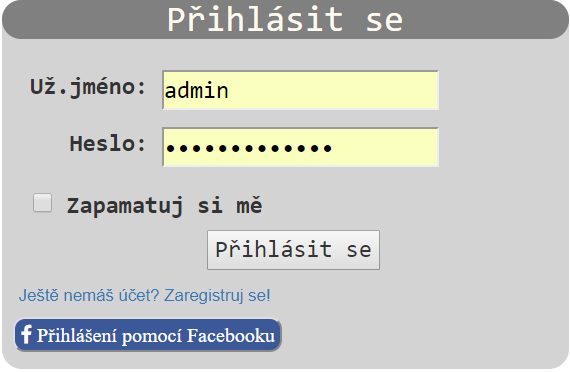


Na tomto výstřižku lze vidět, jak vypadá formulář pro registraci. Uživatel si zvolí své uživatelské jméno, zadá emailovou adresu, která musí být platná v daném formátu a heslo, které musí mít minimálně 7 znaků. Pokud již uživatel má účet, může se pomocí odkazu vlevo dole přesunout rovnou k přihlášení.

*Obrázek č. 6 Registrace*

## Přihlášení

Přihlašovací formulář je podobný registračnímu formuláři, jak svými styly, také i velikostí a požadovanými daty. Pokud má uživatel založený účet, tak jednoduše napíše uživatelské jméno, které si při registraci vybral a zadá heslo. Aby se uživatel nemusel stále znovu přihlašovat, tak může použít funkci „Zapamatuj si mě“ a aplikace si při



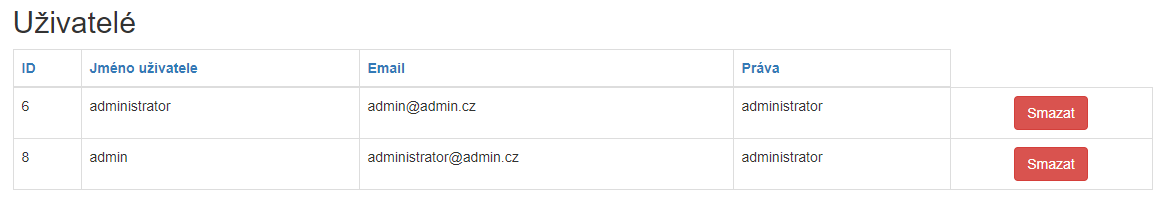
*Obrázek č. 7 Přihlášení*

příští návštěvě bude uživatele pamatovat a tím se může vyhnout opětovnému přihlášení.

Pokud uživatel nemá založený účet, tak si může prostřednictví rychlého odkazu založit nový účet. Do budoucna bude plně zprovozněna funkce pro přihlášení přes sociální síť Facebook.

## Administrační funkce

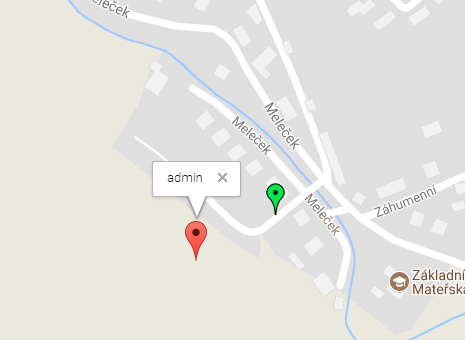
Pro přehlednost a zamezení velkého množství registrovaných uživatelů, má Administrátor možnost smazání účtu přímo prostřednictvím webové aplikace. Z toho důvodu nemusí zasahovat přímo do databáze.



*Obrázek č. 8 Přehled uživatelů a mazání*

## Práce s mapou

### Dva uživatelé

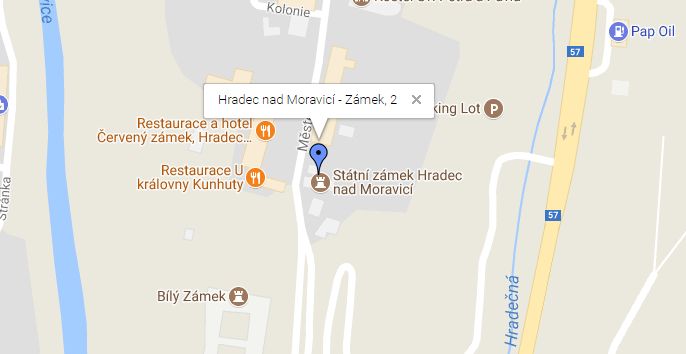


Při přihlášení do webové aplikace se nám zobrazí dva body (neboli markery) na mapě. Jeden červený (uživatel aktuálně přihlášený) a zelený (uživatel z jiného zařízení). Při přejetí myší nad body, kde se nachází uživatelé, se zobrazí okno s uživatelským jménem.

*Obrázek č. 9 Mapa – více uživatelů*

### Oblíbené místo

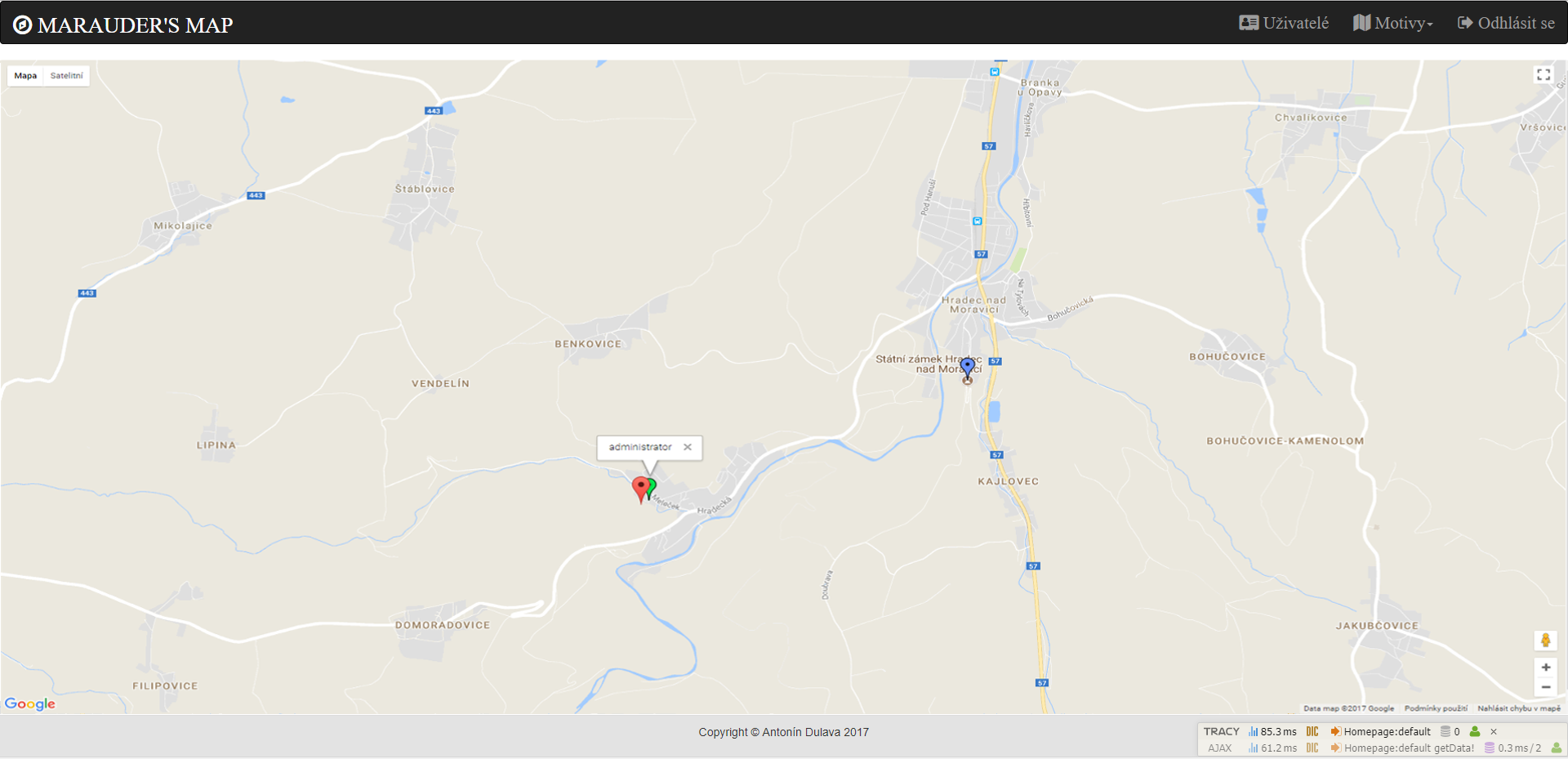
Funkce Oblíbené místo umožní uživateli umístit bod odlišné barvy oproti uživatelům na mapu. Tento bod vidí všichni přihlášení uživatelé. Když uživatel přejede myší na bod, zobrazí se mu název místa. Tento název zadal uživatel, který Oblíbené místo vytvořil. Společně s názvem se zobrazuje i hodnocení místa. Místo je hodnoceno jako ve škole (od 1 do 5).



*Obrázek č. 10 Mapa – Oblíbené místo*

### Náhled

Zastávám se pravidla „v jednoduchosti je krása“, proto tento design vypadá dosti jednoduše. Podle mě není potřeba příliš využívat ostré barvy. Na výstřižku můžete vidět navigační lištu, která nabízí uživateli funkce. Výstřižek je pořízen z účtu s rolí administrátora, proto můžeme vidět funkci „Uživatelé“, která normálním uživatelům není k dispozici. Funkce motivů je spíše takový nástřel do budoucna.



*Obrázek č. 11 Mapa - náhled*

# **Závěr**

Cílem projektu bylo vytvořit webovou aplikace, která umožňuje sledovat pozice připojených uživatelů. Dílčí cíle jako vykreslení mapy, možnost registrace, přihlášení, odhlášení, vykreslení pozic uživatelů a přidání oblíbeného místa s hodnocením byly víceméně splněny.

Tuto webovou aplikaci jsem navrhl pro vlastní použití. V budoucnu bych chtěl tuto aplikaci vylepšit o další funkce jako je např. nastavení motivů, chatovací místnost, vypočítávání vzdálenosti mezi uživateli a zlepšení rozhraní. Nejdůležitější však je pro použití pořídit funkční doménu, na které by webová aplikace mohla bezproblémově fungovat.

Seznam použitýCH INFORMAČNÍCH ZDROJů

[1] Google Maps API Events [online]. 2017 [cit. 2017-12-28]

<<https://developers.google.com/maps/documentation/javascript/events>>.

[2] Google Maps API Geolocation [online]. 2017 [cit. 2017-12-28]

<<https://developers.google.com/maps/documentation/javascript/geolocation>>.

[3] Google Maps API Markers [online]. 2017 [cit. 2017-12-28]  
< <https://developers.google.com/maps/documentation/javascript/markers>>.

[4] Nette AJAX [online]. 2017 [cit. 2017-12-28]

<<https://doc.nette.org/cs/2.4/ajax>>.

[5] PROCHÁZKA, Filip [online]. 2017 [cit. 2017-12-28]

<<https://github.com/Kdyby/Facebook/>>.

[6] Google Maps API Styling [online]. 2017 [cit. 2017-12-28]

<<https://developers.google.com/maps/documentation/javascript/styling>>.

[7] W3Schools [online]. 2017 [cit. 2017-12-28]

<<https://www.w3schools.com/bootstrap>>.