

System ekonomického řízení přepravy zboží v široké distribuční síti

The Economical Control of an Assets Transportation within Complicated Distributional Relations

Bc. Zbyněk Zamazal

Diplomová práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Zbyněk Zamazal**
Osobní číslo: **A11723**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **kombinovaná**
Téma práce: **Systém ekonomického řízení přepravy zboží v široké distribuční síti**

Zásady pro vypracování:

1. Vypracujte stručný přehled zadané problematiky, včetně popisu nejvýznamnějších dostupných řešení.
2. Naprogramujte softwarový systém, který bude schopen ekonomicky kvantifikovat jednotlivé úseky na trase svozu zboží podle zadaných kritérií.
3. Umožněte uživateli získat informaci o výnosu z jednotlivých úseků a dynamicky se rozhodnout o vypuštění těch nevýnosných.
4. Navrhněte tento systém jako on-line serverovou aplikaci schopnou otevřeně komunikovat se softwarovým řešením libovolné třetí strany.
5. Věnujte pozornost zabezpečení aplikace.
6. Vyvinutý systém testujte na teoretickém příkladu, eventuálně v podmínkách reálné společnosti.
7. Na základě předchozího bodu odladte výslednou aplikaci a připravte ji pro nasazení do reálného provozu.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. SHARP, John. Microsoft Visual C-Sharp 2008: krok za krokem. Vyd. 1. Brno: Computer Press, 2008, 592 s. ISBN 978-80-251-2027-9.
2. MAREŠ, Amadeo. 1001 tipů a triků pro C-Sharp 2010. Brno: Computer Press, 2011, 416 s. ISBN 978-80-251-3250-0.
3. Google Maps API – Google Developers. GOOGLE. <https://developers.google.com/> [online]. 17.12.2012 [cit. 2013-01-24]. Dostupné z: <https://developers.google.com/maps/>
4. Javascript – návody, objekty, příklady. JANOVSKEÝ, Dušan. Jak psát web, návod na html stránky [online]. 6.12.2012 [cit. 2013-01-24]. Dostupné z: <http://www.jakpsatweb.cz/javascript/>
5. ZELINKA, Ivan. Umělá inteligence I: Neuronové sítě a genetické algoritmy. 1. vyd. Brno: VUT v Brně, 1998, 126 s. ISBN 80-214-1163-5.
6. ZELINKA, I., Z. OPLATKOVÁ, M. ŠEDA, P. OŠMERA a F. VČELARĚ. Evoluční výpočetní techniky: principy a aplikace. 1. české vyd. Praha: BEN, 2009, 534 s. ISBN 978-80-7300-218-3.
7. OWASP. OWASP Top10 – 2010. [online]. [cit. 2012-10-24]. Dostupné z: https://www.owasp.org/images/0/0f/OWASP_T10_-_2010_rc1.pdf

Vedoucí diplomové práce:

Ing. Bc. Pavel Vařacha, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

22. února 2013

Termín odevzdání diplomové práce:

22. května 2013

Ve Zlíně dne 22. února 2013

prof. Ing. Vladimír Vašek, CSc.

děkan



doc. Mgr. Roman Jašek, Ph.D.

ředitel ústavu

ABSTRAKT

Cílem této diplomové práce je vypracovat stručný přehled nejvýznamnějších nástrojů pro ekonomické řízení přepravy zboží, které jsou v současnosti dostupné. Na základě přehledu je proveden výběr vhodného mapového podkladu včetně rozhraní, které umožní získání údajů nezbytných pro ekonomickou kvantifikaci jednotlivých úseků na trase rozvozu zboží. V praktické části je vytvořena funkční aplikace, která umožňuje uživateli získat informaci o výnosu z jednotlivých úseků a dynamicky se rozhodnout o vypuštění těch nevýnosných. Je vyřešeno rozhraní, které je schopné otevřeně komunikovat s databázovým systémem libovolné třetí strany. Vyvinutý systém je testován na teoretickém příkladu a v neposlední řadě je věnována pozornost také zabezpečení aplikace. Na závěr jsou zmíněna možná vylepšení a rozšíření při dalším vývoji aplikace.

Klíčová slova: ekonomické řízení přepravy zboží, optimalizace tras, rozvoz a svoz zboží, zásobování, úspora dopravních nákladů, software.

ABSTRACT

The aim of this thesis is to develop a brief overview of the most important tools for economic management of goods transportation, which are currently available. Based on the review is conducted selection of a suitable mapping system including interface, which allows to obtain the data necessary for economic quantification of individual sections of the goods distribution route. In the practical part is created application, which allows the user to get information about revenue from individual segments and dynamically decide on the release of unprofitable ones. It is resolved the interface that is capable of openly communicate with any third party database system. The developed system is tested on a theoretical example and is also addressed security issues. At the end are mentioned possible improvements and enhancements for further application development.

Keywords: economical control of goods transportation, routes optimization, delivery and collection of goods, supply, savings in transport costs, software.

Poděkování

Chtěl bych velmi poděkovat mému bratrovi, Ing. Jiřímu Zamazalovi, jehož několik cenných telefonických konzultací ohledně praktického programování zefektivnily moji práci při vytváření kódu natolik, že funkční aplikace mohla být dokončena včas. Díky brácho, máš to u mě!

Dále bych chtěl poděkovat mému vedoucímu diplomové práce, Ing. Bc. Pavlovi Vařachovi, Ph.D. za vedení mé diplomové práce a za to, že mi umožnil práci právě na tomto tématu.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve

.....

Zlíně

podpis diplomanta

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 PŘEHLED SOFTWAREVÝCH ŘEŠENÍ PRO EKONOMICKÉ PLÁNOVÁNÍ OBCHODNÍCH CEST	12
1.1 DESKTOPOVÉ APLIKACE.....	12
1.1.1 MapInfo Professional & TourSolver.....	12
1.1.2 PLANTOUR.....	13
1.1.3 COLLI.....	13
1.2 ON-LINE WEBOVÉ APLIKACE.....	15
1.2.1 Google Maps	15
1.2.2 Mapy.cz.....	18
1.2.3 OpenLayers.....	20
2 VÝBĚR MAPOVÉHO PODKLADU	23
II PRAKTICKÁ ČÁST	24
3 NÁVRH ŘEŠENÍ APLIKACE	25
3.1 POUŽITÉ TECHNOLOGIE.....	26
3.1.1 Mapy.cz API v4	26
3.1.2 Knihovna JAK	26
3.1.3 Programovací jazyk JavaScript.....	26
3.1.4 Jazyk HTML.....	26
3.1.5 CSS styly.....	27
3.1.6 Framework jQuery.....	27
3.1.7 Jazyk PHP	27
3.1.8 Jazyk SQL	28
3.1.9 HTTP server Apache	28
3.1.10 Databázový systém MySQL.....	28
3.1.11 Databázový systém PostgreSQL.....	28
3.1.12 Jazyk C.....	28
3.2 POUŽITÉ VÝVOJOVÉ NÁSTROJE	29
3.2.1 Vývojové prostředí Microsoft Visual Studio 2010.....	29
3.2.2 Textový editor PSPad 4.5.4	30
3.2.3 Webový prohlížeč Mozilla Firefox 19.0.2 (a vestavěná chybová konzole).....	30
3.3 SYSTÉMOVÉ POŽADAVKY PRO SPUŠTĚNÍ	31
3.3.1 Programový balík EasyPHP 12.1 + PHP 5.4.6 + server Apache.....	31
3.3.2 Programový balík PostgreSQL version 9.1.9 pro Win x86-32.....	32
3.3.3 Knihovna Microsoft .NET Framework 4	32
3.3.4 Webový prohlížeč	32
3.3.5 Internetové připojení.....	32
3.3.6 Rozlišení obrazovky.....	32
3.3.7 Operační systém.....	32
4 VNITŘNÍ STRUKTURA APLIKACE A POPIS JEDNOTLIVÝCH	

SOUČÁSTÍ PROGRAMU	33
4.1 UMÍSTĚNÍ SOUBORŮ APLIKACE A JEJICH STRUKTURA	33
4.2 PROBLÉM SYNCHRONIZACE PŘI NAČÍTÁNÍ A ZPRACOVÁNÍ DAT.....	34
4.3 DATABÁZE.....	35
4.4 UŽIVATELSKÝ INTERFACE.....	39
4.5 GEOKODÉR MAPY.CZ	40
4.5.1 AJAX	40
4.5.2 Jak pracuje Geokodér Mapy.cz	41
4.6 PRÁCE S DATY UVNITŘ APLIKACE	42
4.6.1 Způsob uložení dat.....	42
4.6.2 Redukce množství požadavků na geokodér	44
4.6.3 Předávání dat mezi jednotlivými komponentami aplikace	44
4.7 PRINCIP VÝPOČETNÍHO ALGORITMU A JEHO VLASTNOSTI	45
4.7.1 Charakteristika vstupních dat	45
4.7.2 Zjednodušující podmínky	46
4.7.3 Vlastnosti výpočetního algoritmu	47
4.7.4 Algoritmus č. 1 – permutace	48
4.7.5 Algoritmus č. 2 – kombinace s permutací	49
5 ZABEZPEČENÍ APLIKACE	51
5.1 ÚTOK NA DATABÁZI	51
5.2 ODPOSLECH SÍŤOVÉHO PROVOZU	52
5.3 OVLÁDÁNÍ APLIKACE NEPOVOLANOU OSOBOU	52
6 POPIS OVLÁDÁNÍ APLIKACE	53
6.1 SPUŠTĚNÍ APLIKACE	53
6.2 NAČTENÍ ZÁKAZNÍKŮ Z DATABÁZE	54
6.3 RUČNÍ ZADÁNÍ ZÁKAZNÍKA	55
6.4 SMAZÁNÍ POLOŽKY	56
6.5 VLOŽENÍ WAYPOINTU PŘÍMO DO MAPOVÉHO PODKLADU	56
6.6 DOTAZOVÁNÍ GEOKODÉRU	57
6.7 VÝPOČET A ZOBRAZENÍ OPTIMÁLNÍ TRASY	58
6.8 DALŠÍ MOŽNOSTI OVLÁDÁNÍ APLIKACE	59
7 MODELOVÝ PŘÍPAD POUŽITÍ APLIKACE.....	61
ZÁVĚR	65
ZÁVĚR V ANGLIČTINĚ	CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.
SEZNAM POUŽITÉ LITERATURY.....	67
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	70
SEZNAM OBRÁZKŮ	71

SEZNAM TABULEK	73
SEZNAM PŘÍLOH	74

ÚVOD

Motivace pro používání systémů ekonomického řízení přepravy zboží je především ekonomická. Organizace všech typů a velikostí hledají způsob, jak zlepšovat své výsledky a zároveň snižovat náklady a přispívat k udržitelnému vývoji [1].

Aby se těmto cílům přiblížily, přechází řada z nich na nová softwarová řešení, umožňující optimalizaci jejich výrobních a obchodních procesů. Společnosti, které tyto možnosti využívají, zaznamenávají jednoznačné zlepšení svých výsledků, mimo jiné zvýšení produktivity, celkovou úsporu nákladů a zmírnění dopadu své činnosti na životní prostředí.

Organizace implementují technologie pro optimalizaci ekonomického řízení přepravy zboží v distribuční síti proto, že významně ovlivňují jejich výsledky. Většina společností se pro tyto systémy rozhoduje kvůli očekávanému zvýšení produktivity a snížení nákladů.

Hlavní výhody použití systémů pro optimalizaci rozvážkových tras:

- úspora času a finančních prostředků, které by musely být vynaloženy na pohonné hmoty a mzdu zaměstnance v důsledku projíždění neefektivních tras.
- dopravce je schopen s každým vozidlem absolvovat mnohem více zakázek, než by bylo možné při neefektivním plánování tras a tím pádem je schopen dosáhnout většího zisku za stejnou jednotku času.
- snížení nákladů na údržbu vozového parku v důsledku nižšího opotřebení vozidel.
- snižování produkce CO₂ omezením neefektivního objíždění zákazníků.

I. TEORETICKÁ ČÁST

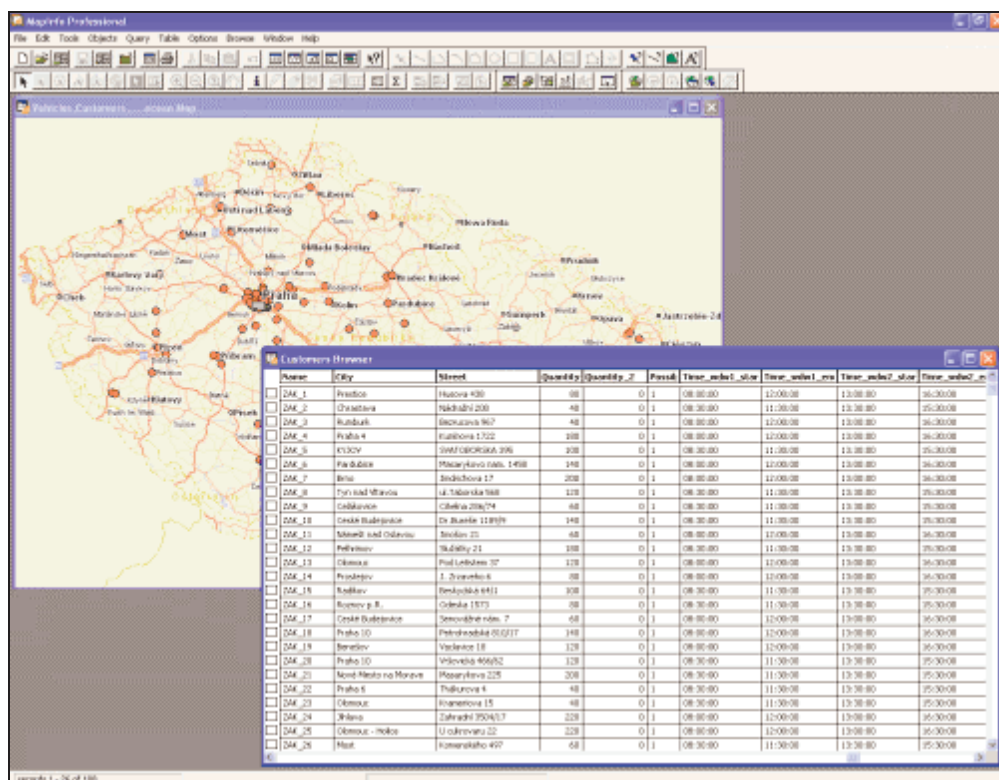
1 PŘEHLED SOFTWAREVÝCH ŘEŠENÍ PRO EKONOMICKÉ PLÁNOVÁNÍ OBCHODNÍCH CEST

Nabídka již hotového softwaru určeného pro ekonomické plánování obchodních cest je na trhu značně omezená. Jako zástupce desktopových aplikací lze uvést následující dva produkty.

1.1 Desktopové aplikace

1.1.1 MapInfo Professional & TourSolver

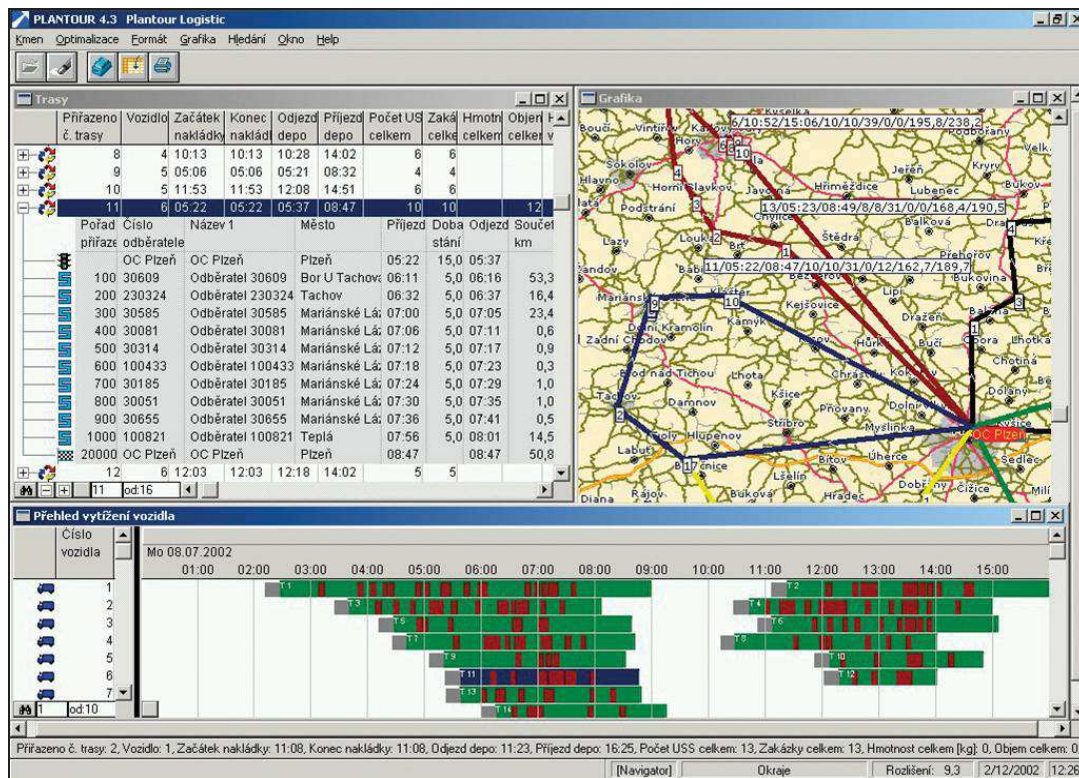
Software od firmy CMap, s.r.o. [2] nabízí optimalizaci rozvozu nebo svozu zboží, jako například rozvoz zásilek, tisku, jídla pro školy, farmaceutické a drogistické produkty atd. Aplikace umožňuje optimalizovat dopravu na základě celé řady faktorů, jako je množství zboží, pracovní dny, otevírací hodiny, náklady na kilometr, fixní náklady, fixní a variabilní doba vykládky a spousta dalších.



Obr. 1. Okno programu TourSolver

1.1.2 PLANTOUR

Řešení od firmy LOGICON Partner s.r.o. [3] umožňuje navrhovat optimální trasy pro závoz dodacích míst včetně zohlednění zpětných svozů. Trasy jsou navrhovány na základě aktuálních objednávek a vozového parku dynamicky tak, aby byly optimální z hlediska nákladů a zároveň splňovaly všechny zadané restriktce (doba závozu, limit vytížení vozidla, požadavky na vybavenost vozidla aj.).



Obr. 2. Okno programu PLANTOUR

Oba tyto produkty vypadají velmi profesionálně zpracované a nabízející velké množství funkcí. Cena softwarového balíku MapInfo Professional & TourSolver firmy CSmap, s.r.o. se pohybuje kolem 100 000,- Kč. Cenu programu PLANTOUR firmy LOGICON Partner s.r.o. se nepodařilo zjistit.

1.1.3 COLLI

Informační systém pro spedici a sběrnou službu od firmy CID International [36] se sídlem v Olomouci je vlastně celý balík produktů, které kompletně řeší logistiku od kontaktu zákazníka a zpracování objednávky, příjmu a výdeje zboží do/ze skladu, jeho evidenci a

uložení do databáze až po plánování tras a sledování vozidel. Hlavními součástmi produktu COLLI jsou nastavby řešící:

- Mobilní terminály.
- Webové nastavby.
- Datové sklady.
- Optimalizace a plánování tras.
- Skenovací technika.
- GPS sledování pohybu vozidel.

Zajímavostí je, že právě funkce optimalizace a plánování tras je zajišťována online subdodavatelsky firmou DIGITECH, která má tuto optimalizaci postavenou na systému PLANTOUR, který je popsán výše.



Obr. 3. Spolupráce systému COLLI se systémem PLANTOUR

1.2 On-line webové aplikace

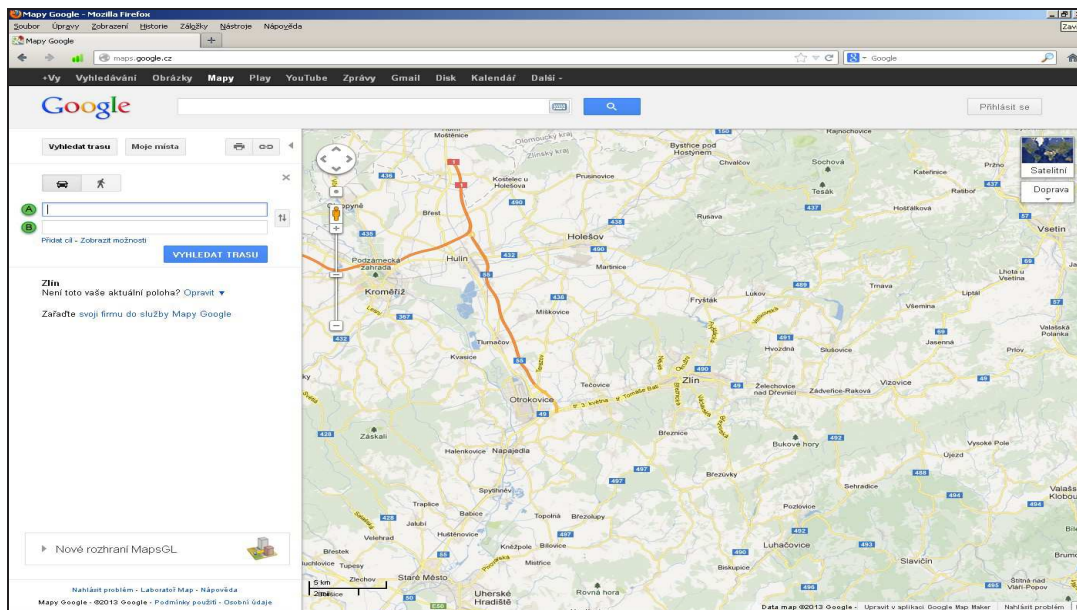
V podstatě neexistuje varianta jako on-line webová aplikace umožňující ekonomické plánování rozvážky zboží (alespoň se mi ji nepodařilo nalézt). Z hlediska ekonomičnosti cest se on-line webové aplikace omezují pouze na výpočet spotřeby pohonných hmot vozidla.

Aplikace obsahují jádro starající se o obsluhu požadavků uživatelů (tzv. „geokodér“), například požadavek na zvětšení mapy, požadavek na vyhledání a zobrazení lokality, atd. Většina z těchto aplikací umí na základě zadání startovního a cílového bodu na mapě vypočítat a zobrazit trasu pro vozidlo pohybující se po silnici (tzv. „routing“). A některé z těchto aplikací též poskytují příkazové rozhraní (API) a za určitých podmínek umožňují implementovat a používat mapových služeb ve svém vlastním softwarovém řešení.

Následuje několik aplikací, které splňovaly požadavky na možnost jejich využití při programování vlastního softwaru pro ekonomické plánování rozvážky zboží. Požadavky byly, že geokodér musel umět funkci routing a funkce k tomu přidružené (výpočet vzdálenosti a času této trasy, zobrazování mapového podkladu a základní práce s ním), dále muselo být k dispozici programové rozhraní API a v poslední řadě tyto funkce musely být poskytovány (alespoň v nějaké omezené míře) zdarma.

1.2.1 Google Maps

Google Maps [4] společnosti Google nabízí vlastní mapovou službu, která poskytuje mnoho možností. Kromě těch standardních navíc můžeme zmínit funkci Street View, která umožňuje prohlížení různých míst celého světa prostřednictvím panoramatických snímků.



Obr. 4. Webová aplikace Google Maps

Vlastnosti programového rozhraní Google Maps API:

- Technická omezení – API spolupracuje s programovacím jazykem JavaScript.
- Mapový podklad – je k dispozici celý svět a mapový podklad je velmi podrobný.
- Poskytované verze API – v současné době je k dispozici API ve verzi 3.11. [5] Předchozí verze tohoto API (API verze 2 , Maps API for Flash, Local Search) jsou zastaralé a dále nepodporované.
- Licenční podmínky (freewarová omezení) – nejzásadnější je pravděpodobně omezení počtu požadavků na geokódování za den. Úplný seznam omezení Google Maps API [6] pro bezplatné užívání služby je uveden na obrázcích níže.
- Technická dokumentace – dokumentace je obsáhlá a také je kvalitně zpracovaná. Vzhledem k rozšířenosti a oblíbenosti služby není problém s uživatelskou podporou, například ve formě rady na různých fórech atd. K dispozici je též řada tutorialů a vzorových příkladů implementace Google Maps API.
- Budoucí stav – Google Maps API je soustavně vyvíjeno a zdokonalováno společností Google, průběžně jsou vydávány nové verze a doplňována technická dokumentace.

Features	Maps API
Street View	✓
Geocoding Web Service	2500 requests per day
Directions Web Service	2500 requests per day with 10 waypoints per request
Distance Matrix Web Service	100 elements per query 100 elements per 10 seconds 2500 elements per day
Elevation Web Service	2500 requests per day with 25 000 samples per day
Static Maps API maximum resolution	640 x 640
Static Maps API maximum scale	2X
Street View Image API maximum resolution	640 x 640
Analytics	
Demographics Layer	

Obr. 5. Omezení freewarové služby Google Maps API

Support	Maps API
Google Maps API Developer resources	✓
Service Level Agreement	
Technical Support	
Support portal & usage reporting	

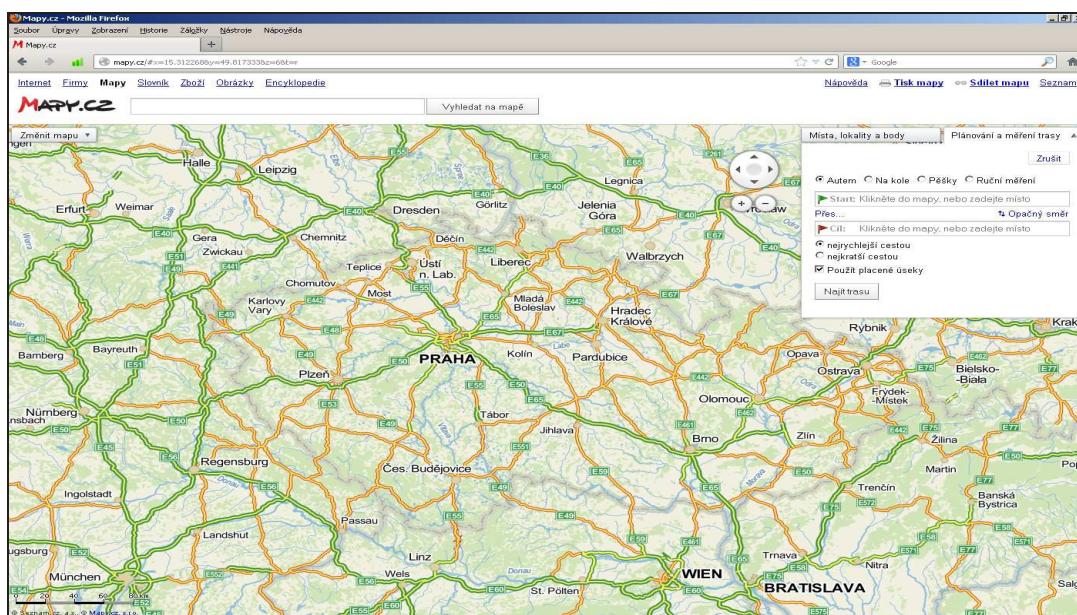
Obr. 6. Omezení freewarové služby Google Maps API

Use cases	Maps API
Free & publicly available	✓
Internal deployments	
Embedding in software and applications for fee	
Reselling services with Google Maps	
Control of advertising	
Private asset tracking	

Obr. 7. Omezení freewarové služby Google Maps API

1.2.2 Mapy.cz

Mapy.cz [7] provozované společností Seznam.cz, a.s. provozuje vlastní mapovou službu. Nabízené funkce této služby jsou sice pouze standardní, ale dostačující. Mapový podklad je dokonce pro ČR republiku kvalitnější (podrobnější), než u Google Maps.



Obr. 8. Webová aplikace Mapy.cz

Vlastnosti programového rozhraní Mapy.cz API:

- Technická omezení – API spolupracuje s programovacím jazykem JavaScript a JavaScriptovým frameworkem JAK.
- Mapový podklad – je k dispozici celá Evropa a část Ruska. Dále jsou částečné přesahy na Indický poloostrov a sever Afriky. Mapový podklad pro ČR je velmi podrobný. Velká nevýhoda je ovšem to, že mimo ČR je pro API omezeno zvětšení mapového podkladu tak, že nejsou patrné ulice (podrobnosti končí na úrovni zvětšení „města“, kde jsou patrné pouze obrysy polygonů celých zastavěných aglomerací a hlavní tahy městem).
- Poskytované verze API – v současné době je k dispozici API ve verzi 4.8. [8] Předchozí verze tohoto API (verze 3) je zastaralá a její podpora bude ukončena. Dále ještě existuje verze Simple API, která má ovšem značně omezené funkce (dovoluje pouze vložit do stránky malou mapku se značkou, načítá se rychleji a není interaktivní) a pro vývoj softwaru je nepoužitelná.
- Licenční podmínky (freewarová omezení) – použití API je zcela zdarma a to s jistým omezením i pro komerční účely. Podstatné je také to, že denní počet požadavků na geokodér není nijak omezen.
- Technická dokumentace – dokumentace je sice kompletně k dispozici, nicméně je velmi strohá a balancuje na hraně použitelnosti. Přímo v dokumentaci zoufale chybí krátké a jednoduché příklady použití jednotlivých příkazů. Také hypertextové odkazy mezi příkazy a typy jejich možných parametrů ve většině případů chybí. Obecně se s touto dokumentací pracuje velmi špatně. Tuto špatnou dokumentaci ovšem Mapy.cz poměrně úspěšně vyvažují velmi pěknou sbírkou řešených ukázek implementace s komentáři, pokrývající většinu nabízených funkcionalit. V případě nutnosti je k dispozici fórum v českém jazyce [15], na kterém vývojáři tohoto API dvakrát denně odpovídají na dotazy.
- Budoucí stav – Mapy.cz API je soustavně vyvíjeno a zdokonalováno společností Seznam.cz, a.s., průběžně jsou vydávány nové verze (jen mezi lednem a březnem 2013 byly vydány další dvě nové verze) a je také doplňována technická dokumentace.

1.2.3 OpenLayers

OpenLayers [9] je provozován organizací Open Source Geospatial Foundation. Open Source Geospatial Foundation je nezisková organizace spolupracující a zastřešující komunitu lidí, kteří tento open-source software zdarma a ve svém volném čase vyvíjejí. OpenLayers jsou postaveny na mapovém podkladu OpenStreetMap [10], o který se stará mezinárodní nezisková organizace OpenStreetMap Foundation a který je vytvářen a doplňován každým uživatelem, který o to projeví zájem a který se zaregistruje, nebo který je ochoten poskytnout mapové podklady, které má ve svém vlastnictví (například firmy, obce). Mapové podklady OpenStreetMap jsou šířeny pod Open Data Commons Open Database licencí, což znamená, že je lze používat naprosto k čemukoliv a jakkoliv s jedinou podmínkou, a to že bude uváděno autorství OpenStreetMap.

The screenshot shows the OpenLayers website homepage. The browser title is "OpenLayers: Home - Mozilla Firefox". The page content includes:

- OpenLayers: Free Maps for the Web**
- Get OpenLayers Now!**
 - 2.12 (Stable): tar.gz | zip
 - 2.12 Release Notes
 - API Documentation, User documentation
 - See examples of OpenLayers Usage: Release Examples (2.12), Development Examples
 - Fork us on GitHub
- About OpenLayers**

OpenLayers makes it easy to put a dynamic map in any web page. It can display map tiles and markers loaded from any source. OpenLayers has been developed to further the use of geographic information of all kinds. OpenLayers is completely free, Open Source JavaScript, released under the 2-clause BSD License (also known as the FreeBSD).
- Toward OpenLayers 3!**

We've begun the development effort to make the next major version of OpenLayers a reality. OpenLayers 3 is a comprehensive rewrite of the library, targeting the latest in HTML5 and CSS3 features. The library will continue to have broad support for projections, standard protocols, and editing functionality from OpenLayers 2.x. The new version of the library will focus on performance improvements, lighter builds, prettier visual components, an improved API, and more. Some of the major highlights are:

 - WebGL** promises to bring 3D capabilities and increased performance for all mapping needs to the latest browsers. OpenLayers 3.0 will offer WebGL, while degrading nicely in less capable browsers.
 - Cesium**: The OpenLayers community will also integrate the new Cesium library to enable full 3D spinning globe capabilities directly into the 3.0 release.
 - Closure Compiler**: By utilizing the Closure Compiler, applications developers will be able to create smaller and faster libraries, easing the use of the extensive OpenLayers 3.0 toolkit.
 - A new codebase**: This offers an opportunity to clean up some of the "clunky" ways of doing things in OpenLayers. The team will also create with new API designs, which will be more accessible to all.
 - High-quality documentation**: The new release will also feature documentation with fresh examples and default designs in OpenLayers 3.0. Making a toolkit standout is about more than the actual code.

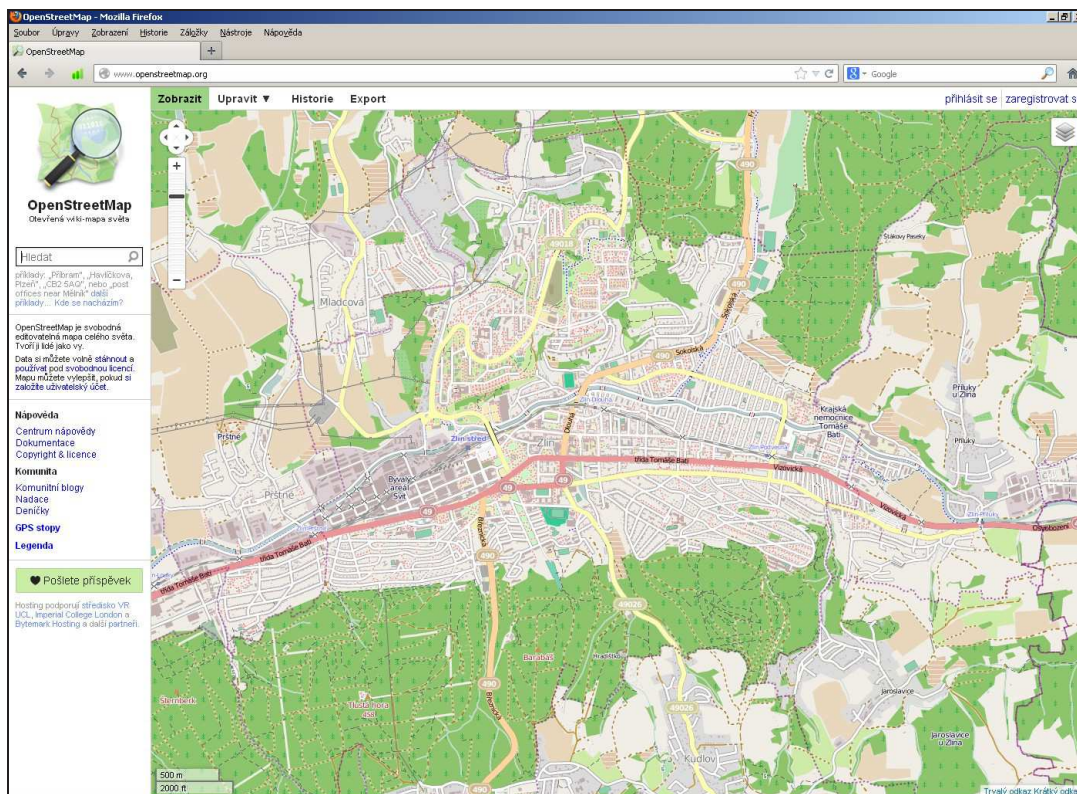
If you've used OpenLayers in the past, if you're using it in production deployments, or if you may use it in the future, please consider contributing to our collaborative funding effort. We are aiming to raise \$350,000, to fully fund OpenLayers 3.0 in one shot. Contributing right now is the most effective time to help OpenLayers. Kickstarter-style campaign is running at <http://www.indiegogo.com/vol3>, read our blog post to know more about it. Any amount will help but if you are able to give \$5,000 or more we're offering a number of different "perks" to help make it easier to contribute:

- \$5,000: Your organization's logo will appear on the OpenLayers 3.0 Thank You page.
- \$25,000: A 1 year OpenLayers support package with two named users.
- \$40,000: Logo and link will appear on the OpenLayers homepage as a "Sustaining Sponsor", as well as at the top of the OpenLayers 3.0 Thank You page.
- \$75,000: A 1 year OpenLayers enterprise support package, with up to 8 named users, and "Sustaining Sponsor" logos and credits.

These packages will be delivered by companies employing OpenLayers committers. For more information please email ol3-funding@googlegroups.com.

On the right side of the page, there is a world map widget with the text "Put an open map widget in any web page!". Below the map is a "kUSD" gauge showing a value of 336. At the bottom right, there is a section titled "Books about OpenLayers" featuring the "OpenLayers Cookbook" by Antonio Santiago Perez, available in print and eBook formats from Packt Publishing.

Obr. 9. Webová aplikace OpenLayers



Obr. 10. Webová aplikace OpenStreetMap

- Technická omezení – API spolupracuje s programovacím jazykem JavaScript.
- Mapový podklad – je k dispozici celý svět a mapový podklad je velmi podrobný. V některých lokalitách se však jeho kvalita může různit.
- Poskytované verze API – API není verzované, technická dokumentace [11] je průběžně doplňována. Jako hlavní dokumentace slouží sada vzorových příkladů použití [12]. Základní API neumí funkce geokodéru, tyto funkce (například routování) musejí být zajištěny řešením třetích stran (například OpenRouteService [13] nebo pgRouting [14]).
- Licenční podmínky (freewareová omezení) – kromě požadavku na uvádění autorství OpenStreetMap nejsou žádná omezení.
- Technická dokumentace – dokumentace je poměrně obsáhlá. K dispozici je též řada tutorialů a vzorových příkladů implementace. Nicméně je nepříjemné, že základní dokumentace neobsahuje příkazy pro funkci routování. Ty je nutno hledat i s příklady použití na stránkách třetích stran.

- Budoucí stav – Projekt OpenLayers je nadále vyvíjen komunitou, v budoucnu se dá očekávat, že technická dokumentace bude postupně doplňována a funkcionality rošiřovány.

2 VÝBĚR MAPOVÉHO PODKLADU

Při výběru technologie, na které postavit aplikaci pro ekonomického řízení přepravy zboží v široké distribuční síti jsem se rozhodoval mezi Google Maps a Mapy.cz. Nakonec jsem dal přednost Mapy.cz a to z následujících důvodů:

- Mapy.cz nemají narozdíl od Google Maps žádné omezení, co se týče denního počtu požadavků na geokódování. Toto by mohl být problém, jak při vývoji samotné aplikace (mnoho opakujících se požadavků při testování), tak při samotném používání aplikace ve firmě, protože překročení denního limitu by aplikaci znefunkčnilo a na zprovoznění by bylo nutno čekat do druhého dne.
- Mapy.cz i přes poměrně katastrofální stav technické dokumentace mají velmi pěkně řešené vzorové příklady, které kompletně pokrývají funkcionality potřebné pro chod mojí aplikace.
- Mapy.cz provozují diskuzní fórum, na kterém je možné se v případě nouze na cokoliv dotázat vývojářů, kteří na API pracují a to v českém jazyce.

II. PRAKTICKÁ ČÁST

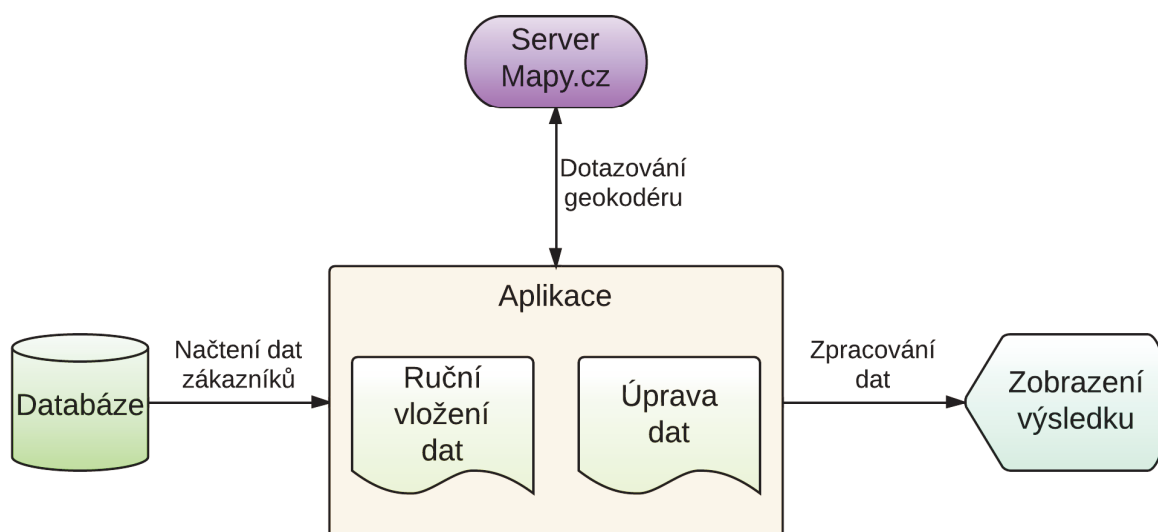
3 NÁVRH ŘEŠENÍ APLIKACE

Aplikace pro ekonomického řízení přepravy zboží v široké distribuční síti (dále jen „aplikace“) načte údaje o požadovaných bodech průjezdu. Body průjezdu jsou reprezentovány adresou (ulice, číslo popisné a město). Dále je u každého bodu průjezdu požadováno též zadání jména zákazníka a částky, kterou je možné zákazníkovi fakturovat v případě realizace obchodu v tomto bodě průjezdu (na této adrese). V účetní terminologii se tato částka nazývá „tržba“.

Jména zákazníků, adresy a částky mohou být načítány z libovolného typu databáze přes univerzální rozhraní, nebo mohou být v aplikaci zadávány ručně. Data načtená z databáze, nebo zadaná ručně mohou být libovolně editována, doplňována a mazána.

Až jsou body průjezdu zadány, je možné stisknutím příslušného tlačítka spustit algoritmem řízené dotazování na geokodér serveru Mapy.cz a jsou získány vzdálenosti úseků mezi jednotlivými průjezdními body. Tyto vzdálenosti, které jsou nezbytné pro výpočet optimální trasy, jsou postupně skládány do dotazu, který je po skončení komunikace s geokodérem předán výpočetnímu algoritmu.

Výpočetní algoritmus na základě dodaných údajů spočítá optimální trasu a výsledek předá zpět do aplikace. Ta výsledky načte a uživateli je zobrazí v textové i v grafické podobě. Výsledky je možné opět upravit (něco doplnit nebo smazat) a může být znovu spuštěn výpočet hledání optimální trasy nad těmito upravenými daty.



Obr. 11. Schema obecného fungování aplikace

Celá aplikace běží v okně Internetového prohlížeče a chová se jako webová stránka.

3.1 Použité technologie

3.1.1 Mapy.cz API v4

Toto programové rozhraní umožňuje použití většiny poskytovaných funkcí, které jsou k vidění na stránkách Mapy.cz. API [8] je navrženo pro spolupráci s programovacím jazykem JavaScript a je složeno z mnoha tříd, starajících se o jednotlivé funkcionality.

3.1.2 Knihovna JAK

Knihovna JAK [16] je neoddělitelná komponenta Mapy.cz API a zajišťuje některé funkce tohoto API. Tato knihovna je kompaktní a jednoduchý objektově orientovaný framework, usnadňující práci v prostředí jazyka JavaScript. Je vyvíjen společností Seznam.cz, a.s. a je šířen jako Open Source.

3.1.3 Programovací jazyk JavaScript

JavaScript [17] je multiplatformní, objektově orientovaný skriptovací jazyk. Původně byl vyvinut jako doplněk k jazyku HTML. Program v JavaScriptu je spouštěn až po stažení webové stránky. Jedná se o interpretovaný jazyk, tzn. že program se před spuštěním nekompile a nepřevádí do strojového kódu, ale je interpretován za chodu, nejčastěji webovým prohlížečem, ve kterém se stránka s JavaScriptem načítá.

Vzhledem k nutnosti použití JavaScriptu pro komunikaci s API jsem se rozhodl v jazyce JavaScript naprogramovat též většinu mojí aplikace. Vývoj je tímto zjednodušen, protože odpadá nutnost použití dalšího programovacího jazyka a vytvoření rozhraní mezi tímto jazykem a JavaScriptem.

3.1.4 Jazyk HTML

Jazyk HTML [18] je značkový jazyk pro hypertext. Jsou v něm vytvářeny webové stránky, které umožňují vystavení dokumentů na Internetu. HTML zpravidla funguje ve spolupráci s JavaScriptem. Zatímco HTML zaobaluje vzhled stránky, JavaScript má na

starosti funkcionality této stránky. V mojí aplikaci je použita HTML verze 5, která přináší některé změny (například nové HTML značky, zkrácené zápisy těchto značek atd.).

Z HTML byly pro mojí aplikaci použity především tlačítka (button), formulářové prvky (input), formátovaný text (span), logické bloky textu (div) a tabulky (table) pro uspořádání prvků na stránce.

3.1.5 CSS styly

CSS styly je novější způsob formátování dokumentů. Usnadňují a zrychlují formátování webových stránek a velmi dobře spolupracují s JavaScriptem.

V mojí aplikaci jsou použity u progressbaru, který se zobrazuje při dotazování serveru na vzdálenosti. Dále je jich využito pro nastavení některých vlastností při vytváření objektu markeru knihovnou JAK.

3.1.6 Framework jQuery

jQuery [19] je JavaScriptová knihovna, která usnadňuje práci s JavaScriptem. Zapouzdřuje nejčastěji užívané funkce tak, že jejich použití je rychlé a jednoduché. Řeší také interakci mezi HTML a JavaScriptem a v neposlední řadě je multiplatformní a je šířena jako svobodný a otevřený software pod licencí MIT.

V mojí aplikaci jsem z této knihovny hojně využil tzv. „Selektory“, což jsou nástroje pro výběr DOM (HTML prvku) podle unikátního ID.

3.1.7 Jazyk PHP

Jazyk PHP [20] skriptovací programovací jazyk, jehož skripty jsou vykonávány nikoliv v prohlížeči, ale na straně serveru. K uživateli je přenesen až výsledek po vykonání tohoto PHP skriptu. S PHP je možné ukládat a měnit obsah webových stránek. Je šířen jako Open Source. Pro fungování musí být k dispozici server, který příkazy jazyka PHP bude vykonávat.

V mojí aplikaci je PHP použito pro komunikaci s databází a dále pro komunikaci s algoritmem pro výpočet optimální trasy (předání vstupních dat a vyzvednutí výsledku).

3.1.8 Jazyk SQL

SQL [21] je dotazovací jazyk, který se používá pro práci s daty v relačních databázích. Umožňuje data v databázi vkládat, vypisovat, upravovat a mazat. SQL je tzv. deklarativní programovací jazyk (říká jen "co" se má provést) a musí být vkládán do jiného programovacího jazyka, který je procedurální (přesně specifikuje "jak" se to má provést).

V mojí aplikaci je SQL použito pro získání vstupních dat z databáze a SQL dotazy jsou vloženy do procedurálního jazyka PHP.

3.1.9 HTTP server Apache

HTTP server Apache [30] je program, který zajišťuje obsluhu dotazů webového prohlížeče a posílá jim webové stránky. Je k dispozici pro obě hlavní platformy (Windows, Linux) a je vyvíjen jako Open Source.

3.1.10 Databázový systém MySQL

MySQL [23] je relační databáze typu DBMS a vychází z deklarativního programovacího jazyka SQL. Mezi výhody MySQL patří, že je malý, rychlý, jednoduchý a je šířen jako Open Source, což ho řadí mezi nejoblíbenější databázové systémy. Nejčastěji se MySQL používá ve spojení s jazykem PHP, který umožňuje přístup k datům.

3.1.11 Databázový systém PostgreSQL

PostgreSQL [24] je další databázový systém, který je šířen jako Open Source, díky čemuž bylo možné jej stáhnout a otestovat na něm proceduru připojení k libovolné databázi a stažení požadovaných dat (narozdíl od komerčních produktů, jako je například Oracle).

V mojí aplikaci je PostgreSQL použito pro demonstraci schopností aplikace komunikovat přes univerzální rozhraní s libovolným databázovým řešením třetí strany.

3.1.12 Jazyk C

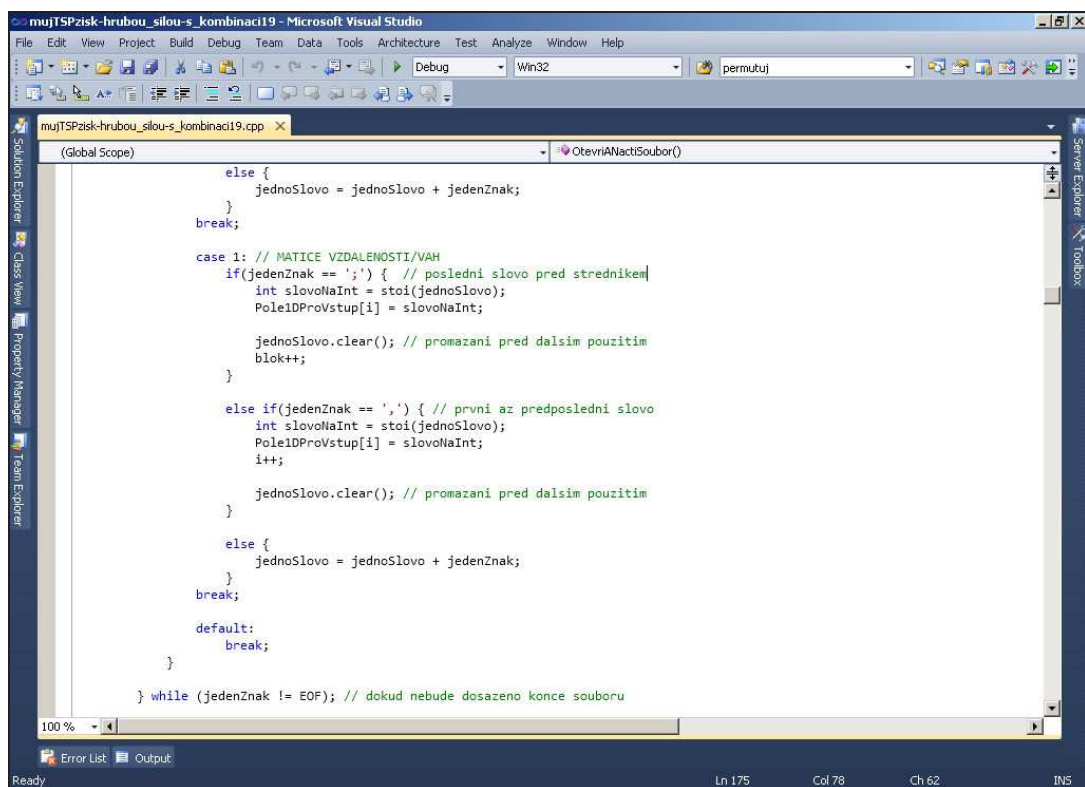
Jazyk C [25] byl vyvinut v 70. letech pro OS Linux. Je to nízkourovňový, kompilovaný programovací jazyk, jehož vykonávání je velmi rychlé (ve srovnání například s JavaScriptem). Je vhodný k implementaci výpočetně náročných úloh. Jeho nevýhodou je, že není objektově orientovaný, což komplikuje tvorbu rozsáhlejších programů.

V mojí aplikaci je jazyk C použit ve výpočetní části, kdy je hledána optimální trasa. Byl zvolen z důvodu jednoduchosti a také proto, že na Internetu je volně k dispozici mnoho vyřešených permutačních a kombinačních algoritmů, které jsou napsány právě v jazyce C.

3.2 Použité vývojové nástroje

3.2.1 Vývojové prostředí Microsoft Visual Studio 2010

Visual Studio 2010 [26] je vývojové prostředí od firmy Microsoft. Lze v něm psát a editovat kódy programů a podporuje celou řadu programovacích jazyků (například jazyk C, C++, C#, F#, Visual Basic, atd.). Dále obsahuje kromě prvků nezbytných pro vývoj programu (editor kódu, zvýraznění syntaxe, debugger, designer) také velmi užitečné nástroje, které programování usnadňují a zrychlují (IntelliSense, refaktorování). Toto vývojové prostředí bylo použito pro psaní a ladění kódu v jazyce C.



```

    else {
        jednoSlovo = jednoSlovo + jedenZnak;
    }
    break;

    case 1: // MATICE VZDALENOSTI/VAH
        if(jedenZnak == ';') { // posledni slovo pred strednikem
            int slovoNaInt = stoi(jednoSlovo);
            Pole1DProVstup[i] = slovoNaInt;

            jednoSlovo.clear(); // promazani pred dalsim pouzitim
            blok++;
        }

        else if(jedenZnak == ',') { // prvni az predposledni slovo
            int slovoNaInt = stoi(jednoSlovo);
            Pole1DProVstup[i] = slovoNaInt;
            i++;

            jednoSlovo.clear(); // promazani pred dalsim pouzitim
        }

        else {
            jednoSlovo = jednoSlovo + jedenZnak;
        }
        break;

    default:
        break;
}

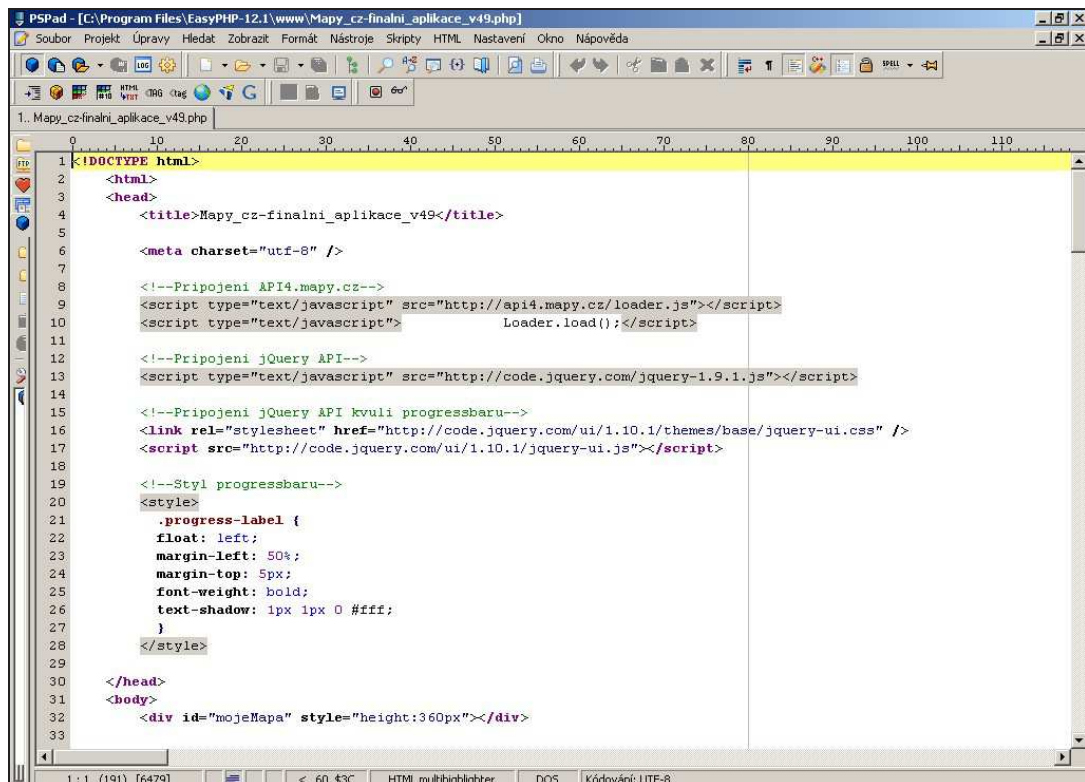
} while (jedenZnak != EOF); // dokud nebude dosazeno konce souboru

```

Obr. 12. Okno programu Microsoft Visual Studio 2010

3.2.2 Textový editor PSPad 4.5.4

PSPad [27] je volně šiřitelný univerzální editor pro OS Microsoft Windows. Mezi jeho výhody patří jednoduchost, nenáročnost na systémové prostředky počítače a fakt, že je zcela zdarma. Dále obsahuje řadu užitečných funkcí jako je zvýraznění syntaxe dle typu editovaného kódu, šablony, HEXA editor atd. Tento editor byl použit pro psaní té části aplikace, která je v jazyce HTML, JavaScript, PHP a SQL.

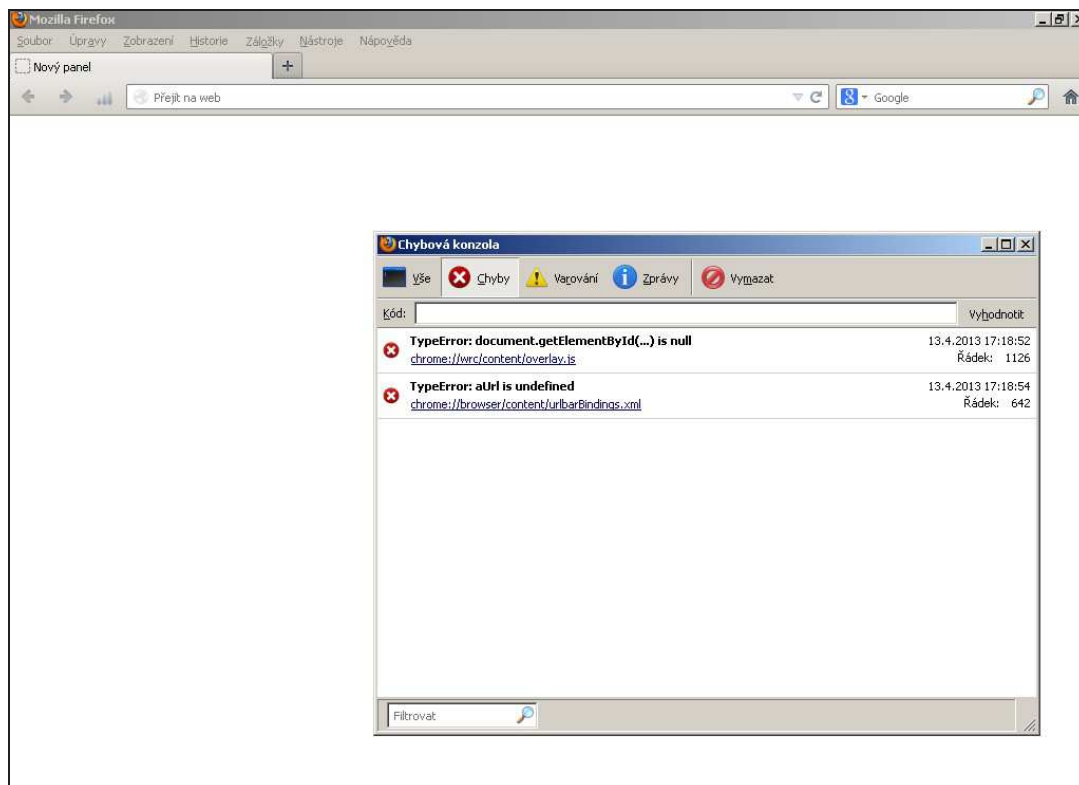


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Mapy_cz-finalni_aplikace_v49</title>
5
6 <meta charset="utf-8" />
7
8 <!--Pripojeni API4.mapy.cz-->
9 <script type="text/javascript" src="http://api4.mapy.cz/loader.js"></script>
10 <script type="text/javascript" src="http://api4.mapy.cz/loader.js">Loader.load();</script>
11
12 <!--Pripojeni jQuery API-->
13 <script type="text/javascript" src="http://code.jquery.com/jquery-1.9.1.js"></script>
14
15 <!--Pripojeni jQuery API kvůli progressbaru-->
16 <link rel="stylesheet" href="http://code.jquery.com/ui/1.10.1/themes/base/jquery-ui.css" />
17 <script src="http://code.jquery.com/ui/1.10.1/jquery-ui.js"></script>
18
19 <!--Styl progressbaru-->
20 <style>
21 .progress-label {
22 float: left;
23 margin-left: 50%;
24 margin-top: 5px;
25 font-weight: bold;
26 text-shadow: 1px 1px 0 #fff;
27 }
28 </style>
29
30 </head>
31 <body>
32 <div id="mojeMapa" style="height:360px"></div>
33
```

Obr. 13. Okno programu PSPad 4.5.4

3.2.3 Webový prohlížeč Mozilla Firefox 19.0.2 (a vestavěná chybová konzole)

Mozilla Firefox [28] je volně šiřitelný multiplatformní webový prohlížeč. K jeho výhodám patří rychlost, bezpečnost a spolehlivost. Obsahuje spoustu užitečných funkcí, jako například blokování vyskakovacích oken nebo obnovu relace po pádu OS. Tento webový prohlížeč byl použit pro zobrazování a testování funkčnosti hotové aplikace.



Obr. 14. Okno webového prohlížeče Mozilla Firefox 19.0.2

3.3 Systémové požadavky pro spuštění

3.3.1 Programový balík EasyPHP 12.1 + PHP 5.4.6 + server Apache

Tento balík programů [22] je určen pro testování stránek. Obsahuje skriptovací jazyk PHP, HTTP server Apache, databázi MySQL s konfiguračním rozhraní PHPMyAdmin a umožňuje spustit PHP skripty na vlastním počítači pro jejich otestování před tím, než jsou nasazeny do reálného provozu a umístěny například na fyzický server.

Přes konfigurační rozhraní PHPMyAdmin je možné si vytvořit vlastní databázi a naplnit ji daty. Po instalaci je nutné restartovat počítač a ověřit si, že server Apache a databáze skutečně běží.

3.3.2 Programový balík PostgreSQL version 9.1.9 pro Win x86-32

Obsahuje relační databázový systém a mimo jiné i rozhraní pgAdmin III, ve kterém je možné databázi vytvořit a spravovat (vytvářet tabulky a vkládat do nich data). Ke stažení [29] je jak verze pro Windows, tak i pro Linux.

3.3.3 Knihovna Microsoft .NET Framework 4

Knihovna je nutná pro běh výpočetní části aplikace, která je psaná v jazyce C, protože výsledný spustitelný exe soubor byl pod tímto frameworkem zkompilován.

3.3.4 Webový prohlížeč

Teoreticky by aplikace měla fungovat pod jakýmkoliv prohlížečem, ale nedoporučuje se použít hodně zastaralé verze, protože by v nich nemusely korektně fungovat některé použité komponenty. Aplikace byla testována v prohlížeči Mozilla Firefox verze 19.0.2.

3.3.5 Internetové připojení

Je potřeba pro veškerou komunikaci se serverem Mapy.cz (zobrazení a práce s mapovým podkladem, dotazování geokodéru, vykreslení trasy).

3.3.6 Rozlišení obrazovky

Aplikace a její ovládací prvky jsou optimalizovány pro rozlišení obrazu 1400 x 1050 bodů a vyšší. Pokud je rozlišení nižší, lze si zvětšení v novějších webových prohlížečích upravit přidržetím klávesy Ctrl a točením kolečka myši tak, aby se ovládací prvky vešly na celou obrazovku.

3.3.7 Operační systém

Aplikace byla vyvíjena a testována pod OS Windows XP Professional + SP3 s osazenými 2 GB operační paměti. Za předpokladu korektní instalace všech komponent popsaných výše by měla bez problémů fungovat i pod jinými verzemi OS Windows firmy Microsoft.

4 VNITŘNÍ STRUKTURA APLIKACE A POPIS JEDNOTLIVÝCH SOUČÁSTÍ PROGRAMU

Aplikace je rozdělena na několik součástí, každá součást řeší specifický úkol. Součásti mezi sebou komunikují a vzájemně si předávají data.

4.1 Umístění souborů aplikace a jejich struktura

Po instalaci programového balíku EasyPHP 12.1 + PHP 5.4.6 + server Apache je třeba najít podadresář WWW, do kterého se musí zkopírovat všechny soubory aplikace. Při ponechání defaultního nastavení instalace se jedná o cestu C:\Program Files\EasyPHP-12.1\www.

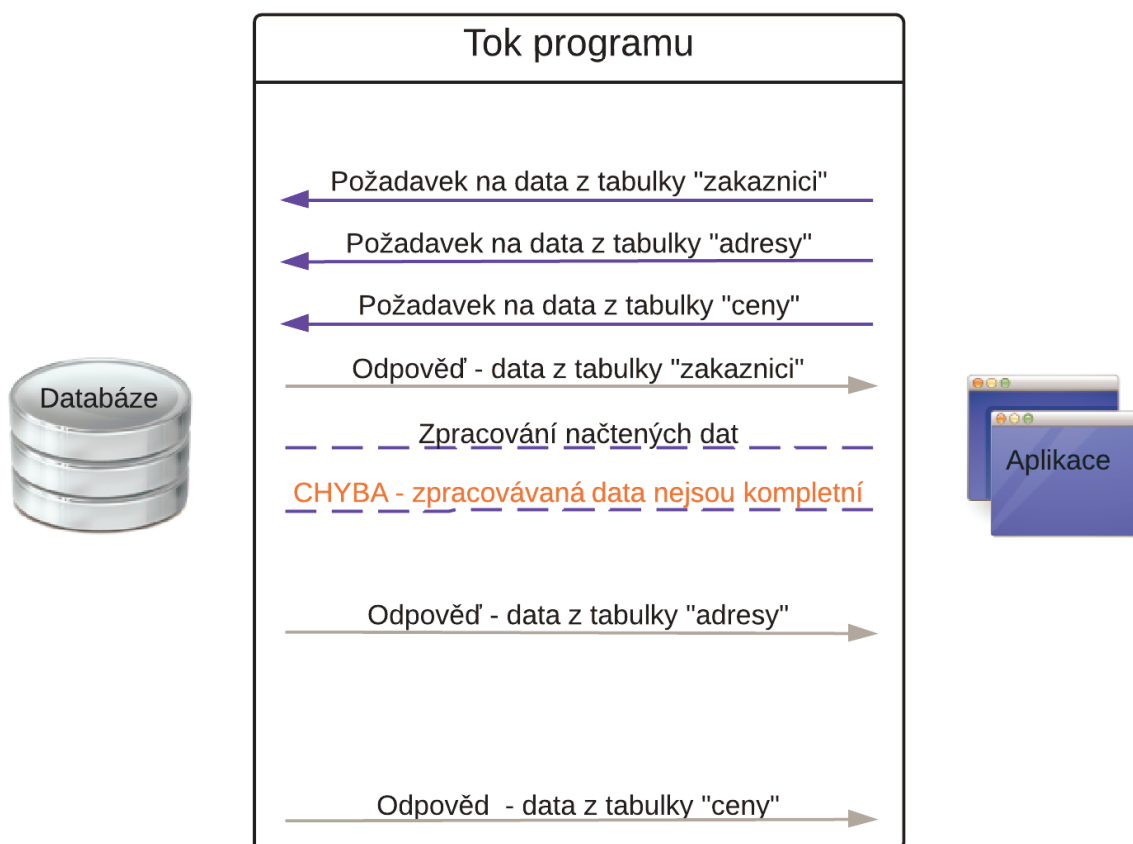
Výčet souborů (a podadresářů) aplikace a stručný popis jejich funkce:

- adresář DB_MYSQL Záloha souborů pro komunikaci s databází MySQL.
- adresář DB_POSTGRESQL Záloha souborů pro komunikaci s databází PostgreSQL.
- soubor aplikace.js JavaScripty, které se připojí k hlavnímu souboru aplikace.
- soubor aplikace.php Hlavní soubor aplikace.
- soubor dotaz_zisk.txt Soubor generovaný aplikací.
- soubor nacti_db_adresy.php Soubor skriptů pro načítání adres z databáze.
- soubor nacti_db_ceny.php Soubor skriptů pro načítání cen z databáze.
- soubor nacti_db_zakaznici.php Soubor skriptů pro načítání zákazníků z databáze.
- soubor odpoved_tsp1_zisk.txt Soubor generovaný výpočetní částí aplikace.
- soubor odpoved_tsp2_zisk.txt Soubor generovaný výpočetní částí aplikace.
- soubor tsp_zisk1.exe Výpočetní část aplikace.
- soubor tsp_zisk2.exe Výpočetní část aplikace.

- soubor volani_tsp_zisk1.php Soubor skriptů řešící předání vstupů do výpočetní části aplikace a načtení výsledků zpět do hlavní aplikace.
- soubor volani_tsp_zisk2.php Soubor skriptů řešící předání vstupů do výpočetní části aplikace a načtení výsledků zpět do hlavní aplikace.

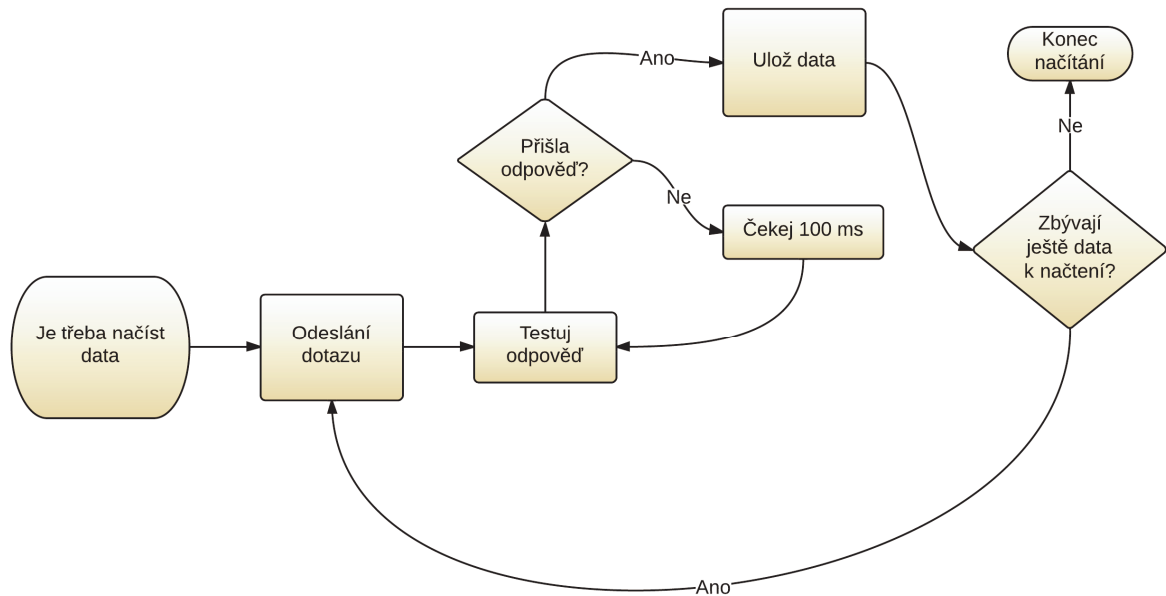
4.2 Problém synchronizace při načítání a zpracování dat

Vzniká při požadavku na jakákoliv data, které jsou mimo hlavní část programu, typicky data v databázi nebo požadavky na geokodér. Problém je v tom, že požadavky na externí servery jsou vyřizovány mnohem pomaleji, než je rychlost vykonávání instrukcí programu běžícím na lokálním počítači. Aplikace nativně nečeká, až je požadavek vyřizen a dorazí odpověď, ale prostě pošle dotaz a pokračuje dále v běhu programu. Pokud se pokusí data zpracovat před tím, než opravdu dorazí, dojde samozřejmě k chybě.



Obr. 15. Problém synchronizace dat při jejich načítání a zpracování

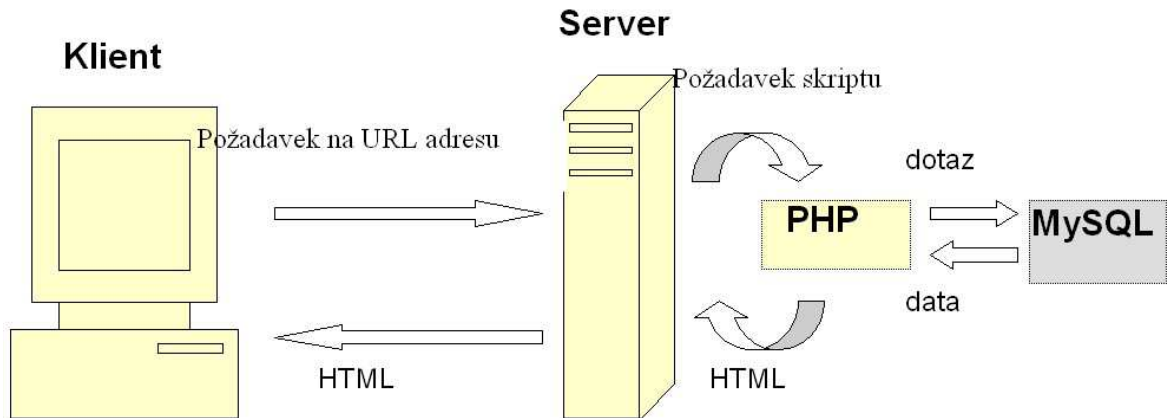
Problém byl vyřešen implementací časovače ve spolupráci s testováním, zda na daný požadavek přišla nebo nepřišla odpověď. Pokud odpověď ještě nedorazila, nastaví se určitý časový interval, po který se bude čekat a po uplynutí tohoto intervalu se testuje znovu. Aplikace odešle další dotaz teprve potom, až obrzší odpověď na dotaz předchozí. Obecně pracuje podle následujícího vývojového diagramu.



Obr. 16. Vývojový diagram testování odpovědi na dotaz

4.3 Databáze

Jako zdroj údajů o zákaznících primárně slouží databáze. Z důvodu nutnosti nákupu licencí u komerčních databází (Oracle, Microsoft SQL, ...) byla použita Open Source databáze MySQL.



Obr. 17. Obecné schéma dotazování databáze MySQL [31]

Data jsou v databázi uložena ve třech tabulkách – zakaznici, adresy, ceny.

id	zakaznik
1	V okno, s.r.o.
2	SVĚT OKEN, s.r.o.
3	ZLINSPOL, spol. s r.o.
4	BROSTAV ZLÍN, s.r.o.
5	NOVÉ OKNO, s.r.o.
6	Okna Macek, a.s.
7	EKO PLAST OKNA, s.r.o.
8	ALFA OKNA s.r.o.
9	M stolárna, s.r.o.

Tab. 1. Příklad tabulky „zakaznici“

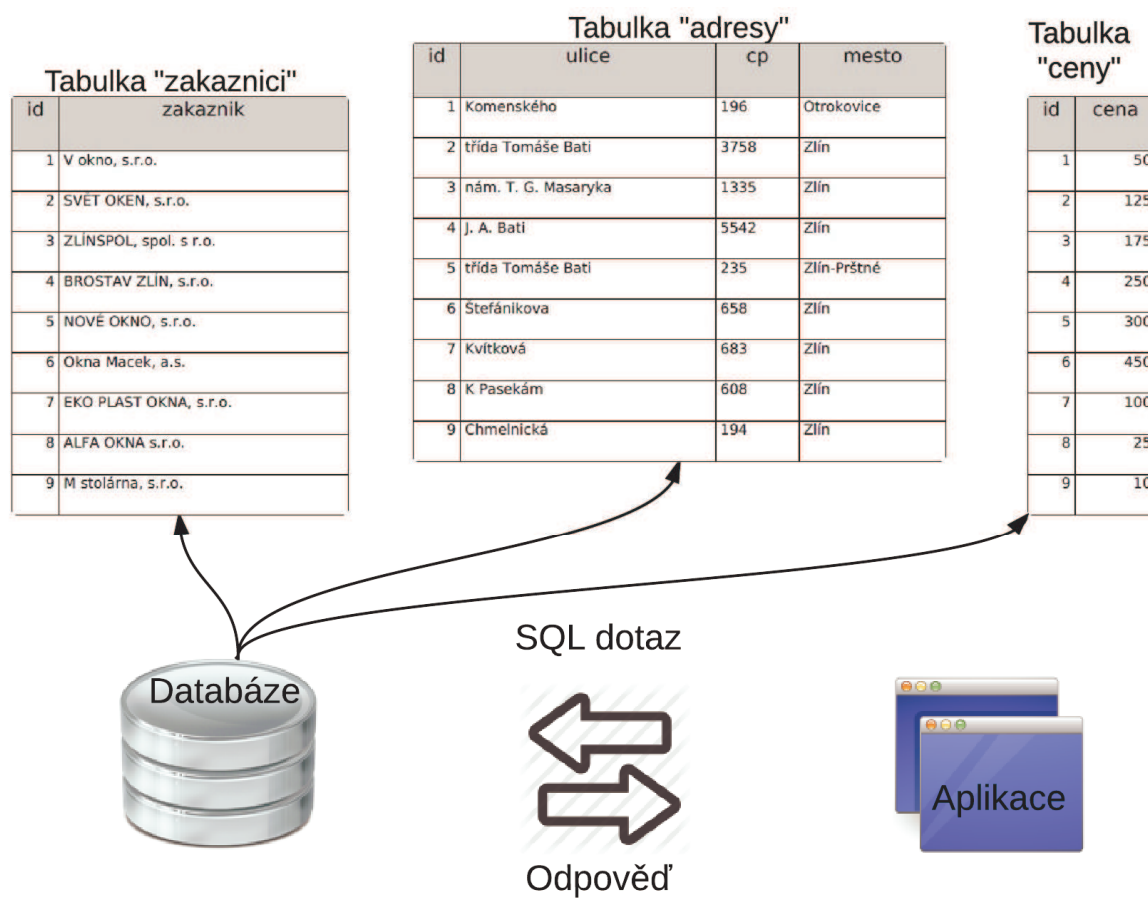
id	ulice	cp	mesto
1	Komenského	196	Otrokovice
2	třída Tomáše Bati	3758	Zlín
3	nám. T. G. Masaryka	1335	Zlín
4	J. A. Bati	5542	Zlín
5	třída Tomáše Bati	235	Zlín-Prštné
6	Štefánikova	658	Zlín
7	Kvítková	683	Zlín
8	K Pasekám	608	Zlín
9	Chmelnická	194	Zlín

Tab. 2. Příklad tabulky „adresy“

id	cena
1	50
2	125
3	175
4	250
5	300
6	450
7	100
8	25
9	10

Tab. 3. Příklad tabulky „ceny“

Data jsou do aplikace předávána jako celé řádky tabulky (bez prvního sloupce „id“), které jsou oddělené středníky. Středník je také za posledním řádkem tabulky. Formát předávaných dat je textový řetězec (datový typ string). Číslo popisné může být v databázi vedeno i jako celé číslo (datový typ integer), při zpracování načítaných dat bude na textový řetězec automaticky převedeno.



V okno, s.r.o.;SVĚT OKEN, s.r.o.;ZLÍNSPOL, spol. s r.o.;BROSTAV ZLÍN, s.r.o.;NOVÉ OKNO, s.r.o.;Okna Macek, a.s.;EKO PLAST OKNA, s.r.o.;ALFA OKNA s.r.o.;M stolárna, s.r.o.;

Komenského 196 Otrokovice;třída Tomáše Bati 3758 Zlín;nám. T. G. Masaryka 1335 Zlín;J. A. Bati 5542 Zlín;třída Tomáše Bati 235 Zlín-Prštné;Štefánikova 658 Zlín;Kvítková 683 Zlín;K Pasekám 608 Zlín;Chmelnická 194 Zlín;

50;125;175;250;300;450;100;25;10;

Obr. 18. Příklad sestavených dat z tabulek databáze předávaných aplikaci

Způsob uložení dat v databázi může být libovolný, například data mohou být uložena jen v jedné tabulce. Potom je třeba vhodným způsobem upravit SQL dotaz tak, aby výstup dat z databáze byl v takovém formátu, jaký aplikace očekává a je nutno zajistit jejich uložení do odpovídajících proměnných v programu.

Načítání dat z databáze řeší skripty v souborech nacti_db_adresy.php, nacti_db_ceny.php a nacti_db_zakaznici.php. Jejich editací lze docílit načítání dat z libovolného druhu databáze a podle libovolného způsobu uložení dat v této databázi.

V podadresářích DB_MYSQL a DB_POSTGRESQL jsou uloženy hotové skripty, které umožní (za předpokladu, že daný databázový server běží a data jsou v něm uložena

způsobem popsaným výše) připojení a načtení dat. Tyto 3 soubory stačí překopírovat (nebo nahradit stávající soubory) do hlavního adresáře a znovu načíst aplikaci v okně webového prohlížeče.

Skripty neřeší kontrolu správnosti dat vložených do databáze, například jestli mají všechny 3 tabulky shodný počet řádků, jestli není některá hodnota chybně vyplněna nebo jestli některá hodnota úplně chybí. Předpokládá se, že správnost dat uložených v databázi zajistí odpovídající správce databáze. V opačném případě načítání dat z databáze a následný výpočet optimální trasy nebude pracovat korektně.

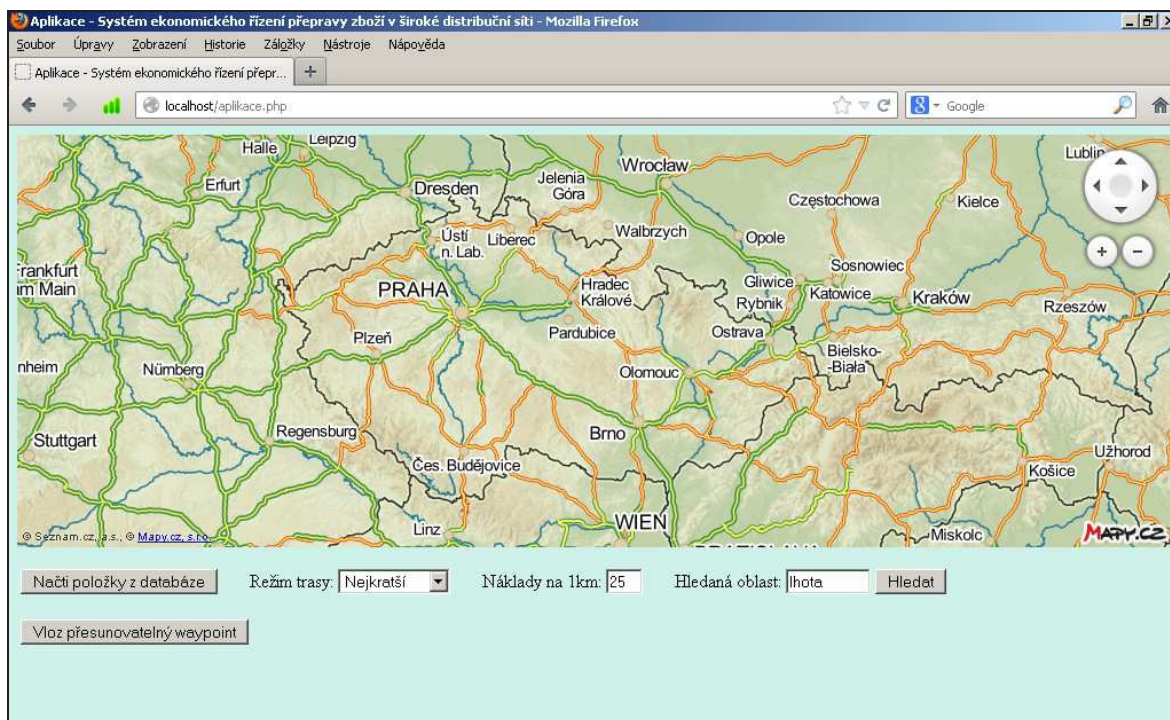
4.4 Uživatelský interface

Celá aplikace běží v okně webového prohlížeče a chová se jako klasická webová stránka. Okno s mapovým podkladem Mapy.cz a veškeré dění v něm (zvětšování, zmenšování, posun mapového podkladu, zobrazování výsledné trasy, ...) obsluhuje programové rozhraní Mapy.cz API v4.

Ovládací prvky pod mapovým podkladem jsou vytvořeny standardními HTML formulářovými prvky, které čekají na uživatelskou interakci. Typická uživatelská interakce je například kliknutí na tlačítko myši. Funkcionality HTML prvků zajišťuje JavaScript, který je na tyto události navázaný.

Aplikace se ovládá myší, do vstupních textových polí jsou data zadávána pomocí klávesnice.

Tlačítka nebo vstupní pole pro zadání dat, která právě nejsou potřeba jsou dynamicky skrývána a opětovně zobrazována tak, aby nedocházelo ke kolizím a nechtěným událostem, například pokud by mělo dojít ke klepnutí na tlačítko vyvolávající funkci výpočtu předtím, než byla vložena data atp.



Obr. 19. Uživatelský interface aplikace

4.5 Geokodér Mapy.cz

4.5.1 AJAX

Způsob dotazování geokodéru je postaven na technologii AJAX. Jedná se o kombinaci několika starších technologií (HTML, JavaScriptu, XML a XMLHttpRequestu) a novém způsobu jejich využití. Hlavní výhodou použití technologie AJAX spočívá v tom, že mění obsah webové stránky bez nutnosti ji celou znova načítat. Server zašle jen ty části stránky, které se v důsledku uživatelské interakce změnila a tyto části se aktualizují a překreslí. AJAX šetří datové přenosy a urychluje uživatelské ovládání a komfort při práci s aplikací. K hlavní nevýhodě patří to, že znefunkčňuje tlačítko „zpět“ ve webovém prohlížeči, respektive, po kliknutí na tlačítko „zpět“ vrátí uživatele na předchozí stránku, ale nevrátí aplikaci do předchozího stavu, což má za následek její další nekorektní chování. Proto v této aplikaci tlačítka „zpět“ a „vpřed“ nepoužívat.

Existují dva způsoby, jak může AJAX přistupovat k datům na externím serveru.

- **Synchronní přístup** - při odeslání žádosti na server se skript zastaví a čeká na odpověď. Teprve po obdržení odpovědi skript pokračuje dále. Webová stránka je po tuto dobu „zamrzlá“ a nelze s ní pracovat, dokud odpověď nedorazí.
- **Asynchronní přístup** - skript odešle žádost na server, zároveň je předána i adresa funkce, která se má vykonat po nalezení a doručení těchto dat a dále pokračuje v činnosti. Výhoda je, že umožní uživateli s webovou stránkou mezitím dále pracovat. Odpověď ze serveru je zpracována až v okamžiku, kdy (nebo jestli vůbec) dorazí.

4.5.2 Jak pracuje Geokodér Mapy.cz

Geokodér Mapy.cz pro vyřizování požadavků používá asynchronní přístup a zároveň při přijetí více dotazů nemusí odpovídat ve stejném pořadí, v jakém mu byly dotazy odeslány. Geokodér nám sice vrátí odpověď, ale my nevíme, ke kterému dotazu tato odpověď patří. Způsobuje to problém, když například odešleme dotaz na vzdálenost mezi konkrétními místy a tuto vzdálenost potřebujeme uložit na specifickou pozici v poli a dále s ní pracovat.

Stejný problém se objevuje i u dotazování doby průjezdu dané trasy a dále i při vykreslování vypočítané optimální trasy do mapového podkladu.

Problém byl vyřešen synchronizací těchto asynchronních požadavků podle stejného schématu, jaký je uveden na obrázku č. 15. Požadavky jsou odesílány sériově, žádný další požadavek není odeslán dříve, než dorazí odpověď na požadavek předcházející.

U některých dotazů může být doba, než přijde odpověď poměrně dlouhá, klidně i více než 10 sekund (typicky dotaz na délku trasy z jednoho konce Evropy na druhý). Je to způsobeno velkým množstvím dat, které je geokodér nucen zpracovat.

Geokodér nefunguje úplně bezchybně, při dotazování mohou nastat následující situace:

- **Odpověď na požadavek nedorazí** – toto lze pozorovat zastavením progressbaru, který se na delší dobu přestane pohybovat. Vzniká zpravidla při dotazu na délku trasy mezi dvěma velmi vzdálenými waypointy. Řešení: spustit dotazování znovu, případně postupným střídáním waypointů a dotazování zjistit se kterou adresou má

geokodér problém a zkusit ji přeformulovat, nebo ji nahradit přesouvateľným waypointem, který bude umístěn někde poblíž původní adresy.

- **Geokodér vrátí neplatnou odpověď** – toto se stává velice ojediněle, při testování bylo pozorováno pouze několikrát. Progressbar sice doběhne do konce, ale jedna z vrácených odpovědí je neplatná (v poli je místo konkrétní hodnoty, třeba čísla, uloženo „NaN“). Při pokusu o výpočet optimální trasy to má za následek chybný výsledek. Proto byl do programu přidán test, zda-li vrácená hodnota není „NaN“ a pokud ano, uživatel je na to upozorněn. Řešení: je stejné jako bod výše.
- **Geokodér nenajde adresu (například ulici), i když určitě existuje (je zobrazena v mapovém podkladu)** – typický problém při hledání adres za hranicemi ČR. Řešení: na danou adresu ručně vložit přesouvateľný waypoint.
- **Geokodér vrací odpovědi velmi pomalu** – toto je zřejmě způsobeno vysokou zátěží serveru při současném velkém množství požadavků od různých uživatelů, případně dotazem na mezi sebou velmi vzdálené waypoints. Mimo špičky však dotazování funguje rychleji. Řešení: žádné, je třeba si počkat.

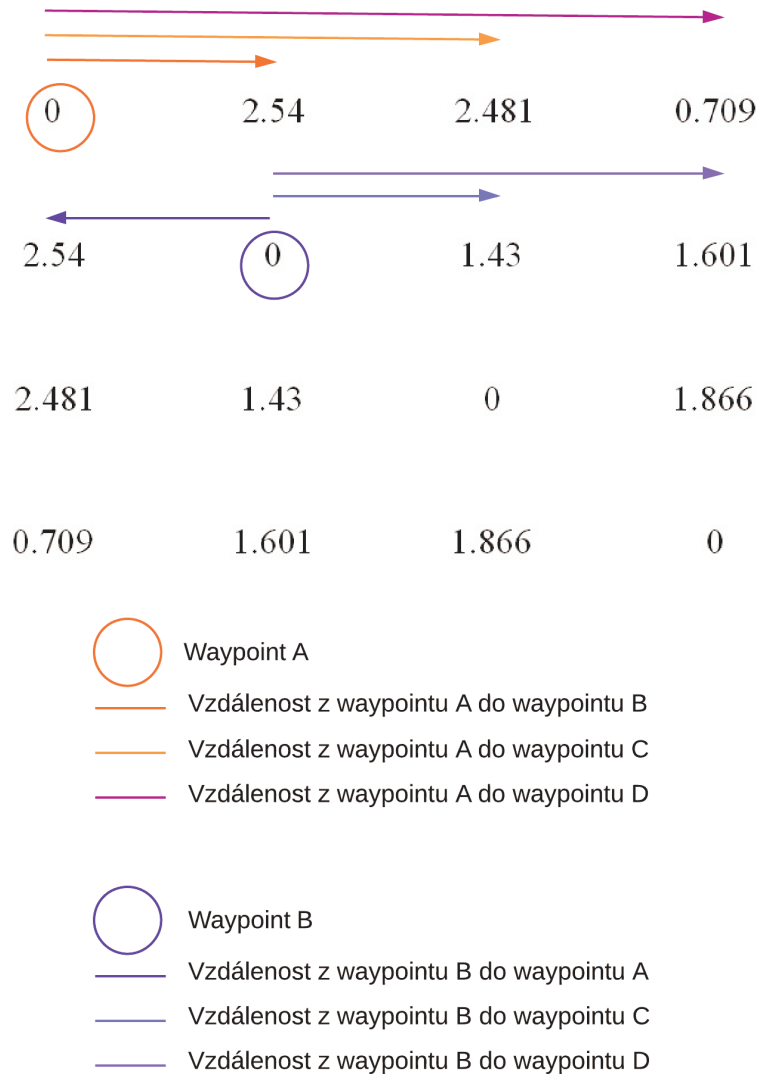
4.6 Práce s daty uvnitř aplikace

4.6.1 Způsob uložení dat

Data, která aplikace sbírá od uživatele, případně načítá z databáze jsou ukládána do jednorozměrných nebo dvourozměrných polí. Do jednorozměrných polí jsou ukládány například jména zákazníků.

Do dvourozměrných polí jsou ukládány odpovědi geokodéru a vlastně se jedná o klasickou matici. Vzniká „mapa“ vazeb mezi jednotlivými waypoints. Vazba je například vzdálenost z waypointu A do waypointu B, cena (náklady) cesty z waypointu A do waypointu B, atd. Tento způsob je nutné použít z toho důvodu, protože při načítání dat zatím nevíme návaznost waypointů (jakým způsobem máme trasu projet) pro dosažení optimální trasy. Proto je do té doby nutné uchovávat informace pro výpočet všech možných způsobů

průjezdu trasy. Optimální trasu se dozvíme až po spuštění výpočetního algoritmu a teprve na základě tohoto výsledku jsou z matic vybrány a sestaveny odpovídající hodnoty. Na následujícím obrázku je uveden příklad načítání načtených vzdáleností mezi waypointy a jejich způsob uložení v matici.



Obr. 20. Příklad složené matice vzdáleností

Analogicky jsou tímto způsobem také uloženy údaje o čase potřebném k průjezdu každého úseku, ceně (tržbě) kterou obdržíme za zakázku realizovanou v daném waypointu a nakonec zisku (tržba mínus náklady na cestu), který je dosažen v daném úseku.

4.6.2 Redukce množství požadavků na geokodér

Podstatnou část doby získání optimální trasy tvoří postupné dotazování geokodéru a získávání vzdáleností mezi jednotlivými waypointy. Proto byla snaha počet těchto dotazů redukovat a dobu dotazování zkrátit. Toho bylo dosaženo eliminací nadbytečných dotazů, protože výsledek dotazu na vzdálenost z waypointu A do waypointu B a z waypointu B do waypointu A je totožný.

Při pozornějším pohledu na obrázek výše si všimneme, že vzdálenosti jsou vůči hlavní diagonále (tvořené nulami) symetrické. Proto stačí geokodér dotazovat jen na úseky nad hlavní diagonálou a zbylé vzdálenosti prostě zrcadlově zkopírovat.

Tímto byl redukován počet dotazů na geokodér o 50% na nezbytné minimum a tedy o polovinu zkrácen dotazovací čas.

4.6.3 Předávání dat mezi jednotlivými komponentami aplikace

Databáze ⇒ aplikace – načítání dat řeší PHP skript, který zajišťuje připojení k databázi a zaobaluje SQL dotaz, kterým jsou data řádek po řádku načítána do textové proměnné. Jednotlivé řádky jsou pro další snadnější zpracování oddělovány středníky. PHP skripty jsou uloženy v samostatných souborech `nacti_db_adresy.php`, `nacti_db_ceny.php` a `nacti_db_zakaznici.php`, pro každou načítanou tabulku z databáze je vytvořen samostatný soubor. Soubory s těmito skripty jsou prováděny postupně, takže pro získání kompletních údajů o zakázce (zákazník, adresa, cena) se musíme k databázi připojit třikrát.

Po dokončení načítání každé tabulky je výstup předán aplikaci jako jeden dlouhý textový řetězec s položkami oddělenými středníky a je dále zpracováván.

Aplikace ⇔ aplikace – po přijetí jsou data v rámci aplikace z textových řetězců od sebe oddělena a jsou převedena na vhodnější formát, buď do jednorozměrného pole nebo do dvourozměrného pole (matice). Tyto operace zajišťuje JavaScript. Při dalším vkládání nebo mazání položek přes uživatelské rozhraní jsou zároveň dynamicky aktualizována data v těchto polích.

Aplikace ⇔ geokodér – data z geokodéru jsou získávána prostřednictvím JavaScriptu ve spolupráci s `Mapy.cz` API, respektive s třídami tohoto API. Odpovědi geokodéru jsou

primárně vracena ve formátu DOM objekt, nicméně API poskytuje třídy pro jejich převod a to buď na číslo (u souřadnic) nebo na textový řetězec (u adres). Dále už jsou data standardně zpracována aplikací a uložena do odpovídajících polí.

Aplikace \Leftrightarrow výpočetní algoritmus – když je seznam waypointů připraven a je kliknuto na tlačítko „Výpočet optimální trasy“, je zahájeno zpracování a odeslání dat výpočetnímu algoritmu. Protože je výpočetní algoritmus postaven jako spustitelný exe soubor na disku, nelze mu tyto data předat přímo pomocí JavaScriptu, protože JavaScriptem není možné přistupovat k jakýmkoliv datům na pevném disku. Tuto operaci nám ale může zajistit jazyk PHP. Data jsou z matic posbírána, převedena na řetězec oddělený středníky a přes HTML formulářový prvek předána do proměnné jazyka PHP. Ten textový řetězec uloží do souboru `dotaz_zisk.txt` a spustí exe soubor odpovídajícího výpočetního algoritmu. Výpočetní algoritmus si načte data ze souboru `dotaz_zisk.txt`, jeho obsah převede na matici a zahájí výpočet. Po dokončení výpočtu je výsledek jako textový řetězec předán přes soubor `odpoved_tsp1_zisk.txt` nebo `odpoved_tsp2_zisk.txt` zpět do aplikace jazykem PHP, který čeká na odpověď.

4.7 Princip výpočetního algoritmu a jeho vlastnosti

4.7.1 Charakteristika vstupních dat

Pro výpočet optimální (nejziskovější) trasy je nutné znát tyto údaje:

- **Náklady na provoz vozidla za 1 ujetý kilometr** – jedná se o součet všech možných nákladů a poplatků spojené s provozem motorového vozidla, jako především náklady na mzdu řidiče, pohonné hmoty, pojištění, leasing, pravidelný servis a opravy, dálniční známka, silniční daň, výměna a přezouvání pneumatik atd. Tyto náklady jsou podle vhodného časového úseku (třeba za rok) sečteny a dispečerem je odhadnuto, kolik kilometrů za tento časový úsek vozidlo naježdí. Podílem těchto dvou čísel získáme náklady na 1 ujetý kilometr provozu motorového vozidla.

- **Délky všech tras v kilometrech spojující všechny waypointy mezi sebou navzájem (každý s každým).**
- **Cenu (tržbu) za realizovanou zakázku v každém waypointu.**

Na základě těchto tří údajů je pro každý možný úsek mezi všemi waypointy vypočítána a sestavena „matice vah“. Tuto matici lze též nazvat „maticí zisků“, protože každé číslo mimo hlavní diagonálu reprezentuje čistý zisk (tržba minus náklady), který nám zůstane po realizaci dané zakázky.

Po vypočítání a složení „matice zisků“ je tato převedena na textový řetězec a pomocí jazyka PHP uložena do souboru dotaz_zisk.txt na pevný disk. Z tohoto souboru si výpočetní algoritmus vyzvedne data, složí si je zpět do matice a začíná hledat takovou kombinaci, která jako celek bude mít za výsledek největší zisk.

4.7.2 Zjednodušující podmínky

Pro snadnější implementaci celého programu a výpočtu nejziskovější trasy byly uvažovány následující zjednodušující podmínky:

- Rozvoz bude zajištěn osobním vozidlem (dodávkou) do celkové hmotnosti 3,5 tuny, takže při průjezdu po zpoplatněných úsecích (dálnice, rychlostní komunikace) nebude placeno mýto. Poplatek za dálnici je vyřešen nákupem roční dálniční známky, která již bude započítána do nákladů vozidla za 1 ujetý kilometr.
- Nejsou uvažovány přestávky v řízení (povinný odpočinek řidiče, přestávka na oběd) a s tím související (velmi mírné) zvyšování nákladů.
- Nejsou uvažovány doby nutné pro nakládku nebo vykládku zboží.
- Nejsou uvažovány doby strávené v hustém provozu v ranních a odpoledních špičkách a čekání kvůli dopravní nehodě.
- Není uvažována varianta, že nákladový prostor vozidla bude před dokončením celé trasy zcela zaplněn tak, že další zboží už nepojme.

- Nejsou uvažovány různé provozní hodiny v každém waypointu (tzv. „time-window“), ve kterých by bylo možné náklad naložit nebo složit a mimo tyto hodiny by to možné nebylo. Je uvažována varianta, že například v 8:00 všichni zákazníci své provozovny otevřou a v 18:00 je zase všichni zavřou. Mezi 8:00 až 18:00 je možné kdykoliv kamkoliv přijet a realizovat jakoukoliv zakázku.
- Nejsou uvažovány dodatečné daňové odvody ze zisků, které by tento zisk dále snižovaly a ovlivňovaly by výsledek, například daň z příjmů nebo DPH.
- Není uvažována varianta, že řidič některý waypoint projede v jiném pořadí, než je stanoveno nebo ho úplně vynechá (například kvůli zpoždění z důvodu hustého provozu na poslední waypoint dorazí 18:15 a provozovna bude již uzavřena).

Počet waypointů, které je řidič během směny schopen objet, musí dispečer na základě zkušeností s provozem v dané lokalitě odhadnout a musí započítat i nějakou časovou rezervu pro mimořádné situace.

4.7.3 Vlastnosti výpočetního algoritmu

Výpočetní algoritmus je postaven na tzv. „výpočtu hrubou silou“. To znamená, že propočítá všechny existující kombinace a máme jistotu, že nalezne tu nejlepší. Hlavní nevýhoda je ovšem to, že při větším počtu waypointů narůstá počet kombinací takovým způsobem, že od určité hranice se současnou výpočetní technikou již není možné (a je matematicky dokázáno, že se sebelepším počítačem nikdy ani možné nebude) takové množství výpočtů reálně zvládnout.

Přehled základních vlastností výpočetního algoritmu:

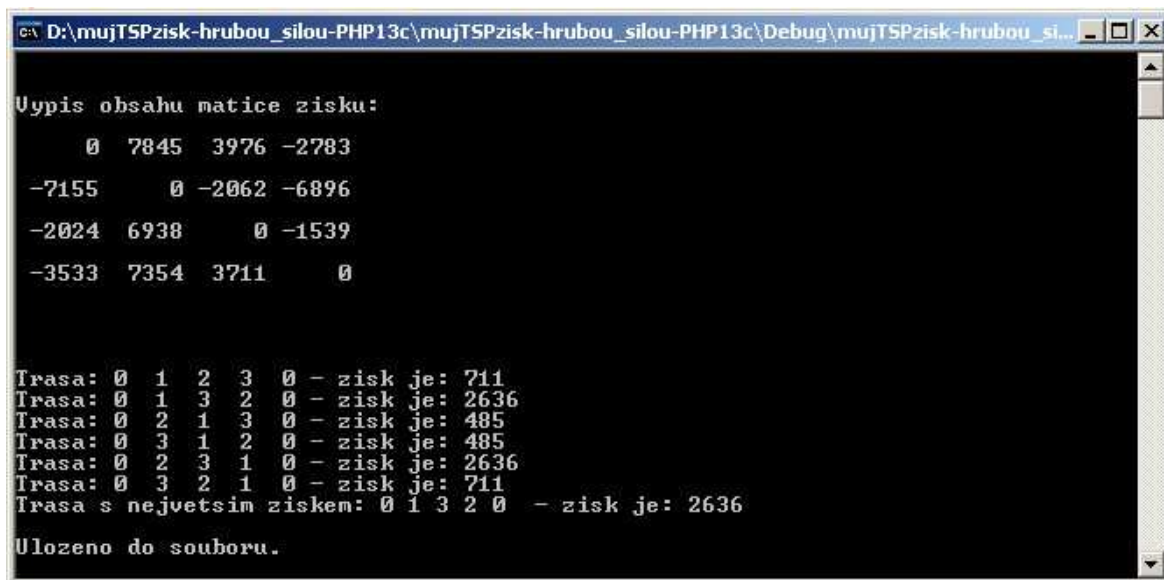
- Algoritmus je schopen spočítat trasu pro maximálně 10 waypointů. Při pokusu o výpočet s větším množstvím waypointů dojde k chybě.
- Je uvažována varianta, že se vozidlo na konci směny vrací zpět na základnu, odkud vyjelo. Jedná se tedy o okružní jízdu.
- Při výpočtu jsou použity 2 nezávislé algoritmy.

4.7.4 Algoritmus č. 1 – permutace

Permutace znamená vzájemné kombinace všech prvků ze zadané množiny. Zároveň nejsou počítány varianty, kdy by se jeden nebo více prvků z dané množiny odstranily a k výsledku by se přidaly i kombinace bez těchto prvků.

Tento algoritmus je použit pro výpočet nejziskovější trasy a její zobrazení v mapovém podkladu. Předpokládá se, že uživatel všechny zadané waypointy opravdu chce projet, i když některé jsou jasně prodělečné. Je to z důvodu, pokud by existovaly nějaké další okolnosti, proč nevýhodnou zakázku přeci jen realizovat. Příkladem může být snaha zákazníkovi vyhovět za účelem udržení obchodního kontaktu, nebo nutnosti vyřízení jiných záležitostí blízko tohoto waypointu.

Z Internetu byl stažen vhodný permutační algoritmus [32] a byl implementován tak, aby správně načítal vstup a na výstup složil výsledek do požadovaného tvaru. Jelikož startovní a cílový bod jsou pevně dány (na obrázku jsou reprezentovány číslem 0), permutují se pouze kombinace mezi těmito nulami. Funkci algoritmu č.1 pěkně ilustruje obrázek konzole při výpočtu se zapnutými výpisy:



```
C:\D:\mujTSPzisk-hrubou_silou-PHP13c\mujTSPzisk-hrubou_silou-PHP13c\Debug\mujTSPzisk-hrubou_si...
Uypis obsahu matice zisku:
    0  7845  3976 -2783
-7155    0 -2062 -6896
-2024  6938    0 -1539
-3533  7354  3711    0

Trasa: 0 1 2 3 0 - zisk je: 711
Trasa: 0 1 3 2 0 - zisk je: 2636
Trasa: 0 2 1 3 0 - zisk je: 485
Trasa: 0 3 1 2 0 - zisk je: 485
Trasa: 0 2 3 1 0 - zisk je: 2636
Trasa: 0 3 2 1 0 - zisk je: 711
Trasa s nejvetsim ziskem: 0 1 3 2 0 - zisk je: 2636

Uloženo do souboru.
```

Obr. 21. Funkce algoritmu permutace

Algoritmus č. 1 svůj výstup uloží do souboru `odpoved_tsp1_zisk.txt`, který je PHP skriptem načten a data jsou předána zpět do aplikace.

4.7.5 Algoritmus č. 2 – kombinace s permutací

Kombinace znamená, že z dané množiny jsou postupně odstraňovány žádné a dále postupně jeden až všechny prvky. Pod každou touto kombinací je zavolána permutace, která z vybrané množiny prvků zjistí všechny možné kombinace bez dalšího odstraňování prvků. Zároveň jsou ošetřeny stavy, kdy vznikne nesmyslná kombinace, která v reálném provozu nemůže nastat, jako například když není žádný waypoint nebo když chybí startovní waypoint.

Algoritmus č. 2 je spouštěn hned po skončení výpočtu Algoritmu č. 1 a jeho výsledek je uživateli zobrazován jako tip pro získání ziskovější trasy. Uživatel se může rozhodnout, zda se bude řídit doporučením a navrhované waypointy smaže, nebo zda mu vyhovuje původní trasa.

Z Internetu byl stažen vhodný kombinační algoritmus [33] a byl implementován tak, aby pracoval nad algoritmem č. 1 a volal si ho pro výpočet každé platné kombinace. Funkci algoritmu č. 2 ilustruje následující obrázek konzole při výpočtu se zapnutými výpisy:

```

D:\mujTSPzisk-hrubou_silou-s_kombinaci19\mujTSPzisk-hrubou_silou-s_kombinaci19\Debug\mujTSP...
  0 7845 3976 -2783
-7155 0 -2062 -6896
-2024 6938 0 -1539
-3533 7354 3711 0
Ted mas kombinaci 0 0 0 0 -> STARTOVNI MESTO NEEEXISTUJE - VARIANTU PRESKAKUJI
Ted mas kombinaci 1 0 0 0 -> EXISTUJE POUZE JEDNO MESTO - VARIANTU PRESKAKUJI
Ted mas kombinaci 0 1 0 0 -> STARTOVNI MESTO NEEEXISTUJE - VARIANTU PRESKAKUJI
Ted mas kombinaci 0 0 1 0 -> STARTOVNI MESTO NEEEXISTUJE - VARIANTU PRESKAKUJI
Ted mas kombinaci 0 0 0 1 -> STARTOVNI MESTO NEEEXISTUJE - VARIANTU PRESKAKUJI
Ted mas kombinaci 1 1 0 0 -> zavolej si na ni Permutaci()...
Aktualni trasa: a b a <Akt. t. red. 0 1 0 > aktualniZisk je: 690
Ted mas kombinaci 1 0 1 0 -> zavolej si na ni Permutaci()...
Aktualni trasa: a c a <Akt. t. red. 0 1 0 > aktualniZisk je: 1952
Ted mas kombinaci 1 0 0 1 -> zavolej si na ni Permutaci()...
Aktualni trasa: a d a <Akt. t. red. 0 1 0 > aktualniZisk je: -6316
Ted mas kombinaci 0 1 0 1 -> STARTOVNI MESTO NEEEXISTUJE - VARIANTU PRESKAKUJI
Ted mas kombinaci 0 0 1 1 -> STARTOVNI MESTO NEEEXISTUJE - VARIANTU PRESKAKUJI
Ted mas kombinaci 1 1 1 0 -> zavolej si na ni Permutaci()...
Aktualni trasa: a b c a <Akt. t. red. 0 1 2 0 > aktualniZisk je: 3759
Aktualni trasa: a c b a <Akt. t. red. 0 2 1 0 > aktualniZisk je: 3759
Ted mas kombinaci 1 1 0 1 -> zavolej si na ni Permutaci()...
Aktualni trasa: a b d a <Akt. t. red. 0 1 2 0 > aktualniZisk je: -2584
Aktualni trasa: a d b a <Akt. t. red. 0 2 1 0 > aktualniZisk je: -2584
Ted mas kombinaci 1 0 1 1 -> zavolej si na ni Permutaci()...
Aktualni trasa: a c d a <Akt. t. red. 0 1 2 0 > aktualniZisk je: -1096
Aktualni trasa: a d c a <Akt. t. red. 0 2 1 0 > aktualniZisk je: -1096
Ted mas kombinaci 0 1 1 1 -> STARTOVNI MESTO NEEEXISTUJE - VARIANTU PRESKAKUJI
Ted mas kombinaci 1 1 1 1 -> zavolej si na ni Permutaci()...
Aktualni trasa: a b c d a <Akt. t. red. 0 1 2 3 0 > aktualniZisk je: 711
Aktualni trasa: a c b d a <Akt. t. red. 0 2 1 3 0 > aktualniZisk je: 485
Aktualni trasa: a d b c a <Akt. t. red. 0 3 1 2 0 > aktualniZisk je: 485
Aktualni trasa: a b d c a <Akt. t. red. 0 1 3 2 0 > aktualniZisk je: 2636
Aktualni trasa: a c d b a <Akt. t. red. 0 2 3 1 0 > aktualniZisk je: 2636
Aktualni trasa: a d c b a <Akt. t. red. 0 3 2 1 0 > aktualniZisk je: 711

Trasa s nejvetsim ziskem: 0 1 2 0 - zisk je: 3759
Uloženo do souboru.

```

Obr. 22. Funkce algoritmu kombinace s permutací

Algoritmus č. 2 svůj výstup uloží do souboru `odpoved_tsp2_zisk.txt`, jehož obsah je opět PHP skriptem načten a data jsou předána zpět do aplikace.

U obou algoritmů byl pro ukázkou použit stejný vstup a vidíme, že algoritmus permutace s kombinací byl schopen nalézt ziskovější trasu a to za předpokladu, že se vynechá waypoint č. 3.

5 ZABEZPEČENÍ APLIKACE

U programu, který je postavený jako webová aplikace anebo ke svému chodu využívá webové služby, musí být věnována zvýšená pozornost jeho zabezpečení. Je to z toho důvodu, protože útok na aplikaci může být veden vzdáleně přes síť a k napadení může být využita kterákoliv komponenta, technologie nebo služba, kterou aplikace ke svojí funkci využívá.

U mojí aplikace byla vyhodnocena potenciální rizika, která by mohla přicházet do úvahy a byly navrženy způsoby jejich řešení.

5.1 Útok na databázi

Jako nejcitlivější součást aplikace, která by potenciálního útočníka mohla zajímat je rozhraní pro připojení k databázi a přístupu k datům uložených v databázi. Předpokládaná metoda použitého útoku je SQL injection.

SQL injection je způsob napadení databáze podstrčením vlastního kódu přes neošetřený vstup. Tímto postupem můžeme například získat data, smazat celou tabulku a v extrémním případě zjistit jméno a heslo správce databáze a získat nad cizí databází plnou kontrolu.

V mojí aplikaci tento typ útoku naštěstí nepřichází do úvahy, protože připojení k databázi a načítání dat zajišťuje fixně psaný skript, chybí zde vstupní formulářový prvek, do kterého by potenciální útočník mohl SQL dotaz zapsat. A dále také proto, že ovládací interface běží na lokálním počítači a není možné ho ovládat z webu.

Pro zvýšení bezpečnosti databáze je předpoklad, že přihlašovací účet, který používá aplikace, bude mít v databázi nastavena omezená práva pouze ke čtení a přístup bude pouze do těch tabulek, které potřebuje ke své funkci. Do zbylých částí databáze nebude mít aplikace přístup. Tímto je potenciálnímu útočníkovi znemožněno udělat nějaké škody, i kdyby se dostal k fyzickému počítači, na kterém je aplikace provozována a ručně editoval připojovací skripty.

5.2 Odposlech síťového provozu

Aplikace komunikuje po síti pouze s databází nebo s geokodérem. Při odposlechu tohoto síťového provozu by tedy potenciální útočník mohl zjistit adresu každého zákazníka, při komunikaci s databází i jméno zákazníka a cenu každé zakázky.

Proto je při komunikaci s databází vhodné používat přenosy šifrované pomocí protokolu SSL, případně TLS.

U každého druhu databáze je třeba toto nastavit na straně databáze individuálně, pro databázový systém MySQL toho lze dosáhnout například instalací nástroje OpenSSL [34]. Na straně serveru (v našem případě Apache) je dále nutné podporu SSL protokolu nastavit [35].

Při dotazování geokodéru by útočník mohl zjistit pouze adresy zákazníků, což z hlediska bezpečnosti představuje poměrně malé riziko. Geokodér Mapy.cz nativně komunikuje v nezašifrované podobě (z důvodu menší zátěže serveru), nicméně pokud požadavek na geokódování přijde přes protokol HTTPS, automaticky také přes HTTPS odpoví.

5.3 Ovládání aplikace nepovolanou osobou

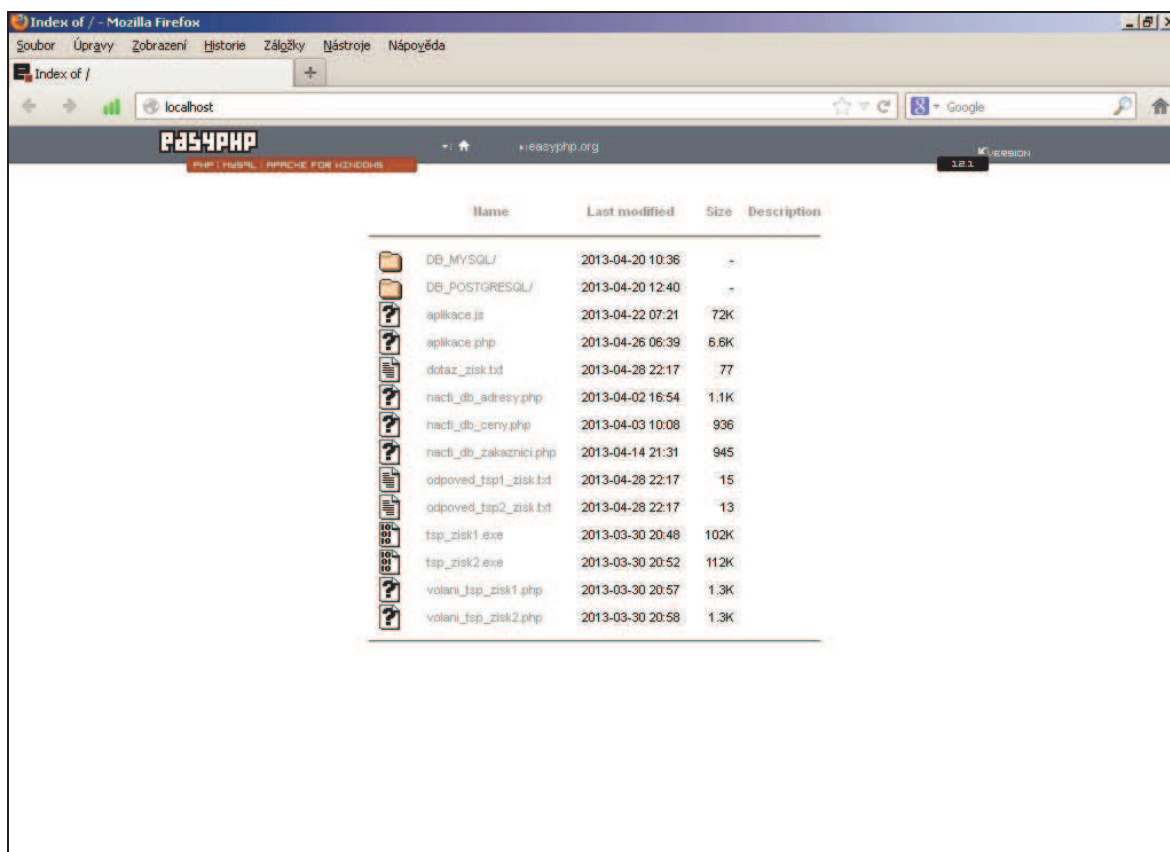
Přístupu nepovolanou osobou k aplikaci, která je uložena na lokálním PC, lze čelit poměrně jednoduše, například použitím šifrovaného souborového systému v kombinaci se zapnutím uživatelských účtů pro přihlášení k počítači.

6 POPIS OVLÁDÁNÍ APLIKACE

Předpokladem pro spuštění aplikace je mít správně nainstalované všechny potřebné programové balíky uvedené výše, vytvořenou a odpovídajícími daty naplněnou databázi a zkopírované soubory do složky www.

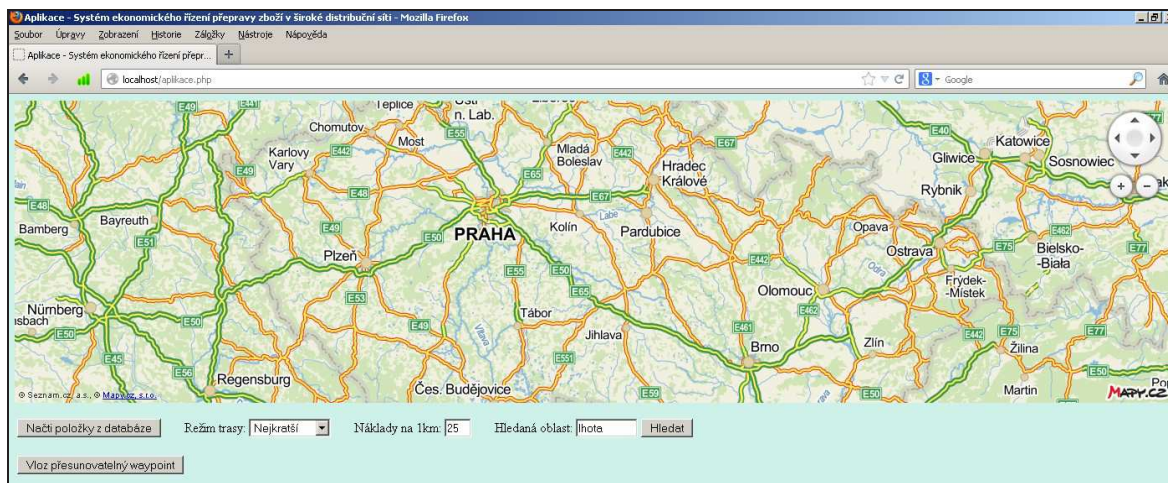
6.1 Spuštění aplikace

Spustíme webový prohlížeč a do příkazového řádku napíšeme „localhost“ a stiskneme klávesu enter. Měla by se objevit souborová struktura složky www.



Obr. 23. Ovládání aplikace - spuštění

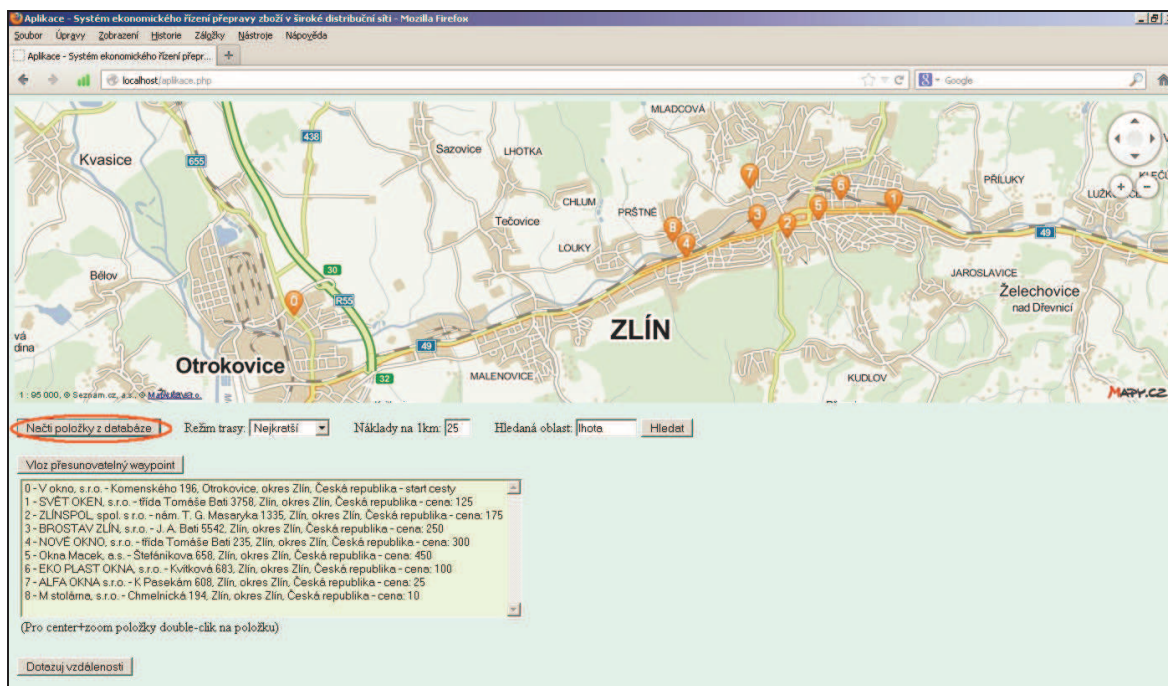
Levým tlačítkem myši klepneme na soubor aplikace.php. Následně se zobrazí uživatelský interface.



Obr. 24. Ovládání aplikace - uživatelský interface

6.2 Načtení zákazníků z databáze

Načítání databáze se spustí po klepnutí na tlačítko „Načti položky z databáze“. Pokud již nebyl zadán startovní waypoint (bod, odkud vozidlo vyjíždí a kam se vrací), je o této skutečnosti uživatel informován dialogem a jako startovní waypoint se nastaví první načítaná položka z databáze.

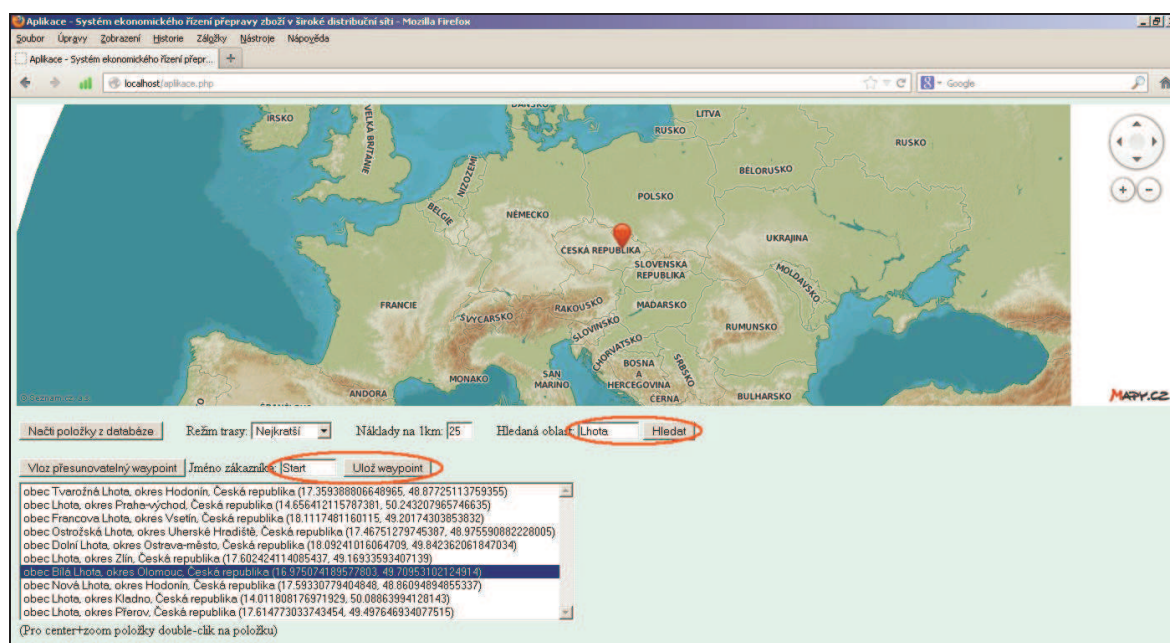


Obr. 25. Ovládání aplikace - načtení dat z databáze

6.3 Ruční zadání zákazníka

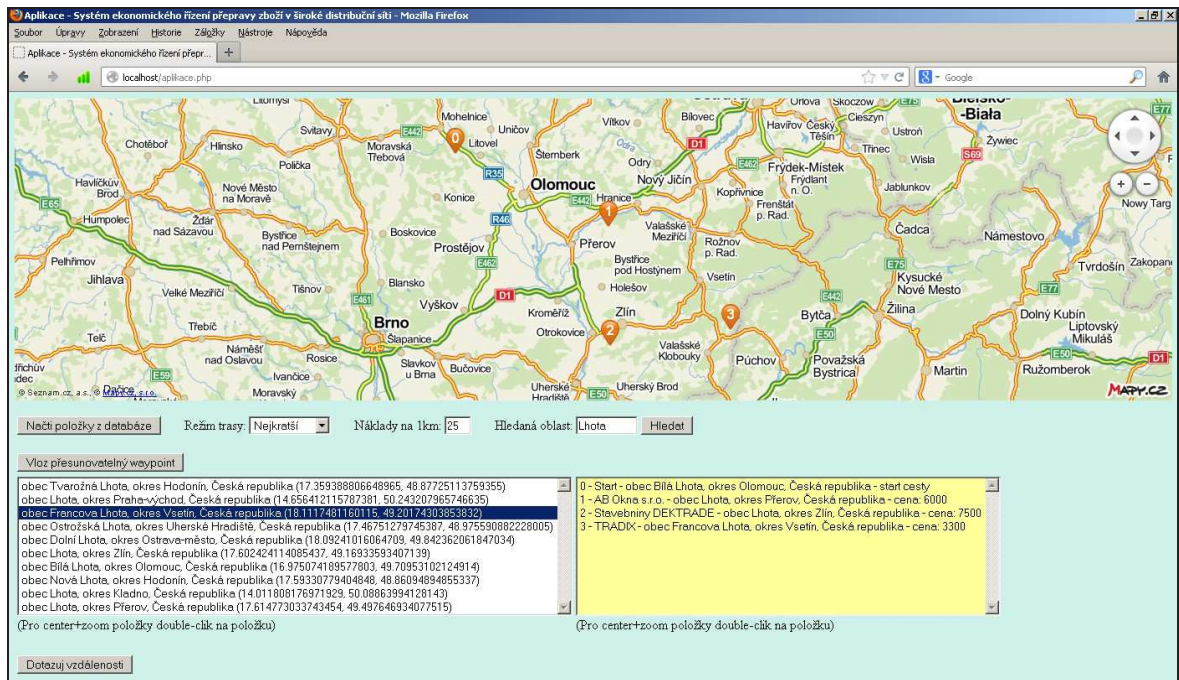
Do pole „Hledaná oblast“ napíšeme adresu a stiskneme tlačítko „Hledat“. Pokud je na dotaz nalezen pouze jeden výsledek, aplikace zobrazí dialog s nalezeným výsledkem a na adresu vystředí a zvětší mapový podklad.

Pokud je nalezeno více výsledků, aplikace zobrazí prvních 10 nalezených, zobrazí mapu celé Evropy a nalezené výsledky zobrazí v bílém selectboxu. Klepnutím levým tlačítkem myši do selectboxu se adresa zobrazí pomocí markeru na mapě, objeví se textové pole pro zadání jména zákazníka a tlačítko s možností uložení. Pokud již byla nějaká položka uložena (čili již byl zadán startovní waypoint), objeví se navíc i textové pole pro zadání ceny zakázky.



Obr. 26. Ovládání aplikace - ruční zadání zákazníka

Klepnutím na tlačítko „Ulož waypoint“ se vybraná položka uloží do žlutého selectboxu a na všechny dosud zadané waypointy se mapa zvětší tak, aby byly všechny viditelné.



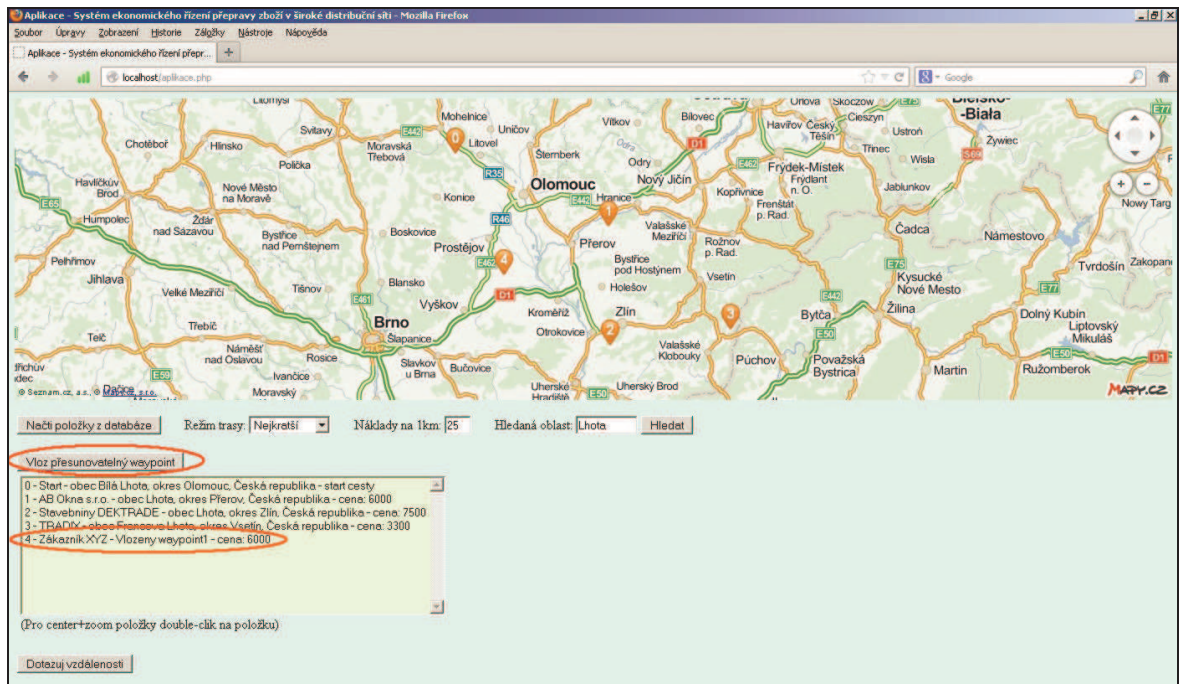
Obr. 27. Ovládání aplikace - uložení zákazníka

6.4 Smazání položky

Kteroukoliv uloženou položku lze kdykoliv smazat, po klepnutí na vybranou položku ve žlutém selectboxu se objeví tlačítko „Smaž waypoint“, po jehož stisknutí je waypoint odstraněn.

6.5 Vložení waypointu přímo do mapového podkladu

Pro případ, že by geokodér nenašel zadanou adresu, byla přidána funkce, kdy lze po klepnutí na tlačítko „Vlož přesouvateľný waypoint“ vložit waypoint do mapového podkladu. Stiknutím levého tlačítka myši, jeho držetím a posouváním myši lze waypoint umístit na požadované místo na mapě. Následně stačí vypsát údaje o zákazníkovi a ceně a klepnutím na tlačítko „Ulož waypoint“ je bod uložen.

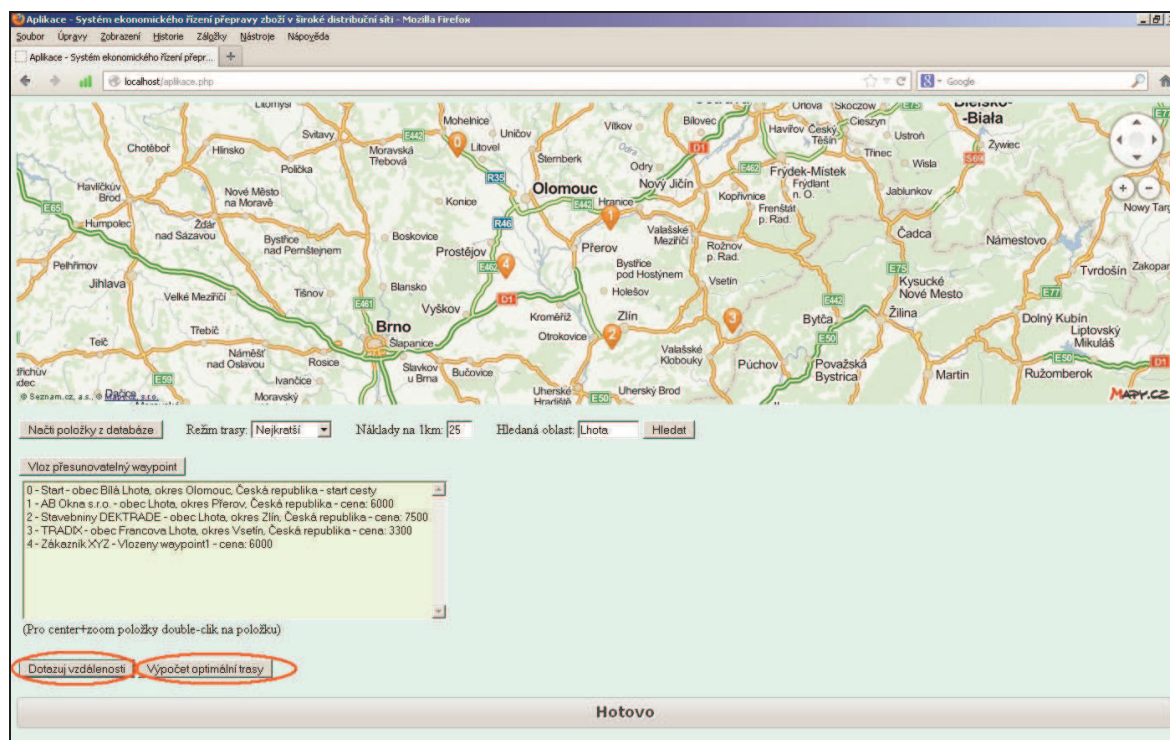


Obr. 28. Ovládání aplikace - vložení přesunovatelného waypointu

6.6 Dotazování geokodéru

Před samotným dotazováním je třeba zadat náklad na 1 km provozu vozidla. Také je možné přepínat režim trasy mezi „Nejkratší“ a „Nejrychlejší“.

Po skončení vkládání waypointů je klepnutím na tlačítko „Dotazuj vzdálenosti“ zahájeno postupné odesílání požadavků na zjištění vzdálenosti mezi waypointy. Průběh vyřizování požadavků geokodérem lze sledovat na progressbaru. Po skončení dotazování se v progressbaru objeví nápis „HOTOVO“ a nad ním tlačítko „Výpočet optimální trasy“.

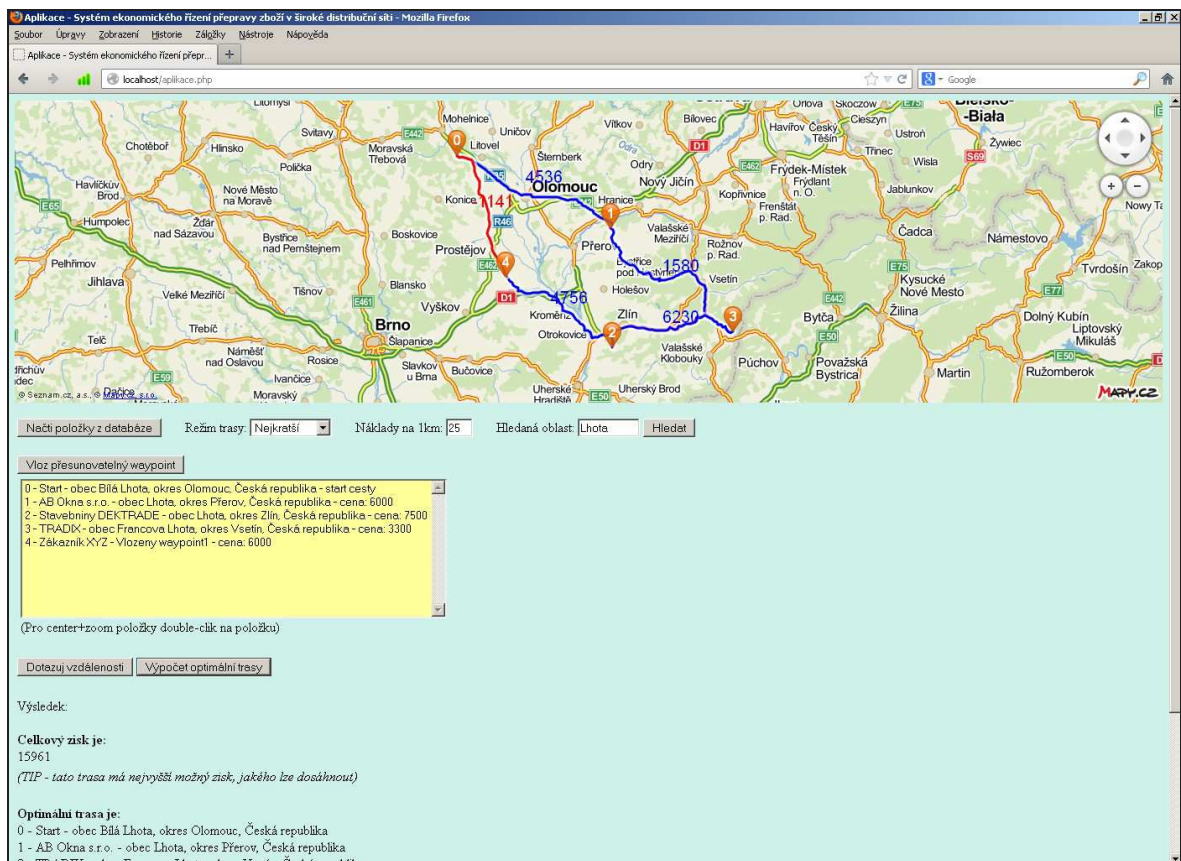


Obr. 29. Ovládání aplikace - dotazování geokodéru

6.7 Výpočet a zobrazení optimální trasy

Klepnutím na tlačítko „Výpočet optimální trasy“ jsou data předána algoritmu a začíná výpočet. Po jeho dokončení (což je prakticky okamžitě) je výsledek vypsán pod okno s mapovým podkladem a do mapy je výsledná trasa postupně vykreslována. Modře jsou vyznačeny úseky, na kterých cena za vyhotovení cílové zakázky převyšuje náklady na projetí tohoto úseku, červeně kde jsou náklady na průjezd tohoto úseku vyšší než je cena dané zakázky. Čísla znamenají konkrétní cenové bilance pro každý úsek.

Poslední (návratový) úsek okružní cesty je vždy červený, protože v cíli již není přítomen žádný platící zákazník, který by cestovní náklady na návrat kompenzoval.



Obr. 30. Ovládání aplikace - zobrazení výsledné trasy

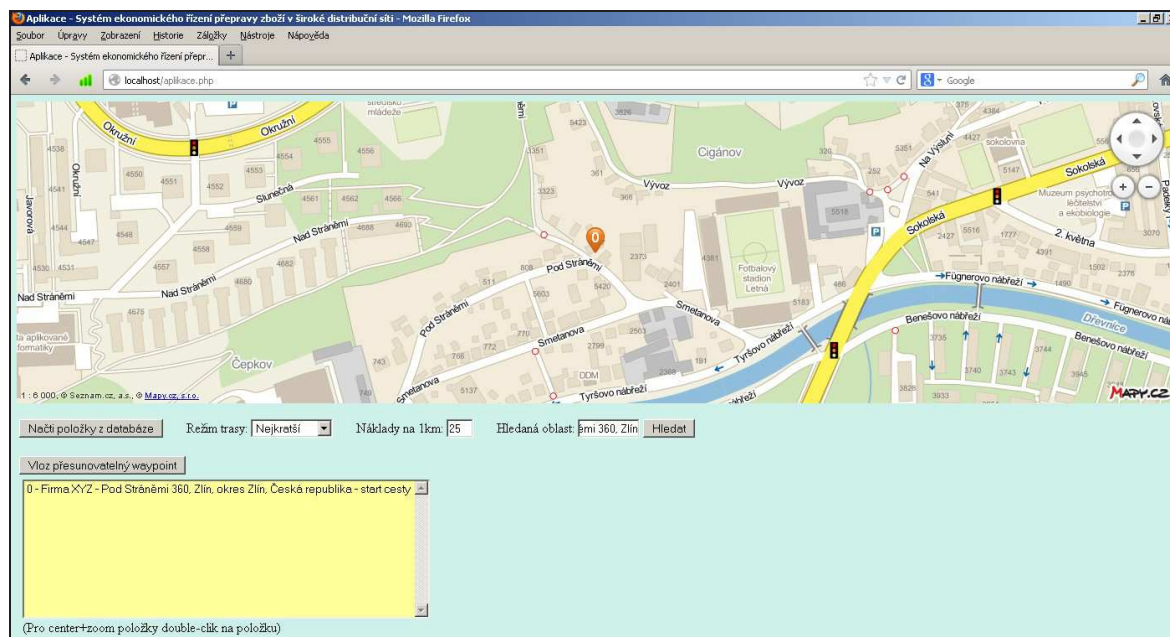
6.8 Další možnosti ovládání aplikace

- Načítání z položek z databáze lze libovolně kombinovat s ručním zadáváním waypointů.
- Pokud chceme začít ukládat nebo načítat waypointy znovu od začátku do prázdného žlutého selectboxu, než body odmazávat po jednom, je rychlejší celou stránku znovu načíst (tlačítko „Obnovit“ ve webovém prohlížeči).
- Již jednou vypočítanou optimální trasu lze přidáním nebo smazáním waypointů upravit a spustit dotazování a následně výpočet pro tyto nové hodnoty.
- Kolečkem myši lze v okně mapového podkladu mapu libovolně přibližovat nebo oddalovat, stiskem a podržením levého tlačítka myši lze mapu posouvat. Ekvivalentně také fungují ovládací prvky v pravém horním rohu mapy.
- Jméno zákazníka nemusí být zadáno.

- Cena zakázky by zadána být měla, pokud není, je na to uživatel upozorněn dialogem a cena dotyčné zakázky je nastavena na hodnotu 0.
- Režim trasy musí být zvolen ještě před dotazováním vzdáleností a nesmí být poté před stiskem tlačítka „Výpočet optimální trasy“ změněn. V opačném případě by byly hodnoty vypočtené pro jeden režim trasy, ale úseky v mapovém podkladu by byly znázorněny pro druhý režim trasy.
- Dvojitým klepnutím na položku v bílém nebo žlutém selectboxu bude tato položka v mapovém podkladu vycentrována a zvětšena.

7 MODELOVÝ PŘÍPAD POUŽITÍ APLIKACE

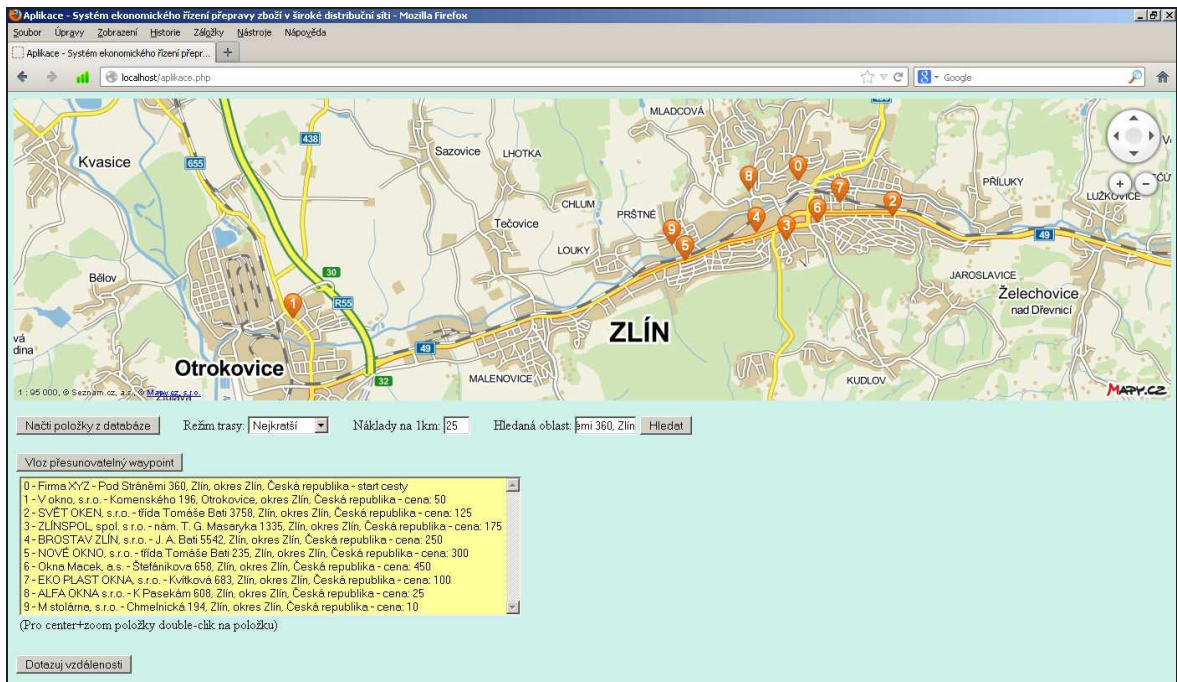
Funkci aplikace si ověříme na modelovém příkladu použití. Mějme firmu XYZ se sídlem na ulici Pod Stráněmi 360 ve Zlíně.



Obr. 31. Modelový příklad použití – sídlo společnosti XYZ

Firma XYZ, která se zabývá výrobou a distribucí stavebního kování oken a dveří má možnost realizovat 9 zakázek pro 9 různých zákazníků. Za každou vyhotovenou zakázku již byla předběžně projednána cena, kterou je každý ze zákazníků za doručení zboží ochoten zaplatit.

Dispečer plánující přepravu dostal za úkol maximalizovat zisk z těchto obchodů, i za cenu nedodání zboží k některému ze zákazníků. Dispečer vloží adresy a ceny zakázek do aplikace.



Obr. 32. Modelový příklad použití – zákazníci

Dispečer spustí dotazování vzdáleností a následně výpočet optimální trasy. Dostane tento výsledek:



Obr. 33. Modelový příklad použití – výsledek č. 1

Výsledek:

Celkový zisk je:

679

(TIP - větší zisk: 1096 dosáhnete smazáním waypointů 1, 9)

Optimální trasa je:

- 0 - Firma XYZ - Pod Stráněmi 360, Zlín, okres Zlín, Česká republika
- 8 - ALFA OKNA s.r.o. - K Pasekám 608, Zlín, okres Zlín, Česká republika
- 9 - M stolárna, s.r.o. - Chmelnická 194, Zlín, okres Zlín, Česká republika
- 5 - NOVÉ OKNO, s.r.o. - třída Tomáše Bati 235, Zlín, okres Zlín, Česká republika
- 1 - V okno, s.r.o. - Komenského 196, Otrokovice, okres Zlín, Česká republika
- 4 - BROSTAV ZLÍN, s.r.o. - J. A. Bati 5542, Zlín, okres Zlín, Česká republika
- 3 - ZLÍNSPOL, spol. s.r.o. - nám. T. G. Masaryka 1335, Zlín, okres Zlín, Česká republika
- 6 - Okna Macek, a.s. - Štefánikova 658, Zlín, okres Zlín, Česká republika
- 2 - SVĚT OKEN, s.r.o. - třída Tomáše Bati 3758, Zlín, okres Zlín, Česká republika
- 7 - EKO PLAST OKNA, s.r.o. - Kvítková 683, Zlín, okres Zlín, Česká republika
- 0 - Firma XYZ - Pod Stráněmi 360, Zlín, okres Zlín, Česká republika

Celková vzdálenost je:

32 km

Celkový čas je:

0 hodin, 52 minut, 30 sekund.

Dispečer dle doporučení aplikace smaže waypointy číslo 1 a 9 a znovu spustí výpočet.

Aplikace mu zobrazí tento výsledek:



Obr. 34. Modelový příklad použití – výsledek č. 2

Výsledek:

Celkový zisk je:

1096

(TIP - tato trasa má nejvyšší možný zisk, jakého lze dosáhnout)

Optimální trasa je:

0 - Firma XYZ - Pod Stráněmi 360, Zlín, okres Zlín, Česká republika

6 - EKO PLAST OKNA, s.r.o. - Kvitková 683, Zlín, okres Zlín, Česká republika

1 - SVĚT OKEN, s.r.o. - třída Tomáše Bati 3758, Zlín, okres Zlín, Česká republika

5 - Okna Macek, a.s. - Štefánikova 658, Zlín, okres Zlín, Česká republika

2 - ZLÍNSPOL, spol. s.r.o. - nám. T. G. Masaryka 1335, Zlín, okres Zlín, Česká republika

3 - BROSTAV ZLÍN, s.r.o. - J. A. Bati 5542, Zlín, okres Zlín, Česká republika

4 - NOVÉ OKNO, s.r.o. - třída Tomáše Bati 235, Zlín, okres Zlín, Česká republika

7 - ALFA OKNA s.r.o. - K Pasekám 608, Zlín, okres Zlín, Česká republika

0 - Firma XYZ - Pod Stráněmi 360, Zlín, okres Zlín, Česká republika

Celková vzdálenost je:

13 km

Celkový čas je:

1 hodin, 19 minut, 41 sekund.

Dispečer vidí, že vynecháním zakázek u zákazníků V okno, s.r.o. (waypoint č. 1) a M stolárna, s.r.o. (waypoint č. 9) bude zisk z této okružní trasy nejvyšší možný. Vytiskne tedy cestovní pokyn a obchodní cesta může být zahájena.

ZÁVĚR

Aplikace pro ekonomické řízení přepravy zboží v široké distribuční síti pracuje podle předpokladů. V této chvíli aplikace zvládne vypočítat optimální trasu pro maximálně 10 bodů cesty, což je v rámci obchodních cest mezi městy ještě dostatečné. Nicméně pro cesty v rámci jednoho města je to málo a řidič těchto 10 bodů může projet cca za 2 hodiny, což nestačí na pokrytí jeho denního výkonu. Návrh a hlavně odladění heuristického algoritmu nebylo předmětem řešení této práce.

Proto by se další práce na této aplikaci měla ubírat směrem k nahrazení výpočetního algoritmu hrubou silou nějakou formou heuristického výpočtu, který zajistí možnost práce s větším počtem průjezdních bodů při zachování dobrých výsledků optimalizované trasy.

Dalším vylepšením aplikace může být například přidání více možností parametrizace trasy (například provozní hodiny u zákazníků nebo rozvoz zboží a zpětný svoz obalů) nebo grafické vylepšení ovládacího rozhraní.

V této chvíli je aplikace připravena pro nasazení a testování v reálné společnosti.

CONCLUSION

The Application for economical control of an assets transportation within complicated distributional relations works as expected. At this point, the application can calculate the optimal route for a maximum of 10 waypoints, which is still sufficient for the trade routes between cities. However, for the travel within the city it is not enough and the driver all these 10 points can visit approximately within 2 hours, which is not enough to utilize his daily shift. Design and especially heuristic algorithm verification was not subject in this thesis.

Therefore, further work on this application should go towards replacing the computing brute force algorithm with some form of heuristic calculation, which ensures the ability to work with greater numbers of waypoints while keeping good optimized route results.

Further application improvement may be adding more parameterization route selections (for example, the operating hours for customers or goods delivery and picking up containers) or control interface graphical enhancements.

At this point, the application is ready for deployment and testing in a real company.

SEZNAM POUŽITÉ LITERATURY

- [1] ZAMAZAL, Zbyněk. *Analýza web konferenčních systémů*. Zlín, 2011. Bakalářská práce. UTB, FAI Zlín.
- [2] Rozvoz a svoz zboží | CSmap. s.r.o. *Desktop mapping | CSmap. s.r.o.* [online]. 2008 [cit. 2013-04-08]. Dostupné z: <http://www.csmmap.cz/optimalizace-dopravy/rozvoz-a-svoz-zbozi.htm>
- [3] Doprava | LOGICON Partner s.r.o. LOGICON PARTNER S.R.O. *Logistické poradenství | LOGICON Partner s.r.o.* [online]. 2013 [cit. 2013-04-08]. Dostupné z: <http://www.logicon.cz/sw-podpora/doprava>
- [4] Mapy Google. GOOGLE. *Google* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <http://maps.google.cz/>
- [5] Google Maps API – Google Developers. GOOGLE. *Google* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <https://developers.google.com/maps/>
- [6] Google Maps API licensing – Google Maps API – Google Developers. GOOGLE. *Google* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <https://developers.google.com/maps/licensing>
- [7] SEZNAM.CZ, a.s. *Mapy.cz* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <http://mapy.cz/>
- [8] API Mapy.cz. SEZNAM.CZ, a.s. *Mapy.cz* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <http://api4.mapy.cz/>
- [9] OPEN SOURCE GEOSPATIAL FOUNDATION. *OpenLayers: Home* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <http://www.openlayers.org/>
- [10] OPENSTREETMAP FOUNDATION. *OpenStreetMap Search* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <http://www.openstreetmap.org/>
- [11] OpenLayers. *OpenLayers* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <http://dev.openlayers.org/apidocs/files/OpenLayers-js.html>
- [12] OpenLayers Examples. *OpenLayers* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <http://openlayers.org/dev/examples/>

- [13] *OpenRouteService.org:Home* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <http://openrouteservice.org/>
- [14] *PgRouting Project – Open Source Routing Library* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <http://pgrouting.org/>
- [15] Forum napoveda.seznam.cz • Zobrazit fórum – Mapy API v4.0. *Mapy.cz* [online]. 2013 [cit. 2013-04-09]. Dostupné z: <http://napoveda.seznam.cz/forum/viewforum.php?f=31>
- [16] JAK – javascriptová knihovna – Seznam. *Seznam – Najdu tam, co neznám* [online]. 2013 [cit. 2013-04-11]. Dostupné z: <http://jak.seznam.cz/example/>
- [17] Javascript – návody, objekty, příklady. *Jak psát web, návod na html stránky* [online]. 2010 [cit. 2013-04-13]. Dostupné z: <http://www.jakpsatweb.cz/javascript/>
- [18] HTML příručka, přehled HTML tagů. *Jak psát web, návod na html stránky* [online]. 2010 [cit. 2013-04-13]. Dostupné z: <http://www.jakpsatweb.cz/html/>
- [19] THE JQUERY FOUNDATION. *JQuery* [online]. 2013 [cit. 2013-04-13]. Dostupné z: <http://jquery.com/>
- [20] THE PHP GROUP. *PHP: Hypertext Preprocessor* [online]. 2013 [cit. 2013-04-13]. Dostupné z: <http://php.net/>
- [21] Krokodýlový databáze : SQL. *Krokodýlový databáze : HomePage* [online]. 2011 [cit. 2013-04-13]. Dostupné z: <http://krokodata.vse.cz/SQL/SQL>
- [22] *EasyPHP | Install a local WAMP server : PHP 5 VC9, Apache 2 VC9, MySQL 5, PhpMyAdmin, Xdebug and Modules on Windows XP/Vista/Seven* [online]. 2013 [cit. 2013-04-13]. Dostupné z: <http://www.easyphp.org/>
- [23] *MySQL :: The world's most popular open source database* [online]. 2013 [cit. 2013-04-13]. Dostupné z: <http://www.mysql.com/>
- [24] *PostgreSQL: The world's most advanced open source database* [online]. 2013 [cit. 2013-04-13]. Dostupné z: <http://www.postgresql.org/>
- [25] Reference – C++ Reference. *Cplusplus.com – The C++ Resources Network* [online]. 2013 [cit. 2013-04-13]. Dostupné z: <http://www.cplusplus.com/reference/>

- [26] Microsoft Visual Studio | Vývoj od návrhu až po nasazení | MSDN Česká republika. *Microsoft Home Page | Devices and Services* [online]. 2013 [cit. 2013-04-14]. Dostupné z: <https://www.microsoft.com/cze/msdn/vstudio/>
- [27] *Editor PSPad – freeware HTML editor, PHP editor, XHTML, JavaScript, ASP, Perl, C, HEX editor* [online]. 2013 [cit. 2013-04-14]. Dostupné z: <http://www.pspad.com/>
- [28] *Mozilla — Home of the Mozilla Project — mozilla.org* [online]. 2013 [cit. 2013-04-14]. Dostupné z: <http://www.mozilla.org/en-US/>
- [29] Download PostgreSQL | EnterpriseDB. *EnterpriseDB | The Enterprise PostgreSQL Company* [online]. 2013 [cit. 2013-04-20]. Dostupné z: <http://www.enterprisedb.com/products-services-training/pgdownload#windows>
- [30] *Welcome to The Apache Software Foundation!* [online]. 2013 [cit. 2013-04-20]. Dostupné z: <http://www.apache.org/>
- [31] PROKOPOVÁ, Zdenka. Přístup k databázím z WWW [cdrom]. [cit. 28.4.2013].
- [32] *Permutation Algorithms Using Iteration and the Base-N-Odometer Model (Without Recursion)* [online]. 2013 [cit. 2013-04-28]. Dostupné z: <http://www.quickperm.org/>
- [33] BROADHURST, Martin. Combinatorial Algorithms. *Martin Broadhurst – Software Developer* [online]. 2010 [cit. 2013-04-28]. Dostupné z: <http://www.martinbroadhurst.com/combinatorial-algorithms.html>
- [34] OpenSSL: The Open Source toolkit for SSL/TLS [online]. 2013 [cit. 2013-05-06]. Dostupné z: <http://www.openssl.org/>
- [35] Apache2 SSL on Windows. Neil Obremski's Stuff [online]. 2006 [cit. 2013-05-06]. Dostupné z: <http://www.neilstuff.com/apache/apache2-ssl-windows.htm>
- [36] COLLI - CID International. *COLLI - CID International* [online]. 2011 [cit. 2013-05-14]. Dostupné z: <http://www.cid.cz/page/colli-5>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

MIT Massachusetts Institute of Technology.

HTML HyperText Markup Language.

DOM Document Object Model.

SQL Structured Query Language.

DBMS Database management system.

OS Operační systém.

CSS Cascading Style Sheets.

SP3 Service Pack 3.

AJAX Asynchronous JavaScript and XML.

SSL Secure Sockets Layer.

TLS Transport Layer Security.

HTTP Hypertext Transfer Protocol Secure.

S

SEZNAM OBRÁZKŮ

Obr. 1. Okno programu TourSolver.....	12
Obr. 2. Okno programu PLANTOUR.....	13
Obr. 3. Spolupráce systému COLLI se systémem PLANTOUR.....	14
Obr. 4. Webová aplikace Google Maps.....	16
Obr. 5. Omezení freewarové služby Google Maps API.....	17
Obr. 6. Omezení freewarové služby Google Maps API.....	17
Obr. 7. Omezení freewarové služby Google Maps API.....	18
Obr. 8. Webová aplikace Mapy.cz.....	18
Obr. 9. Webová aplikace OpenLayers	20
Obr. 10. Webová aplikace OpenStreetMap	21
Obr. 11. Schema obecného fungování aplikace.....	25
Obr. 12. Okno programu Microsoft Visual Studio 2010.....	29
Obr. 13. Okno programu PSPad 4.5.4.....	30
Obr. 14. Okno webového prohlížeče Mozilla Firefox 19.0.2.....	31
Obr. 15. Problém synchronizace dat při jejich načítání a zpracování.....	34
Obr. 16. Vývojový diagram testování odpovědi na dotaz.....	35
Obr. 17. Obecné schema dotazování databáze MySQL [31]	36
Obr. 18. Příklad sestavených dat z tabulek databáze předávaných aplikaci.....	38
Obr. 19. Uživatelský interface aplikace	40
Obr. 20. Příklad složené matice vzdáleností.....	43
Obr. 21. Funkce algoritmu permutace	48
Obr. 22. Funkce algoritmu kombinace s permutací.....	50
Obr. 23. Ovládání aplikace - spuštění	53
Obr. 24. Ovládání aplikace - uživatelský interface	54
Obr. 25. Ovládání aplikace - načtení dat z databáze.....	54
Obr. 26. Ovládání aplikace - ruční zadání zákazníka.....	55
Obr. 27. Ovládání aplikace - uložení zákazníka	56
Obr. 28. Ovládání aplikace - vložení přesouvateľného waypointu.....	57
Obr. 29. Ovládání aplikace - dotazování geokodéru	58
Obr. 30. Ovládání aplikace - zobrazení výsledné trasy	59
Obr. 31. Modelový příklad použití – sídlo společnosti XYZ.....	61

Obr. 32. Modelový příklad použití – zákazníci	62
Obr. 33. Modelový příklad použití – výsledek č. 1	62
Obr. 34. Modelový příklad použití – výsledek č. 2	63

SEZNAM TABULEK

Tab. 1. Příklad tabulky „zakaznici“	36
Tab. 2. Příklad tabulky „adresy“	37
Tab. 3. Příklad tabulky „ceny“	37

SEZNAM PŘÍLOH

Příloha PI: Ukázka kódu uživatelského interface aplikace.

Příloha PII: Ukázka kódu výpočetního algoritmu.

Příloha PIII: CD-ROM obsahující vlastní práci a zdrojové kódy.

PŘÍLOHA P I: UKÁZKA KÓDU UŽIVATELSKÉHO INTERFACE APLIKACE

```
function VypisPoleWaypointuSouradnice() {
    PromazaniVypsaneSouradnice();

    for (var i = 0; i < poleWaypointuSouradnice.length; i++) {
        $('#vypsaneSouradnice').append(i + ". položka: " +
        poleWaypointuSouradnice[i] + "\n" + "<br>");
    }
}

function VypisPoleWaypointuSouradniceJakoFloat() {
    PromazaniVypsaneSouradnice();

    for (var i = 0; i < poleWaypointuSouradniceJakoFloat.length; i++) {
        $('#vypsaneSouradnice').append(i + ". položka: " +
        poleWaypointuSouradniceJakoFloat[i] + "\n" + "<br>");
    }
}

function PridejDoPoleCen() {

    if(switchNacitaniDatabaze == 1) { // pokud je prave nacistana databaze

        if(poleCen.length < 1) { // kdyz nebude zadane startovni mesto
            var cena = 0;

            poleCen.push(cena);
        }
        else {
            var cena = parseInt( dbCeny[ (polozkaDb-1) ] );

            poleCen.push(cena);
        }
        switchA2 = 1;
    }

    else{
        if( $('#vstupCena').val() == 0 ) {

            if(poleCen.length > 0) { // nebude upozornovat u prvniho
                waypointu/polozky - start
                alert("POZOR! Cena je 0.");
            }
        }

        var cena = parseInt( $('#vstupCena').val() );

        poleCen.push(cena);

        $('#vstupCena').val(0); // vynulovani ceny
    }

    VypisPoleCen();
}

function VypisPoleCen() {

    $('#span', '#vypsanePoleCen').empty().remove();
```

PŘÍLOHA P II: UKÁZKA KÓDU VÝPOČETNÍHO ALGORITMU

```
int Vytvor1DPoleProVstup(int pocetMest) {
    Pole1DProVstup = (int *) malloc( (pocetMest * pocetMest) * (sizeof(int
*)) ); // alokovani pameti

    if (Pole1DProVstup == NULL) { // kontrola, jestli alokace probehla v
        poradku
        cout << endl << "NEDOSTATEK PAMETI!";
        return 1;
    }

    for (int i=0; i<(pocetMest * pocetMest); i++) {
        Pole1DProVstup[i] = 0; // pocatecni inicializace pole (vsechny
        prvky tohoto pole naplni nulami)
    }
}

int Vytvor2DPole(int pocetMest) {
    cities = (int **) malloc(pocetMest * (sizeof(int *))) ; // vyska pole

    if (cities == NULL) { // kontrola, jestli alokace probehla v poradku
        cout << endl << "NEDOSTATEK PAMETI!";
        return 1;
    }

    for (int i=0; i<pocetMest; i++) { // alokace pameti pro sirku pole
        cities[i] = (int *) malloc(pocetMest * (sizeof(int)) ); //sirka
        pole

        if (cities[i] == NULL) { // kontrola, jestli alokace probehla
            v poradku
            cout << endl << "NEDOSTATEK PAMETI!";
            return 1;
        }
    }
}

void SlozMaticiVzdalenosti() {
    // poskladani do formatu matice vzdalenosti
    int k = 0;

    for (int i=0; i<pocetMest; i++) {
        for (int j=0; j<pocetMest; j++) {

            cities[i][j] = Pole1DProVstup[k];
            k++;
        }
    }
}
```