



## **Čištěníulic.cz**

**s námi nezapomenete přeparkovat své auto**

### **Středoškolská odborná činnost**

**Kategorie:** 18. Informatika

**Autor/Author:** Petr Tomášů

**Vedoucí/Supervisor:** Marek Pinkava

Uherské Hradiště, 2013

## **ANOTACE**

TOMÁŠŮ, Petr Čištěníulic.cz: středoškolská odborná činnost. Uherské Hradiště: Střední škola průmyslová, hotelová a zdravotnická Uherské Hradiště, 2013. 27 s.

Cílem práce bylo vytvořit službu, která bude upozorňovat občany na probíhající čištění ulic dle jejich výběru. Dále také vytvořit rozhraní, ve kterém půjde vše upravovat, vkládat data, objednávat tarify, napojit systém na API SMS brány, API banky a vše skloubit tak, aby celá aplikace byla intuitivní, multiplatformní, více-jazyčná, a tudíž ji šlo v budoucnu jednoduše rozšířit i do jiných zemí světa, než je Česká republika.

## **ABSTRACT**

TOMÁŠŮ, Petr Čištěníulic.cz: secondary school professional activity. Uherské Hradiště: Střední škola průmyslová, hotelová a zdravotnická Uherské Hradiště, 2013. 27 s.

The aim of this work was to create a service that will alert citizens to the ongoing cleaning streets of their choice. And also create an interface in which everything goes to edit, enter data, ordering tariffs, connect the system to the SMS Gateway API, API banks and all combine so that the entire application was intuitive, multi-platform, multi-lingual, and therefore it was easily extended in the future to other countries than the Czech Republic.

## PROHLÁŠENÍ

Prohlašuji, že svou práci „Čištěníulic.cz“ jsem vypracoval samostatně pod vedením vedoucího této práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Na práci si vyhrazuji veškerá práva. S vytvářením této práce mi žádné další osoby ani instituce, kromě vedoucího práce nepomáhali.

Jako autor uvedené práce dále prohlašujeme, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědomi následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Uherské Hradiště .....

.....

Petr Tomášů

# OBSAH

Úvod	6
<b>1 Informace o službě čištěníulic.cz</b>	<b>7</b>
<b>2 Problematika a použité technologie</b>	<b>8</b>
2.1 Problematika	8
2.1.1 Uživatelská oprávnění	8
2.1.2 Interaktivní mapy	8
2.1.3 Jazyk a rozšíření do zahraničí	8
2.1.4 Odesílání e-mailů	8
2.1.5 Odesílání SMS	8
2.1.6 Objednávky	9
2.1.7 Automatizované zpracování plateb a přiřazení tarifu	9
2.1.8 Získávání dat o čištění	9
2.1.9 CRON – automatické spouštění funkcí v daném časovém intervalu	10
2.1.10 Práce v týmu a znovupoužitelnost kódu	10
2.1.11 Podpora pro mobilní telefony	11
2.2 Vybrané technologie	11
2.2.1 Apache HTTP server	12
2.2.2 PHP	12
2.2.3 MySQL	12
2.2.4 Nette Framework	12
2.2.5 Architektura MVP	12
2.2.6 Neon soubory	13
2.2.7 GIT	13
<b>3 Implementace vybraných funkcí</b>	<b>15</b>
3.1 Napojení na API SMS brány	15
3.2 Napojení na API banky	15
3.3 Implementace interaktivní mapy	16
<b>4 Představení služby</b>	<b>18</b>
4.1 Úvodní část aplikace	18
4.2 Přihlašování uživatelů	19
4.3 Uživatelský profil	20
4.4 Úprava geografických dat	20

4.5	Import dat . . . . .	21
4.6	Upozorňování uživatelů . . . . .	22
4.6.1	Upozornění na čištění prostřednictvím e-mailu . . . . .	22
4.6.2	Upozornění na čištění prostřednictvím SMS . . . . .	23
4.6.3	Upozornění na expiraci tarifu . . . . .	23
<b>5</b>	<b>Závěr a budoucnost</b>	<b>25</b>
	<b>Literatura</b>	<b>26</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>27</b>

## SEZNAM OBRÁZKŮ

2.1	Schéma architektury MVP. . . . .	13
2.2	Ukázka workflow s GITem. . . . .	14
4.1	Úvodní část aplikace. . . . .	18
4.2	Možnosti přihlášení uživatele. . . . .	19
4.3	Rozhraní pro správu uživatelského profilu. . . . .	20
4.4	Úprava informací o zemi. . . . .	21
4.5	Přehled dat importovaných z XML. . . . .	21
4.6	Příchozí e-mail s upozorněním na čištění. . . . .	22
4.7	Příchozí SMS s upozorněním na čištění. . . . .	23
4.8	Příchozí e-mail s upozorněním na expiraci tarifu. . . . .	24
4.9	Příchozí SMS s upozorněním na expiraci tarifu. . . . .	24

# ÚVOD

Jsem student elektrotechniky, kterého ovšem velmi baví programování a s tím spojené řešení mnoha problémů. Již několik let se zabývám tvorbou webových stránek a podílím se na vývoji redakčního systému Amapet, který je nyní ve verzi 2.0.

Nyní jsem prozkoumal trh a zjistil jsem, že trh s webovými stránkami začíná být zahlcen a bohužel se na něm vyskytuje spousta nekvalitních programátorů. To vše samozřejmě znamená, že cena jde dolů na úkor kvality.

Při zkoumání trhu jsem objevil, že chybí spousta užitečných služeb pro lidi. Rozhodl jsem se tedy, že na základě tohoto zjištění udělám tuto užitečnou službu a bude mne těšit, když bude využívána a lidem tak bude pomáhat předejít nepříjemnostem. Zároveň jsem si zde mohl zkusit implementovat všechny moderní a pokročilé technologie, které mnoho firem ještě třeba ani nepoužívá.

# 1 INFORMACE O SLUŽBĚ ČIŠTĚNÍULIC.CZ

Ve městech po celém světě probíhá během celého roku čištění ulic. Pokud se má nějaká ulice čistit, tak je to v první řadě závazek pro firmu, která čištění provádí, v druhé řadě potom pro lidi, kteří bydlí v dané ulici.

Největším problémem je to, že se může v ulici v době čištění vyskytovat nějaká překážka (nejčastěji automobil), a ta brání v provedení čištění. Pokud taková situace nastane, je firma provádějící čištění nucena takovou překážku odstranit, a to stojí nemalé peníze a velké množství ztraceného času. Náklady na odstranění překážky potom zaplatí její majitel.

Firmy, které provádí čištění v dnešní době používají jako prevenci tu metodu, že do ulice umístí značku s upozorněním na čištění. V praxi to potom dopadá tak, že se i přes tyto upozornění překážky v ulicích vyskytují, a to i proto, že si majitel překážky nevšiml upozornění, nebo na to jednoduše zapomněl.

Služba, která je předmětem této práce má pomoci upozorňovat efektivně a včas obyvatele daných ulic a ušetřit jim tak starosti s placením nemalých pokut za odstranění překážky, což zároveň také ulehčí městům a firmám, které provádí čištění, mnoho problémů s domáháním se úhrady této pokuty.

## Možnosti služby

Aplikace dokáže s dostatečným časovým předstihem upozornit uživatele na čištění v předem zvolených ulicích. Uživatel si může zvolit mezi upozorněním formou e-mailu, SMS nebo může využít obě formy upozornění současně.

## Cena využívání služby

Úplně nejzákladnější upozorňování mailem na jednu ulici je zcela zdarma. Pokud chce být uživatel upozorňován formou SMS nebo na čištění ve více ulicích současně, tak základní tarif stojí ročně 20 Kč. Uživatel má tak po celý rok za cenu jednoho piva jistotu, že mu nebude udělena za překážku v ulici a její odstranění, pokuta.



## 2 PROBLEMATIKA A POUŽITÉ TECHNOLOGIE

### 2.1 Problematika

Ačkoli se to na první pohled nezdá, tak tato aplikace sebou nese značnou problematiku, a to jak z hlediska uživatelského (práce v týmu, mapy, výběr ulic, atd...), tak z hlediska správy tohoto portálu (získávání dat, napojení na API atd...).

#### 2.1.1 Uživatelská oprávnění

Je potřebné, aby uživatelé měli svůj účet, kde si můžou nastavovat veškerá nastavení. Dále je nutné, aby bylo uděláno rozhraní pro správce portálu. Toto vede k tomu, že je potřeba mít pro každou skupinu uživatelů různá oprávnění. V celé aplikaci se musí dbát na to, aby se nikdo nedostal tam, kam nemá. Aplikace využívá ozkoušený ACL model.

#### 2.1.2 Interaktivní mapy

Pro jednoduchost a uživateli tolik oblíbenou dynamičnost je vhodné použít interaktivní mapy, kde jsou vybrané ulice přehledně zobrazeny a jsou u nich další informace. Bylo nutné vybrat vhodné mapové podklady pro tuto funkčnost.

#### 2.1.3 Jazyk a rozšíření do zahraničí

Vzhledem k tomu, že čištění ulic probíhá po celém světě, je nutné také myslet na to, že aplikace musí podporovat více jazyků.

#### 2.1.4 Odesílání e-mailů

Odesílání mailů se provádí pomocí SMTP serveru. Největším problémem je to, že tento server má omezený počet požadavků, a tudíž se musí posílat e-maily po dávkách, aby nedošlo k jeho přehlcení. Vše je řízeno databází, kde jsou uvedeny dva stavy, a to ČEKAJÍCÍ NA ODESLÁNÍ a ODESLÁNO. Problém může nastat, když je na SMTP serveru porucha, a proto musí být vše ošetřeno.

#### 2.1.5 Odesílání SMS

Odesílání SMS se provádí pomocí SMS brány [10]. Je tedy nutno napojit vše na její API rozhraní. Vzhledem k tomu, že upozornění formou SMS je placená služba, je

nutné dbát na garanci toho, že se SMS opravdu odešle. Problém může nastat, pokud je API pro odesílání SMS nedostupné nebo je číslo příjemce neplatné, uživatel má vypnutý mobil a další. Všechny tyto problémy se musí ošetřit.

### **2.1.6 Objednávky**

Problém objednávky je v tom, že se musí načíst parametry objednaného tarifu, uživatelské údaje a nakonec na základě tohoto vygenerovat platební údaje, zobrazit je a také je odeslat uživateli mailem.

### **2.1.7 Automatizované zpracování plateb a připsání tarifu**

Aby bylo vše co nejrychlejší a s připisováním tarifu nebyla žádná starost, je aplikace napojena na API banky [9], a tudíž je vše prováděno automaticky. K připsání tarifu dochází na dobu jednoho roku, ovšem je nutné řešit to, zda uživatel již nemá zaplacený tarif, jestli náhodou nepřechází na vyšší či nižší tarif, atd. Z toho tedy vyplývá, že k době jednoho roku musí být poměrně přičteny zbývající dny z předchozího tarifu.

Je nutno ošetřit, když uživatel zadá špatný variabilní symbol, malou nebo velkou částku, a proto jsou všechny tyto chyby ukládány a vše se tak může manuálně dořešit. Další problém je, když je API banky nedostupné, to všechno se musí ošetřit, aby nedocházelo k chybám a s nimi spojeným nepříjemnostem. Uživatel takto tedy není o nic ochuzen a přispívá to jen k jeho spokojenosti.

### **2.1.8 Získávání dat o čištění**

Aplikace stojí na správných datech o termínech čištění v daných ulicích. Aby města a firmy provádějící čištění dodávaly ochotně data, musíte pro ně vytvořit pohodlné rozhraní, kde „pár kliknutími“ nahrají data v daném formátu. Jako univerzální formát je nejlépe použít XML 1.

Takový soubor se rozparsuje a následně se s daty pracuje velmi elegantně. Samozřejmě lze také vše přidávat ručně. Problém může nastat, pokud již daný termín čištění existuje a další problémy s duplikátními daty. Tyto problémy není problém ošetřit, tudíž je nakonec vše tak, jak má.

```

<data>
  <country_name>Česká republika</country_name>
  <region>
    <region_name>Zlínský kraj</region_name>
    <city>
      <city_name>Uherské Hradiště</city_name>
      <part>
        <part_name>Mařatice</part_name>
        <street>
          <street_name>Lomená</street_name>
          <cleaning>
            <date_from>2012-04-05 09:30:00</date_from>
            <date_to>2012-04-05 16:30:00</date_to>
          </cleaning>
          <cleaning>
            <date_from>2012-04-08 09:30:00</date_from>
            <date_to>2012-04-08 16:30:00</date_to>
          </cleaning>
        </street>
      </part>
    </city>
  </region>
</data>

```

Program 1: Ukázka XML feedu.

## 2.1.9 CRON – automatické spouštění funkcí v daném časovém intervalu

Aby všechny akce typu zpracování plateb, odesílání upozornění, apod. byly automatizované, je nutné spouštět funkce pro tyto akce určené v určeném čase nebo časovém intervalu. Nastavení parametrů CRONu vzájemně souvisí s mnoha funkcemi aplikace, je tedy nutné postupovat velmi obezřetně.

Dále je také nutné řešit to, aby CRON funkce nemohl spustit kdokoli, takže každou CRON funkci musí být možné spustit pouze pomocí konzole.

### 2.1.10 Práce v týmu a znovupoužitelnost kódu

Jelikož v budoucnu bude potřeba pracovat v týmu, bude důležité co nejefektivněji spolupracovat a spojovat odvedenou práci. Musel tedy být zvolen způsob, jakým to provést. Jako další bylo potřeba docílit toho, aby bylo možné části aplikace jednoho nebo druhého člena z týmu znovupoužít. Bylo tedy nutné zvolit vhodný způsob programování, a tedy i jazyk, který tento způsob bude podporovat.

### 2.1.11 Podpora pro mobilní telefony

Aby služba byla využívána, co nejvíce lidmi, je nutné vše přizpůsobit i pro mobilní zařízení jako jsou nyní stále populárnější smartphony a tablety.

## 2.2 Vybrané technologie

V této části bude popsáno řešení problematiky prakticky. Jako první bylo nutno zvolit programovací jazyk a architekturu, na které bude moci být aplikace postavena, a která dále umožní aplikaci jednoduše rozšiřovat.

Programovacím jazykem pro tuto aplikaci byl zvolen PHP ve verzi 5.4. Vzhledem k tomu, že je nutné, aby aplikace běžela na všech platformách, tak je to nejlepší způsob, jelikož vše poté běží v cloudu a aplikace je dostupná uživatelům různých platforem. Pro rozhraní aplikace je používána stále se rozvíjející technologie HTML5, CSS3 a Javascript (AJAX) [1].

Následně bylo nutné zvolit mapové podklady – pro tento účel poslouží Open street map [7]. Jedná se o opensource projekt, který nabízí volně dostupná geografická data, jenž jsou získávána integrací dat z různých zdrojů, především individuálním sběrem dat pomocí GPS zařízení a v této době je jejich kvalita na velice dobré úrovni. Pro obsluhu těchto map byla použita knihovna LeaFlet.

Aby byl ve všem pořádek, tak jako architektura aplikace byla zvolena MVP (Model-View-Presenter) architektura. To také ulehčí vývoj a ušetří nervy při práci v týmu. Jako hnací motor aplikace je použit Nette Framework, který podporuje architekturu MVP, disponuje mnoha skvělými funkcemi a má velkou základnu české komunity. Také se skrze něj velmi dobře používá AJAX. Aplikace pracuje s mnoha daty mezi kterými potřebujeme vytvářet relace, a proto jako datové úložiště zvolíme MySQL.

Práci je důležité efektivně kompletovat, jednoduše se vracet, pokud se vývojář vydá špatnou cestou. K tomuto slouží verzovací systém GIT. Každý z vývojového týmu si díky němu může vše přehledně verzovat, a tak má daleko lepší přehled o své práci.

Poslední, ale velmi důležitou věcí je snadná údržba aplikace a velmi jednoduchá úprava určitých zákonitostí (e-mail, název aplikace, cena tarifů, parametry tarifů, texty mailů a SMS, údaje pro přístup k API, atd.). Tuto konfiguraci skvěle umožní Neon konfigurační soubory. Celá aplikace bude běžet na Apache HTTP serveru, jelikož podporuje všechny technologie, které chceme použít.

### 2.2.1 Apache HTTP server

Apache HTTP Server je softwarový webový server s otevřeným kódem pro GNU/Linux, BSD, MAC OS X, Microsoft Windows a další platformy. V současné době dodává prohlížečům na celém světě většinu internetových stránek. Plně podporuje propojení s PHP a MySQL, které v aplikaci potřebujeme.

### 2.2.2 PHP

PHP (Hypertext preprocessor) je skriptovací programovací jazyk. Je určený především pro programování dynamických internetových stránek a webových aplikací. PHP lze použít i k tvorbě konzolových a desktopových aplikací. Při použití PHP pro dynamické stránky jsou skripty prováděny na straně serveru, k uživateli je přenášén až výsledek jejich činnosti [6].

### 2.2.3 MySQL

MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními [5].

### 2.2.4 Nette Framework

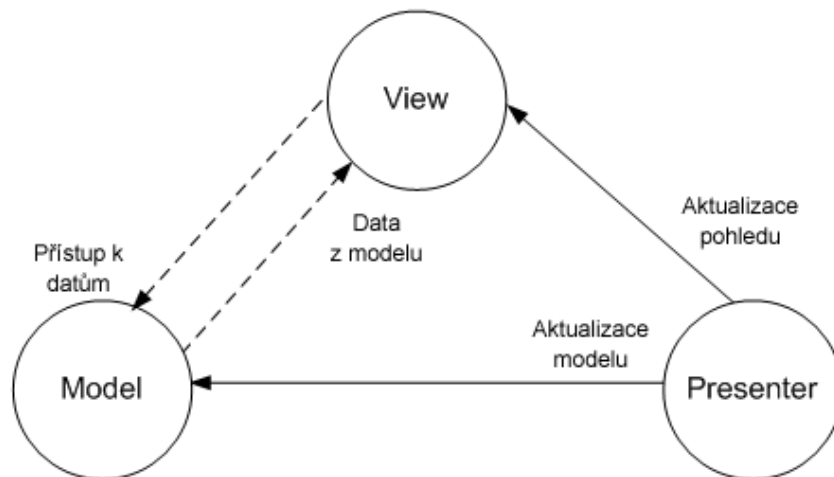
Populární nástroj pro vytváření webových aplikací v PHP 5. Při programování dává volnost a nepřiděává vrásky. Eliminuje bezpečnostní rizika, obsahuje velice propracovanou ochranu proti XSS (Cross-site scripting), CSRF (Cross-site request forgery) apod., podporuje AJAX a znovupoužitelnost kódu. Je postaven na architektuře MVP. Jeho největší silou je rychlost a snadné ladění chyb pomocí laděnky [8].

### 2.2.5 Architektura MVP

Model-View-Presenter je softwarová architektura, která vznikla z potřeby oddělit u aplikací s grafickým rozhraním kód obsluhy (presenter) od kódu aplikační logiky (model) a od kódu zobrazujícího data (view) 2.1. Tím jednak aplikaci zpřehledňuje, usnadňuje budoucí vývoj a umožňuje testování jednotlivých částí zvlášť.

#### Model

Model je datový a zejména funkční základ celé aplikace. Je v něm obsažena aplikační logika. Jakákoliv akce uživatele (přihlášení, vložení dat, změna hodnoty v databázi) představuje akci modelu. Model si spravuje svůj vnitřní stav a ven nabízí pevně



Obr. 2.1: Schéma architektury MVP.

dané rozhraní. Voláním funkcí tohoto rozhraní můžeme zjišťovat či měnit jeho stav. Model o existenci view nebo presenteru neví.

### View

View, tedy pohled, je vrstva aplikace, která má na starost zobrazení výsledku požadavku. Obvykle používá šablonovací systém a ví, jak se má zobrazit určitá komponenta nebo výsledek získaný z modelu.

### Presenter

Řadič, který zpracovává požadavky uživatele a na jejich základě pak volá patřičnou aplikační logiku (tj. model) a poté požádá view o vykreslení dat.

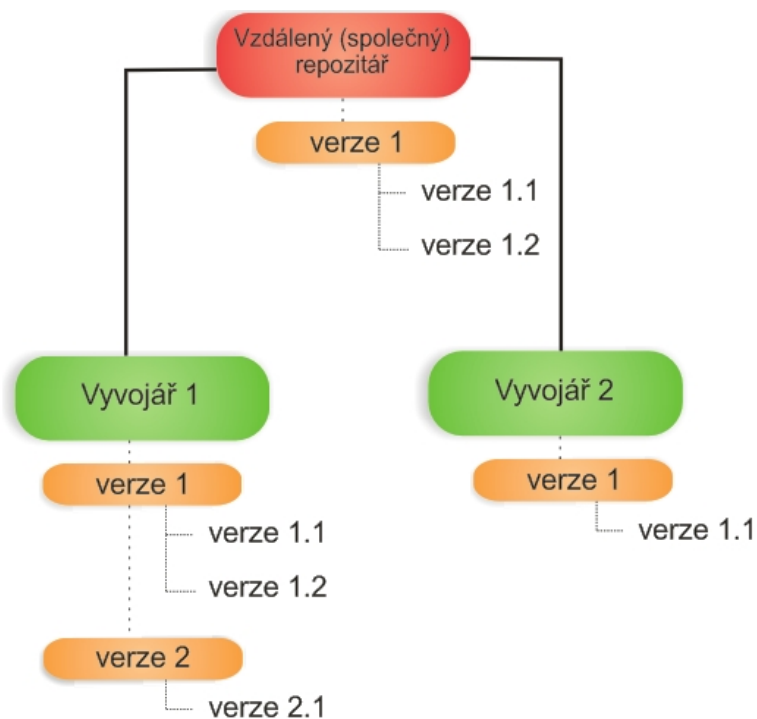
## 2.2.6 Neon soubory

Neon soubory jsou soubory, které se používají pro konfiguraci různých částí programu. Mají příponu `.neon`. Díky nim lze jednoduše změnit parametry čehokoli (názevu aplikace, jazyků, parametrů tarifu, adresy pro API, atd...). Tyto soubory jsou svou syntaxí velmi přehledné a po rozparsování s jednotlivými částmi pracujeme, jako s prvky pole.

## 2.2.7 GIT

GIT byl původně vytvořen pro verzování při vývoji jádra Linuxu. Nyní je tento nástroj hojně využíván mnoha vývojovými týmy softwarových firem. Největší výhodou GITu je to, že lze vše verzovat, ať už v rámci celého týmu, nebo v rámci

jednoho člena týmu (tzv. vytvářením větví) než je s danou prací spokojen a spojí ji s prací ostatních vývojářů (zpravidla má každý vývojář svůj tzv. repozitář a jeden je společný viz obr. 2.2, lze samozřejmě volit i jinou strukturu repozitářů) [12].



Obr. 2.2: Ukázka workflow s GITem.

Změny lze potom jednoduše sledovat v editorech – takto víte, co bylo přidáno, a co naopak odebráno. Každý řádek, který jste odebral nebo změnil je různě podbarven – přehlednější už to snad ani být nemůže.

## 3 IMPLEMENTACE VYBRANÝCH FUNKCÍ

Tato kapitola popisuje implementaci vybraných funkcí. Je zde stručný popis toho, jak daný algoritmus funguje a je tak možnost nahlédnout alespoň do určité části mého zdrojového kódu.

### 3.1 Napojení na API SMS brány

Aplikace odesílá SMS upozornění prostřednictvím SMS brány. Jelikož je SMS zpoplatněna, je nutné vše dokonale ošetřit. API funguje na principu zaslání HTTP požadavku, zpětně je vrácen výsledek operace a aplikace tak může dále vše ošetřit. Problémy jsou například nedostupnost API, příjemce SMS neexistuje, příliš dlouhý text, nedoručeno atd. Aplikace má všechny tyto situace úspěšně vyřešené a může tak uživatelům ručit, že upozornění opravdu bude doručeno.

V kódu 2 je ukázka metody odeslání SMS, které ovšem předchází mnoho kontrolních procesů, než vůbec k zavolání této metody dojde. Metoda přebírá dva parametry, první je číslo příjemce a druhý je text zprávy. Funkce vrací odpověď od SMS brány.

```
function sendSMS($number, $message)
{
    $get_query = $this->getAuthData()."&action=send_sms&number=" .
        urlencode($number)."&message=" .urlencode($message);
    ob_start();
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL,$this->context->parameters['smsGate']
        ['apiScript']."?$get_query");
    $res = curl_exec($ch);
    curl_close($ch);
    $res = ob_get_contents();
    ob_end_clean();

    return $res;
}
```

Program 2: Funkce pro odeslání SMS upozornění.

### 3.2 Napojení na API banky

Jelikož služba nabízí také zpoplatněné tarify bylo nutné ji napojit na bankovní API, aby mohla automatizovaně zpracovávat platby a připisovat uživatelům nový tarif. Vzhledem k tomu, že se jedná o funkci, kde by mohlo při nedůkladném ošetření chyb dojít ke ztrátě peněz, bylo nutné při návrhu algoritmu postupovat velmi obezřetně.



První věcí je zbytečnost stahování veškerých dat o účtu, a proto jako zarážka pro data je používáno datum posledního stažení dat, což sebou potom přináší problém, který algoritmus v aplikaci samozřejmě řeší. Bylo velmi nutné ošetřit, pokud by došlo k chybě na straně banky nebo na straně aplikace (například výpadkem serveru atd.).

Algoritmus také zároveň počítá na základě varianty a data expirace předchozího tarifu datum expirace nového tarifu, který právě uživatel zaplatil, a to tak aby uživatel o nic nepřišel a vedlo to k jeho spokojenosti.

Pro tento účel byl vytvořen univerzální vzorec pro výpočet:

NP – nová cena tarifu

PP – předchozí cena tarifu

ZDDE – zbývající dny do expirace starého tarifu

PLUS – počet dnů navíc k novému tarifu

TODAY – dnešní datum

DE – nové datum expirace

$$\text{PLUS} = (\text{NP}/365)/(\text{PP}/365)*\text{ZDDE}$$

$$\text{DE} = \text{TODAY} + 365 + \text{PLUS}$$

Algoritmus ošetřuje i to, pokud by byla náhodou zaplacená částka malá, velká, variabilní symbol neexistoval atd. Pro tento účel je vše zaznamenáváno a s uživatelem se tak v případě výskytu problému může vše velmi jednoduše dohledat a nepříjemnost vyřešit. Uživatel je o připsání tarifu informován pomocí SMS a emailu.

### 3.3 Implementace interaktivní mapy

Pro zobrazení interaktivních map je v aplikaci používána knihovna Leaflet.js, která využívá mapové podklady projektu openstreetmap. V ukázce kódu 3 je vytvořená funkce `initMap` pro inicializaci mapy. Tato funkce přebírá dva parametry první je identifikátor HTML prvku, ve kterém má být mapa zobrazená a druhý je objekt s nastavením mapy. Tato funkce provede inicializaci mapy a její potřebné nastavení a následně vrátí instanci mapy. Další funkce slouží pro přidání značky do mapy. Funkce přebírá tři parametry, první je instance mapy, do které má být bod vložen, druhý je objekt s GPS souřadnicemi a třetí je objekt s nastavením vlastností bodu.

```
initMap: function(id, options) {
  if ($('#' + id).length <= 0) return null;

  var map = L.map(id).setView(options.center, options.zoom);

  L.tileLayer('http://{s}.tile.cloudmade.com/
    BC9A493B41014CAABB98F0471D759707/997/256/{z}/{x}/{y}.png').
    addTo(map);

  return map;
},

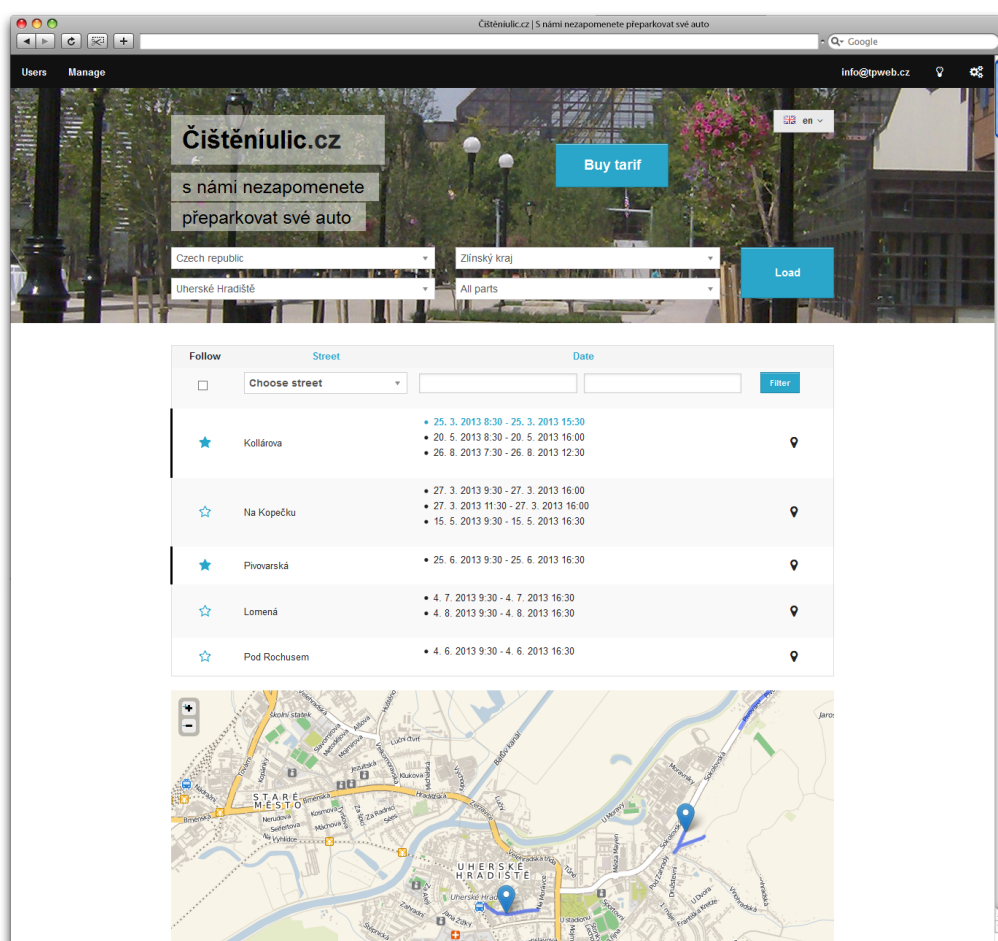
addMarker: function(map, latLng, options) {
  return L.marker(latLng, options).addTo(map);
}
```

Program 3: Vytvořená funkce pro inicializaci mapy a přidání značky do mapy.

## 4 PŘEDSTAVENÍ SLUŽBY

V této kapitole budou popsány a představeny jednotlivé části aplikace. Je zde tedy možnost se ve stručnosti a bez zbytečných složitostí seznámit s tím, jak služba vypadá.

### 4.1 Úvodní část aplikace



Obr. 4.1: Úvodní část aplikace.

V horním panelu se nachází odkazy pro správu uživatelů, správu geografických dat a termínů čištění, uživatelský profil, pomoc, hlášení problémů, přihlášení a odhlášení uživatele.

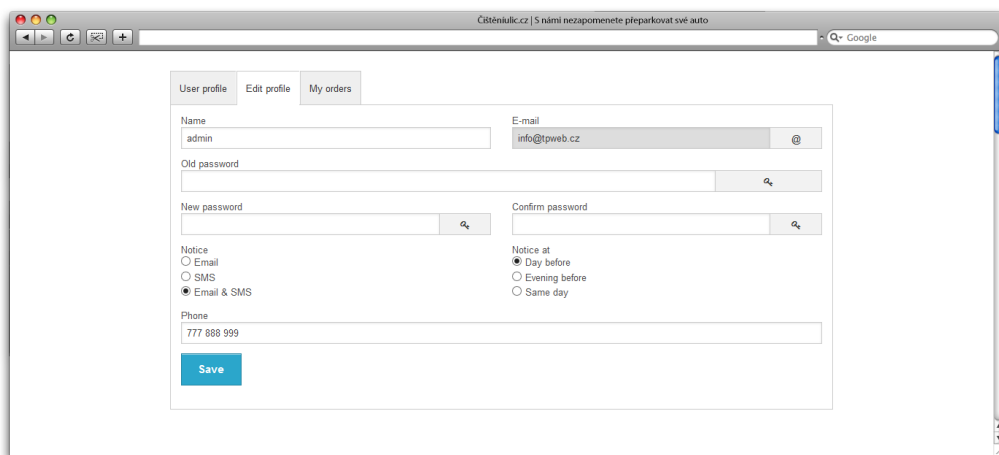
V hlavičce se nachází odkaz pro objednání tarifu a také výběr jazykového prostředí aplikace. Dále si v hlavičce můžeme všimnout selectboxů pro filtraci podle

geografických celků (zemí, regionů, měst a městských částí). Jakmile máme filtr nastaven, v tabulce níže se nám zobrazí přehled ulic a termíny čištění v těchto ulicích. V celé tabulce si můžeme veškerá data intuitivně filtrovat.

Velmi důležitou součástí jsou hvězdičky zobrazené u každé z ulic. Pomocí těchto hvězdiček si můžeme nastavovat ulice, na jejichž čištění chceme být upozorňováni. Plná hvězdička značí to, že ulice je sledována a prázdná to, že ulice sledována není. Uživateli je také umožněno nastavování sledovaných ulic pomocí interaktivní mapy, která se nachází pod přehledovou tabulkou. Pozice a přiblížení interaktivní mapy se vždy mění na základě konkrétního geografického celku, který si uživatel vybral.

## 4.2 Přihlašování uživatelů

Aby uživatel mohl službu využívat plnohodnotně, je nutné se přihlásit. Abychom uživatelům vše ulehčili co nejvíce, tak kromě klasického přihlášení pomocí e-mailu a hesla se uživatelé mohou do aplikace přihlašovat pomocí sociálních sítí. Uživatel se tak zbytečně nemusí zdržovat registrací. Služba disponuje přihlášením pomocí třech největších sociálních sítí. Jsou to Facebook, Twitter a Google+.



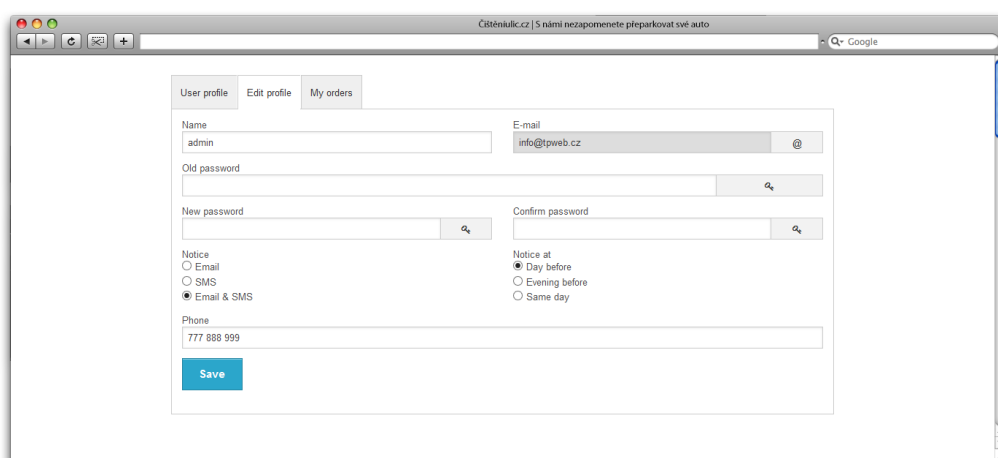
Obr. 4.2: Možnosti přihlášení uživatele.

Na obr. 4.2 vidíme přihlašovací formulář pro klasické přihlášení pomocí mailu a hesla. Uživatel si také může zvolit, zda chce být přihlášen trvale. V pravé části obr. 4.2 můžeme vidět výběr sociální sítě skrze kterou se chceme přihlásit.

Uživatel také může zažádat o zaslání nového hesla, a pokud ještě nemá účet, může se odsud přemístit na stránku, na které účet vytvoří pomocí registrace.

## 4.3 Uživatelský profil

Každý uživatel má svůj vlastní profil, který si může libovolně nastavovat a nastavení kdykoli měnit. Může zde vidět jednoduchý přehled (aktuální tarif, metoda upozorňování, expirace tarifu a další parametry, které má aktuálně nastaveny). Uživatel si zde také může nastavit metodu upozornění, ale také časový předstih, se kterým chce být upozorňován na čištění. Je zde taktéž možno měnit telefonní číslo pro zasílání upozornění formou SMS. Uživatel si také může kdykoli změnit heslo, a to tak, že zadá staré heslo a následně heslo nové, které musí pro kontrolu ještě v dalším formulářovém prvku zopakovat).



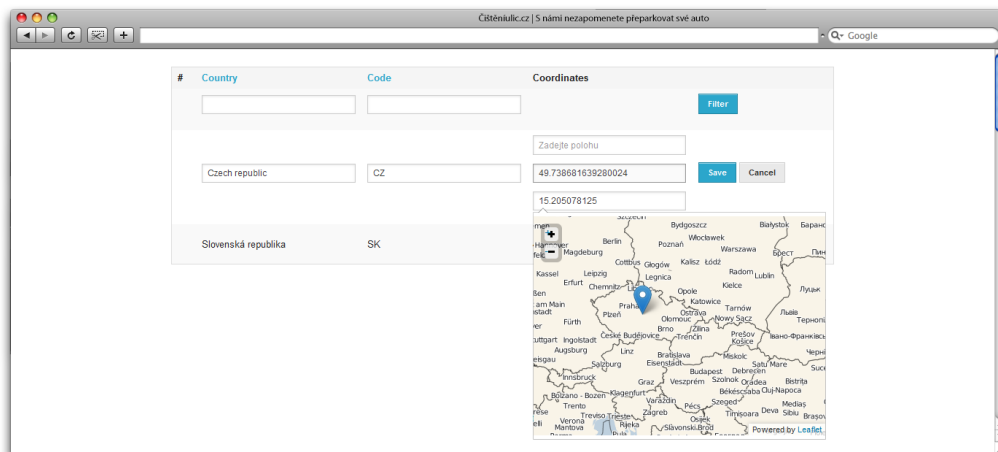
Obr. 4.3: Rozhraní pro správu uživatelského profilu.

Dále je zde přehled objednávek, které může kdykoli rušit, zobrazovat si detaily s parametry objednávky a platebními údaji, či sledovat jejich stav.

## 4.4 Úprava geografických dat

Jsou zde pole pro zadání názvu země a kódu, který vychází z normy ISO 3166-1 alpha-2 [11]. Dále jsou zde pole pro zadání zeměpisné šířky a délky, což lze pohodlně provést pomocí interaktivní mapy.

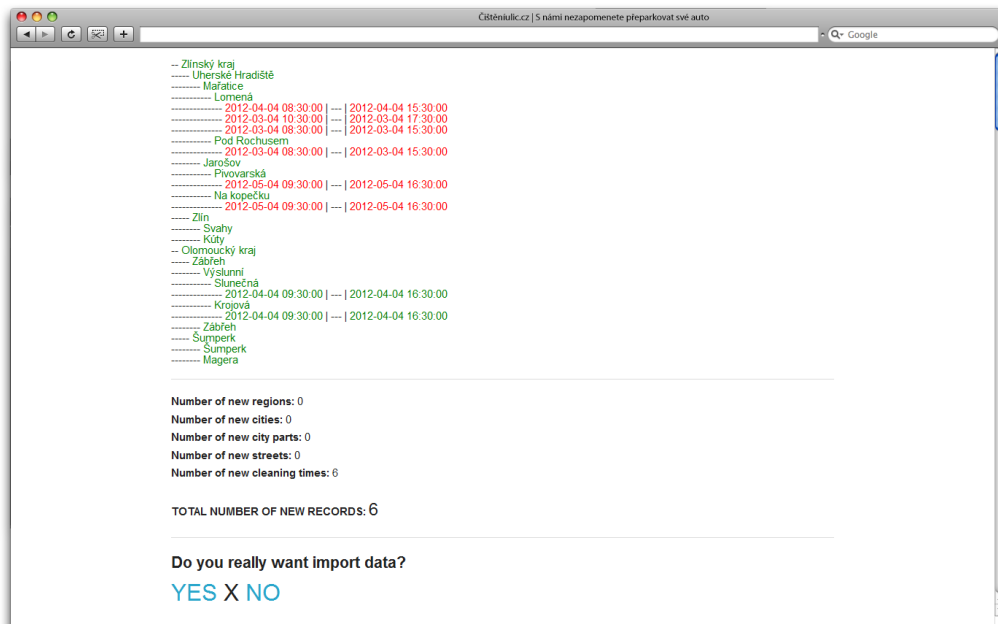
Vložení nové země funguje na stejném principu. Obdobným způsobem také funguje přidání a editace dalších geografických celků (regiony, města, městské části a ulice).



Obr. 4.4: Úprava informací o zemi.

## 4.5 Import dat

Vzhledem k tomu, že aplikace pracuje s velkým množstvím dat, nebylo by příliš efektivní tyto data vkládat ručně. Pro tento účel aplikace disponuje rozhráním, skrze které lze data importovat pomocí XML souboru, který lze jednoduše vytvořit například i z CSV souboru nebo souboru JSON, či mnoha dalších.



Obr. 4.5: Přehled dat importovaných z XML.

Nejprve vybereme XML soubor, který chceme zpracovat. Také určíme zemi, které přísluší data v tomto souboru. Následně vidíme přehled všech dat (obr. 4.9), které XML soubor obsahuje (zeleně podbarvené záznamy již existují, červeně podbarvené

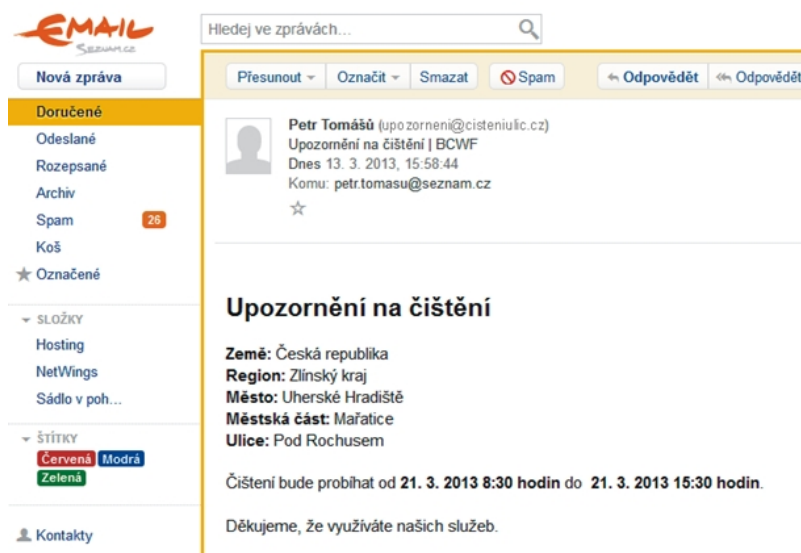
záznamy budou vloženy). Pokud jsme spokojeni, stačí jen potvrdit a data se během okamžiku vloží. Na konci importu uvidíme přehled toho, kolik bylo vloženo nových záznamů, což nám slouží pro efektivní zpětnou kontrolu.

## 4.6 Upozorňování uživatelů

Upozorňování uživatelů na různé akce a oznámení služby probíhá pomocí emailů nebo SMS (Short Message Service).

### 4.6.1 Upozornění na čištění prostřednictvím e-mailu

Uživateli dojde s nastaveným předstihem e-mail s upozorněním na čištění v dané ulici.



Obr. 4.6: Příchozí e-mail s upozorněním na čištění.

### 4.6.2 Upozornění na čištění prostřednictvím SMS

Uživateli dojde s nastaveným předstihem SMS zpráva s upozorněním na čištění v dané ulici.

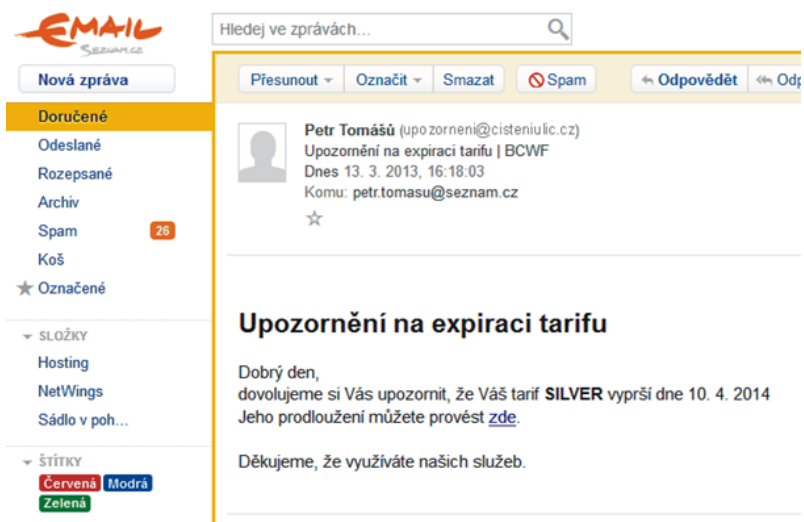


Obr. 4.7: Příchozí SMS s upozorněním na čištění.

### 4.6.3 Upozornění na expiraci tarifu

Aby uživatel mohl i po roce naplno využívat službu a nezapomněl obnovit svůj tarif, služba automaticky upozorní na expiraci jeho tarifu. Upozorněním na expiraci tarifu přijde týden před expirací, Pokud si uživatel tarif neprodlouží, dojde mu ještě upozornění tři dny před expirací.





Obr. 4.8: Příchozí e-mail s upozorněním na expiraci tarifu.



Obr. 4.9: Příchozí SMS s upozorněním na expiraci tarifu.

## 5 ZÁVĚR A BUDOUCNOST

Služba je nasazena do ostrého provozu, je potřeba doladit poslední detaily a poté je nutné zapracovat na získání potřebných informací od dalších měst popř. firem, které provádí čištění. Jedním z nejdůležitějších úkolů bude také dostat do podvědomí veřejnosti informace o tom, že taková služba existuje a že jí můžou využívat ke svému prospěchu.

### **Budoucí práce**

Abychom uživatelům ulehčili výběr jejich konkrétní lokality, bude vhodné implementovat funkci, která podle aktuální pozice uživatele vybere požadovanou lokalitu. K implementaci této funkčnosti nám nejlépe poslouží Geolocation API z nově vznikající technologie HTML5.

Protože jsou mobilní platformy stále více populárnější a mnoho uživatelů přistupuje k internetu pomocí mobilních zařízení více než pomocí počítače, bude nutné ještě více optimalizovat rozhraní aplikace pro tyto zařízení popř. bude vhodné vytvořit nativní aplikace pro jednotlivé platformy.

## LITERATURA

- [1] LAWSON, B. – SHARP, R. *Introducing HTML5*. 2. vyd. Berkeley: New Riders Press, 2011. 312 s. ISBN 978-0-132-79297-4.
- [2] GASSTON, P. *The Book of CSS3*. San Francisco: No Starch Press, 2011. 278 s. ISBN 978-1-59327-286-9.
- [3] FLANAGAN, D. *JavaScript: The Definitive Guide*. Sebastopol: O'Reilly, 2011. 1098 s. ISBN 978-0-596-80552-4.
- [4] RESIG, J. *JavaScript a Ajax – Moderní programování webových aplikací*. Brno: Computer Press, 2007. 360 s. ISBN 978-80-251-1824-5.
- [5] Kolektiv. *PHP5, MySQL, Apache*. Praha: COMPUTER PRESS, 2006. 816 s. EAN 9788025110737.
- [6] VRÁNA, J. *1001 tipů a triků pro PHP*. Praha: COMPUTER PRESS, 2011. 456 s. EAN 9788025129401.
- [7] *OpenStreetMap Wiki* [online]. 2012 [cit. 2012-4-25].  
Dostupné z <[http://wiki.openstreetmap.org/wiki/Main\\_Page](http://wiki.openstreetmap.org/wiki/Main_Page)>.
- [8] *Dokumentace / Nette Framework* [online]. 2013 [cit. 2013-2-14].  
Dostupné z <<http://doc.nette.org/cs/>>.
- [9] *FIO / API bankovníctví* [online]. 2013 [cit. 2013-2-14].  
Dostupné z <[http://www.fio.cz/docs/cz/API\\_Bankovnictvi.pdf](http://www.fio.cz/docs/cz/API_Bankovnictvi.pdf)>.
- [10] *Dokumentace / SMS connect* [online]. 2013 [cit. 2013-2-14].  
Dostupné z <[http://www.smsbrana.cz/dokumenty/smsconnect\\_http.pdf](http://www.smsbrana.cz/dokumenty/smsconnect_http.pdf)>.
- [11] *Standard ISO 3166-1* [online]. 2013 [cit. 2013-2-14].  
Dostupné z <[http://cs.wikipedia.org/wiki/ISO\\_3166-1](http://cs.wikipedia.org/wiki/ISO_3166-1)>.
- [12] *GIT - documentation* [online]. 2013 [cit. 2013-2-14].  
Dostupné z <<http://git-scm.com/documentation>>.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AJAX Asynchronous JavaScript and XML

API Application Programming Interface – aplikační programové rozhraní

CSRF Cross-site request forgery

CSS Cascading Style Sheets

CSV Comma-separated values

DOM Document Object Model – objektový model dokumentu

GPS Global Positioning System – globální polohový systém

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

JSON JavaScript Object Notation

MVC Model-View-Controller

MVP Model-View-Presenter

PHP Hypertext preprocessor

SMS Short Message Service

WWW World Wide Web

XHTML Extensible Hypertext Markup Language

XML eXtensible Markup Language

XSS Cross-site scripting