

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Integrace mapových podkladů do ERP systému
Bakalářská práce

Autor: Michal Renčín
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Pavel Kříž, Ph.D.

Prohlášení:

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a že jsem uvedl všechny použité informační zdroje a literaturu, které jsou citovány na konci v seznamu použité literatury.

V Hradci Králové dne 27.4.2015

Michal Renčín

Poděkování:

Chtěl bych vyjádřit vřelý dík vedoucímu bakalářské práce, Ing. Pavlu Křížovi, Ph.D, za metodické vedení práce, odborné rady a připomínky v průběhu jejího psaní. Také bych chtěl poděkovat za podporu své rodině a přátelům.

ANOTACE

Tato bakalářská práce pojednává o možnosti využití internetových mapových pokladů pro implementování do ERP systému. Jsou zde popsány základní možnosti jednotlivých mapových podkladů. Čtenář bude mít po prostudování textu základní pohled na tuto problematiku.

Hlavním cílem práce je provedení analýzy zvolené problematiky a návrhu aplikace, na které jsou vysvětleny všechny funkční vlastnosti mapových podkladů. Dále je vysvětleno využívání softwarové platformy JavaFX pro zobrazování mapových podkladů.

ANOTATION

This thesis discusses the possibility of using Web maps for implementing to the ERP system. There are described basic capabilities of individual maps. After reading this text the reader gets basic view on this issue.

The main objective of this work is the analysis of the chosen problem and design applications, which are explained all the features of maps. The following explains the use JavaFX software platform for displaying maps.

Obsah

1	Úvod	1
2	Cíl práce	3
3	ERP systémy	4
3.1	Systém OrsoftOpen	4
3.1.1	Funkce systému OrsoftOpen	5
4	Mapové podklady	8
4.1	Google mapy	8
4.1.1	The Google Distance Matrix API	9
4.1.2	The Google Directions API	11
4.1.3	JavaScript API v3	14
4.1.4	Google Places API Web Service	16
4.2	Mapy.cz	18
4.2.1	Mapy API verze 4.8	18
4.3	Here.com	20
4.3.1	Matrix Routing API	21
4.3.2	JavaScript API 3.0	23
4.3.3	HERE Routing API	24
5	Analýza a návrh	26
5.1	Verze aplikace evidence jízdnic cest pro uživatele bez zobrazování mapy	27
5.2	Verze aplikace evidence jízdnic cest pro uživatele se zobrazením mapy	28
6	Technologie využívaná při vývoji aplikace pro evidenci jízdnic cest	30
6.1	Programovací jazyk Java	31
6.2	Verze aplikace evidence jízdnic cest pro uživatele bez zobrazování mapy	31
6.2.1	XML	32
6.2.2	JSON	32
6.2.3	Open Source knihovna Dom4j	32
6.3	Verze aplikace evidence jízdnic cest pro uživatele se zobrazováním mapy	33
6.3.1	Služba JavaFX	33
6.3.2	Služba JavaScript	33
7	Implementace aplikace pro evidenci jízdnic cest do ERP systému	35
7.1	Třída MainFrame	35
7.1.1	Třída SwingWorker	37
7.2	Třídy RouteParser a RouteData	37
7.3	Třída Browser	39

8	Výsledky a doporučení pro vývoj aplikace pro evidenci jízdních cest.....	42
8.1	Testování aplikace pro evidenci jízdních cest	42
8.2	Výsledek aplikace pro evidenci jízdních cest.....	43
8.3	Další vývoj aplikace pro evidenci jízdních cest	44
8.4	Omezení vyvíjené aplikace pro evidenci jízdních cest	45
9	Závěr.....	46
10	Seznam použité literatury	48

1 Úvod

V dnešní době hrají informační technologie nezastupitelnou roli při řízení řady procesů ve firmách. Proto jsou čím dál častěji požadovány komplexní programy pro snadné řízení. Vhodným řešením pro podporu jednotlivých firemních procesů jsou tzv. enterprise resource planning systems (ERP systémy) neboli podnikové informační systémy. ERP systémy jsou aplikovány např. při finančním účetnictví, řízení lidských zdrojů nebo manažerských funkcí typu Business Intelligence.

Cílem všech podniků je zvýšení ziskovosti, která může být dosažena např. prostřednictvím snižování nákladů. Nedílnou součástí finančních výdajů většiny firem jsou i náklady na dopravu. V současnosti řada podniků poskytuje svým zaměstnancům služební vozidla nejen pro firemní, ale i soukromé účely. Proto je dobré monitorovat jejich využívání. Při využívání firemního vozu zaměstnancem na služební cestě, je zaměstnanec povinen vyplnit cestovní protokol, v němž jsou evidovány informace o ujeté trase (především vzdálenost) a výchozí a cílová adresa služební cesty. Je ale také možné, že zaměstnanec využije podnikový automobil pro osobní účely.

Firma Ortex s. r. o. v Hradci Králové je společnost s dlouholetou tradicí zabývající IT řešením a vývojem informačních systémů pro organizace. Mezi základní portfolia této firmy patří také ERP produkty. Z tohoto důvodu se vedení firmy rozhodlo do vyvíjeného ERP systému OrsoftOpen zahrnout i implementaci mapových podkladů pro kontrolu záznamů v deníku jízd. Existuje řada poskytovatelů mapových podkladů, mezi které se řadí Google maps společnosti Google nebo Mapy.cz společnosti Seznam.cz. Zmíněné společnosti zajišťují uživatelům základní funkce mapových podkladů zdarma a jejich poskytovatelé nabízí nejen různé možnosti zobrazení a hledání zájmových adres, ale také nabízí řadu rozšiřujících funkcí pro vhodné využívání map. Hlavní ideou je pomocí výchozí a koncové adresy jízdni cesty zjistit pravděpodobnou trasu, ze které by bylo možné určit čas a vzdálenost pro ujetí této trasy. Výsledná vzdálenost by se následně porovнала se vzdáleností uvedenou zaměstnancem, která se zároveň ukazuje na tachometru automobilu. V praxi je nutné počítat s případnou odchylkou, kterou nedokáží mapové podklady zjistit a která může vzniknout při nečekané objízdě trasy. S ohledem na přičtení této odchylky lze s rezervou určit, zda zaměstnanec dodržel plánovanou trasu, anebo využil firemní vozidlo i pro soukromé účely. Někdy musí zaměstnanec na služební cestě navštívit více míst. Proto je vhodné rozšířit aplikaci i o tzv. průjezdní body, resp. místa, která musí

zaměstnanec nutně projet na své trase a které se započítávají do výsledné trasy. Zanesení průjezdních bodů do systému je výhodnější nejen při měření složitějších tras, ale i při již zmiňovaných objízdných trasách. Na základě parametrů zadaných zaměstnancem je vygenerována trasa. Eventuálně by zaměstnanec mohl změnit umístění jednotlivých průjezdních bodů pouhým posunutím značky znázorňující požadovanou adresu. V případě změny jízdny trasy by aplikace ihned přepočítala výsledek, který je okamžitě zobrazen zaměstnanci.

Vyvíjená aplikace sloužící pro evidenci služební cesty zaměstnance v rámci systému OrsoftOpen firmy Ortex s r. o., záznam její trasy, ujeté vzdálenosti a doba jízdy by mohla přispět k větší zodpovědnosti zaměstnanců a zejména zefektivnění fungování tohoto podniku.

2 Cíl práce

Cílem bakalářské práce je vytvoření prototypu aplikace pro záznam jízdnicích cest pro budoucí implementování do ERP systému OrsoftOpen firmy Ortex s. r. o. Vytvořená aplikace by měla využívat všechny poskytnuté funkce mapových podkladů některého z internetových poskytovatelů. Navíc by měla vyhledat trasu dle zadaných adres a zvolených možností (např. vyhnout se dálnicím, mýtným bránám, trajektům). Výstupem aplikace by měly být údaje zobrazené uživateli – výsledná vzdálenost nalezené trasy a doba jízdy. V případě potřeby může být uživateli zobrazena interaktivní mapa s vyhledanou trasou, kterou by mohl uživatel upravovat natolik, aby odpovídala skutečnosti. Možnou výhodou by mohlo být využívání „našeptávače“, který nabízí většina poskytovatelů mapových podkladů a který by uživateli nabídl adresy podle již předem zadaného textu. Aplikace by měla být v podniku užívající systém OrsoftOpen určena zaměstnanci zodpovídajícímu za zpracování knih jízd při využití služebních automobilů.

3 ERP systémy

Enterprise resource planning system (ERP systém) je informační systém, který integruje a automatizuje velké množství firemních procesů souvisejících s činnostmi firmy. Převážně se jedná o výrobu, logistiku, distribuci, správu majetku, prodej, fakturaci a účetnictví. Do ERP jsou řazeny aplikace, které představují softwarové řešení k řízení firemních dat a pomáhají plánovat celý logistický řetězec. ERP může být chápán jako hotový software, díky kterému může podnik sdílet podniková data a umožňuje jejich dostupnost v reálném čase [1].

Podnikové informační systémy typu ERP přímo zahrnují nebo jsou modulárně rozšiřitelné o [2]:

- MIS (management information system) – jedná o manažerskou nadstavbu podporující vrcholové řízení firmy
- SCM (supply chain management) – řešení pro řízení dodavatelsko-odběratelských řetězců
- CRM (customer relationship management) – aplikace sloužící k řízení vztahů se zákazníky

3.1 Systém OrsoftOpen

Již od r. 1990 firma Ortex s. r. o. v Hradci Králové vyvinula a rozvíjí ekonomický informační systém ERP typu „vše v jednom“ – OrsoftOpen. Jde o systém nabízející širokou škálu modulů podporující různé firemní procesy podle potřeb uživatele (např. finanční účetnictví a správa majetku). Informační systém OrsoftOpen je určený k řízení výrobních i obchodních aktivit malých a středních podniků. V současné době jej využívá více než sto společností s českými i zahraničními majiteli. OrsoftOpen je dostupný i pro neziskový sektor. Je v nabídce i ve verzi Orsoft RADNICE, která je určena městům a obcím. Uživatelé systému mají od vývojářů zajištěnou trvalou podporu prostřednictvím hotline, konzultací a vzdálené pomoci. Systém OrsoftOpen je neustále vyvíjen a aktualizován, a to vždy v souladu s aktuální legislativou [3].

3.1.1 Funkce systému OrsoftOpen

Jak již bylo řečeno, systém OrsoftOpen se dělí na několik dílčích modulů. Výchozí je základní část, do které spadá administrace, technologie a kmenová data. Základní část je provázána s oblastmi logistiky, výroby, lidských zdrojů, správy majetku a finančního účetnictví. Finanční účetnictví může dále spolupracovat s moduly logistiky a lidských zdrojů prostřednictvím dotahování potřebných informací. Nadřazený finančnímu účetnictví je controlling určený účetním. Nejvyšším modulem je Business Intelligence, který slouží manažerským účelům (viz obr. 3.1) [3].



Obrázek 3.1 - globální architektura systému Orsoft Open [3].

Základní část systému OrsoftOpen je dále rozdělena na dvě dílčí součásti, kam patří kmenová data a administrace. Kmenová data systému zahrnují společné datové množiny včetně jejich obsluhy. Jedná se hlavně o číselníky, adresáře a doklady. Dále do základní části patří administrace, kde jsou zařazeny všechny úlohy systému OrsoftOpen, které mají přímou vazbu na technologickou architekturu, provozní prostředí a konfiguraci systému. Jedná se zejména o přístupová práva, elektronické dokumenty a nástroje administrátora [3]

Se základní částí je provázána oblast logistiky. Funkcí logistiky je nákup a obstarávání zásob, jejich skladování a doprava. Nákup je určen pro podporu řízení a automatizaci

zpracování činností souvisejících s objednáváním a nákupem materiálů, zboží a služeb. Skladování umožňuje vedení skladové evidence pro zásobovací, prodejní i konsignační sklady. Doprava poskytuje nástroje k řízení a organizaci dopravy ve firmě a umožňuje sledování a vyhodnocování veškerých nákladů souvisejících s dopravou. Zabývá se především rozvozem zboží formou pravidelných linek, ale může sloužit i pro firmy, kde jsou používána pouze referentská vozidla [3]

Na základní část systému OrsoftOpen navazuje i oblast výroby, která slouží ke sledování a vyhodnocování vlastních výrobků. Vychází ze zjednodušených principů výrobní a controllingové činnosti se zaměřením na plánování a vyhodnocování nákladů na výrobu [3]

Se základní částí systému souvisí i lidské zdroje. Jedná se o modul, jehož součástí jsou kategorie řízení lidských zdrojů, personalistika a mzdy. Řízení lidských zdrojů zahrnuje nástroje pro personální řízení a je určen pro střední a vrcholový management, personalisty a vedoucí. Modul personalistika je určen pro podporu důležitých personálních rozhodnutí a systém pro podporu personální evidence a administrativy. Kategorie mzdy slouží pro výpočet mezd zaměstnanců a spolupracuje s podsystémem personalistika. Oba podsystémy využívají společně část datové základny [3]

Správa majetku je také provázána se základní částí systému OrsoftOpen. Jde o modul určený pro správnou dokumentaci majetku společnosti. Detailně řeší všechny požadavky české legislativy včetně zrychlených odpisů a technického zhodnocení. Jeho součástí je podsystém majetek a investice. Základem podsystému majetek je evidence dlouhodobého majetku [3].

Součástí systému OrsoftOpen je i finanční účetnictví, které je založeno na tzv. deníku dokladů. Těžištěm finančního účetnictví jsou tři deníky, do kterých jsou zaznamenávány doklady podle typu. Deník faktur obsahuje vydané/došlé faktury typu tuzemská/zahraniční faktura a také daňové doklady a ostatní ustanovení dle mzdových předpisů. Deník plateb obsahuje informace o příjmech/výdajích na pokladních nebo bankovních účtech v tuzemské nebo cizí měně. Deník interních dokladů obsahuje ostatní účetní doklady pořízené rukou nebo importované z jiných podsystémů nebo systémů [3]

Na oblast finančního účetnictví navazuje oblast controllingu. Pomocí tohoto modulu je možné řešit standardní úlohy nákladového controllingu. Je umožněno detailní sledování nákladů pomocí deseti nositelů z celkových třiceti možných. Mezi základních deset nositelů patří zákazník, zakázka, nadřizená zakázka, typ zakázky, účet, položka kalkulace, cash-flow zakázky, výrobek, skupina výrobků a množství pro řešení úloh výkonové

orientace. Využití nositelů si klade za cíl minimalizaci pořizovacích výdajů. Uživatel pořizuje tři až čtyři nositele nákladů, ostatní jsou generovány. K základním nositelům patří účet, útvar, zakázka a výrobek [3]

Nejvyšším modulem systému OrsoftOpen je Business Intelligence. Jde o souhrn postupů a softwarových nástrojů, které pomáhají uživatelům zpracovávat a vyhodnocovat velké objemy dat a získávat z nich důležité informace pro podporu rozhodování. Jeho aplikací dochází ke zlepšování výsledků podniku. Pomocí Business Intelligence je také možno odhalit závislosti procesů a událostí či silné a slabé stránky těchto procesů [3]

Informační systém OrsoftOpen firmy Ortex s r. o. je propracovaný celek určený pro podniky za účelem zefektivnění výrobních a obchodních procesů usnadňující pozici této firmy na trhu.

4 Mapové podklady

Nejdůležitější součástí vyvíjeného prototypu aplikace pro evidenci služebních cest jsou mapové podklady. V dnešní době existuje mnoho internetových poskytovatelů mapových podkladů nabízející podobné služby. Podmínkou vyvíjení aplikace je poskytnutí mapových podkladů bezplatně. Další nutnou podmínkou je komunikace uživatele s mapovými podklady prostřednictvím aplikace. Je tedy potřeba najít takové mapové podklady, které poskytují jejich využívání prostřednictvím připraveného rozhraní – tzv. Application Programming Interface (API).

Dle výše zmíněných kritérií bylo vybráno několik poskytovatelů mapových podkladů. Patří sem Mapy.cz, Google.com, Here.com, Yahoo.com, MapQues.com a OpenStreetMap.com. Mezi kandidáty je i jeden český poskytovatel mapových podkladů Seznam.cz. Pro zjednodušení nebudou hodnoceni všichni vybraní poskytovatele, ale pouze tři nejvhodnější, a to Google.com, Mapy.cz a Here.com. Jak již bylo zmíněno, všechny vyjmenované společnosti poskytují mapové podklady zdarma a zároveň nabízejí vlastní API rozhraní.

4.1 Google mapy

Google mapy jsou internetovou službou pro komerční i nekomerční využití, která je poskytovaná společností Google.com. Google mapy poskytují zdarma detailní mapy silnic, plánování trasy pěšky, automobilem, jízdou na kole nebo hromadnou dopravou. Také poskytují polohu a informace o mnoha firmách v řadě zemí po celém světě. Mapy Google používají tzv. Mercatorovo zobrazení, které nezobrazuje oblasti okolo zemských pólů. Tato vlastnost ale nebrání použití mapových podkladů pro vyvíjenou aplikaci.

Google mapy poskytují mnoho rozličných služeb pro měření vzdáleností a zhodnocení přibližné doby jízdy. Jednou z možností je pracovat s mapou na „pozadí“, kdy nelze vidět samotnou mapu, ale pouze získat výsledná data v podobě vzdálenosti trasy a času jízdy. Mezi tyto služby patří *The Google Distance Matrix API* a *The Google Directions API*. Další možností je kompletní zobrazení mapy, které umožní vykreslit jak samotná místa, tak i naplánovanou trasu mezi dvěma místy. Navíc na sledované trase lze přidávat i další průjezdná místa. Možná je i dodatečná úprava již zobrazených míst na mapě pomocí jednoduchého uchopení a následného posunutí prostřednictvím počítačové myši. Službu je

možné využívat prostřednictvím programu *JavaScript API v3*. Google mapy poskytují vhodnou službu The Google Places API Web Service pro vytvoření tzv. „našeptavače“, který uživateli nabízí vyhledané možnosti.

4.1.1 The Google Distance Matrix API

Služba *The Google Distance Matrix API* umožňuje pracovat s mapou na „pozadí“. To znamená, že nelze vidět mapu, ale pouze získat informace o výsledné vzdálenosti a době jízdy na základě zadání vstupních parametrů. Jde o službu poskytující výpočet vzdálenosti a času pro měření mezi více místy najednou. Pracuje na principu vytvoření matice, ve které jsou předávány informace o výchozích a cílových místech. Po vytvoření matice služba vypočítá a poskytne informace, které se skládají z jednotlivých řádků. Na každém řádku je informace obsahující čas a vzdálenost pro každou z dvojic. Nevýhodou této služby je, že neposkytuje detailní informace o nalezené trase [4].

Zadávání požadavku je prostřednictvím jednoduchého HTML dotazu, ve kterém jsou uváděny jednotlivé parametry – výchozí a cílová adresa hledané cesty. Základní tvar adresy URL: `https://maps.googleapis.com/maps/api/distancematrix/output?parameters&key=API_KEY`. Proměnná *output* v tomto dotazu říká, v jakém formátu mají být výsledné informace předány. Na výběr jsou formáty XML a JSON. V proměnné *parameters* musí být vyplněny základní informace. Mezi ně patří *origins* a *destinations*, které obsahují údaje o počátečních a cílových adresách. Adresy se zapisují dvěma způsoby. Buď lze zadat adresu v podobě textu, anebo lze adresu zadat prostřednictvím zeměpisné délky a šířky. Při zadávání více počátečních nebo cílových adres je potřeba jednotlivé adresy oddělit znakem „|“ (např. zadání dvou počátečních/cílových adres – Hradec Králové a Pardubice: *origins=Hradec+kralove|50.039,15.768*) [4]. Do URL adresy lze také přidat další parametry, které by uživateli usnadnily hledání trasy. Mezi tyto parametry patří [4].

- *mode* – Upřesnění, jakým dopravním prostředkem cestovat.
- *language* – Uvádí, v jakém jazyce jsou výsledky požadovány.
- *avoid* – Uživatel může pomocí tohoto parametru zvolit, zda se chce při cestě vyhnout dálnicím, mýtným bránám nebo trajektům.

Podstatnou ale nikoliv nezbytnou částí URL adresy je identifikační kód, tzv. *API_KEY* tzv. *API_KEY*, který je generován každému uživateli individuálně. Slouží převážně

ke sběru informací o používání dané API. Bez použití API_KEY uživatel může zadat pouze 100 elementů na jeden požadavek, 100 elementů za 10 sekund a 2500 elementů za 24 hodin. Počet elementů v požadavku se rovná počtu počátečních adres násobeno počtem cílových adres. Při využívání API_KEY vzroste počet elementů na jeden požadavek na 650, 1000 elementů za 10 sekund a 100 000 elementů za 24 hodin [4].

```
<DistanceMatrixResponse>
  <status>OK</status>
  <origin_address>Pardubice, Česká republika
</origin_address>
  <origin_address>Praha, Česká republika
</origin_address>
  <destination_address>Hradec Králové, Česká republika
</destination_address>
  <row>
    <element>
      <status>OK</status>
      <duration>
        <value>1424</value>
        <text>24 min</text>
      </duration>
      <distance>
        <value>26176</value>
        <text>26,2 km</text>
      </distance>
    </element>
  </row>
  <row>
    <element>
      <status>OK</status>
      <duration>
        <value>4223</value>
        <text>1 hodin, 10 minut</text>
      </duration>
    </element>
  </row>
</DistanceMatrixResponse>
```



```

    <distance>
      <value>112945</value>
      <text>113 km</text>
    </distance>
  </element>
</row>
</DistanceMatrixResponse>

```

Ukázka kódu 4.1.1 XML Dokument ve službě Distance Matrix API společnosti Google – výsledek pro výchozí adresy Praha a Pardubice, Hradec Králové je cílová adresa.

Názorným příkladem je ukázka kódu 4.1.1. V elementu *status* jsou informace o nalezené trase. V tomto případě *OK* znamená nalezení výsledku. V elementech *origin* a *destination* jsou názvy zadaných adres. V elementu *row* jsou dva elementy informující o vyhledané vzdálenosti mezi dvěma adresami. V případě zadání více výchozích adres výsledek obsahuje více elementů *row*, přičemž počet elementů *row* se rovná počtu zadaných výchozích adres. Element *status* v elementu *row* udává, zda byla nalezena cesta pro dvě konkrétní adresy. Počet elementů *element* je přímo úměrný počtu zadaných cílových adres. Pro jednu výchozí adresu jsou vyhledány vzdálenosti se všemi cílovými adresami. V elementu *duration* jsou dva elementy, ve kterých jsou informace o odhadu času potřebného pro ujetí trasy. Časový údaj může být uveden buď v číselné hodnotě v sekundách, anebo v minutách. V elementu *distance* jsou také dva elementy obsahující informaci o vzdálenosti mezi adresami. Vzdálenost je uváděna jako číselný údaj v metrech, nebo jako text v kilometrech [4].

Google Distance Matrix API je služba, která poskytuje uživateli informace o cestovní vzdálenosti a času, a je tedy jednou z možností využití této služby pro tvorbu aplikace vyvíjené v rámci systému OrsoftOpen společnosti Ortex s r. o.

4.1.2 The Google Directions API

Další službou společnosti Google.com je The Google Directions API. Stejně jako v případě The Google Distance Matrix API umožňuje pracovat s mapou na „pozadí“.

Jde o službu umožňující měření vzdáleností a času potřebných pro ujetí určité vzdálenosti. Vzdálenost je měřena mezi dvěma zadanými adresami, které lze v případě

potřeby rozšířit o další průjezdná místa. Nicméně na rozdíl od *The Google Distance Matrix API* je služba *The Google Directions API* schopna poskytovat podrobnější informace o trase. Podává totiž podobné informace jako GPS (např. radí kde a kdy zatočit). Na druhou stranu ale služba provádí výpočet trasy pouze podle „offline“ dat, nevyhledává tedy trasu podle aktuálních informací a situací v silniční síti [5].

Zadávání požadavku v této službě je podobné jako v případě *The Google Distance Matrix API*, a to prostřednictvím HTML dotazu, ve kterém se zadávají informace, pro co nejpřesnější výsledek. Základní tvar adresy URL je: `https://maps.googleapis.com/maps/api/directions /output?parameters&key=API_KEY`. Stejně jako v předchozí službě je na výběr ze dvou výstupních formátů XML a JSON. Dále je nutné vyplnit informace o počáteční a cílové adrese. Případně je potřeba pro přesnější výsledek zadat další parametry (např. typ vozidla). Stejně jako u předchozí služby společnosti Google.com je počáteční a cílová adresa zadávána buď jako text, nebo pomocí zeměpisné délky a šířky do parametru *origin* a *destination*. Navíc přibyl parametr *waypoints*, který může obsahovat informace o průjezdných bodech a které se zadávají obdobně jako počáteční nebo koncová adresa. Je ale nutné jednotlivé průjezdní body oddělit znakem „|“ (např. `waypoints=chrudim/49.176042,16.612293` – zde jsou znázorněny průjezdní body Chrudim a Brno). Uživatel také může zadáním vstupních informací ovlivňovat trasu jízdy (např. vyhýbání se dálnicím, mýtným bránám a trajektům atd.). Nezbytným doporučením pro uživatele je nastavení jazyka, ve kterém mají být informace poskytnuty, pomocí parametru *language*. Zásadní vliv na výslednou trasu má i volba dopravního prostředku. Na výběr je automobil, hromadná doprava (autobus, vlak, tramvaj, metro), jízda na kole nebo chůze pěšky [5].

Podobně jako u služby *The Google Distance Matrix API* jsou zde také využívány služby *API_KEY*. Není-li použit *API_KEY*, společnost Google.com dovoluje využívání služby *The Google Directions API* s omezením na 2500 požadavků během 24 hodin. Navíc při zadávání trasy je dovoleno zadat maximálně osm průjezdních bodů. Poslední omezení je na 2 požadavky během jedné sekundy. Ovšem bude-li použit klíč *API_KEY*, počet požadavků se zvýší na 100 000 za jeden den. I celkový počet průjezdních bodů se zvýší na 23 možných a je možno odeslat 10 požadavků během jedné sekundy [5].

```

<DirectionsResponse>
  <status>OK</status>
  <route>
    <summary>silnici 37</summary>
    <leg>
      <step>...</step>
      ...
      <step>...</step>
      <duration>
        <value>1280</value>
        <text>21 min</text>
      </duration>
      <distance>
        <value>25423</value>
        <text>25,4 km</text>
      </distance>
      <start_location>
        <lat>50.2103543</lat>
        <lng>15.8252088</lng>
      </start_location>
      <end_location>
        <lat>50.0342995</lat>
        <lng>15.7811997</lng>
      </end_location>
      <start_address>Hradec Králové, Česká republika
      </start_address>
      <end_address>Pardubice, Česká republika
      </end_address>
    </leg>
  </route>
</DirectionsResponse>

```

Ukázka kódu 4.1.2 XML Dokument ve službě Directions API společnosti Google – výsledek pro adresy Hradec Králové a Pardubice

V ukázce kódu 4.1.2 je ve zkrácené podobě znázorněn XML dokument, který je vrácen jako výsledek pro zadané adresy Hradec Králové a Pardubice. Na rozdíl od služby The Google Distance Matrix API je zde poskytována informace o trase v jednotlivých elementech *step*. Element *distance* obsahuje informaci o vzdálenosti v metrech a kilometrech. V elementu *duration* je uložen časový údaj. Dále XML dokument obsahuje elementy *start* a *end_location*, které obsahují souřadnice v zeměpisné šířce a délce. Elementy *start* a *end_address* obsahují adresy, pro které byla trasa nalezena.

Jak je vidět z uvedených příkladů ukázek kódu 4.1.1 a 4.1.2 jedná se o podobné služby. Pro plánované možnosti aplikace se lépe hodí The Google Directions API, kdy je trasa plánována přímo pro konkrétní výchozí a cílovou adresu.

4.1.3 JavaScript API v3

Zatímco předchozí dvě popsané služby společnosti Google.com umožňují pracovat s mapovými podklady pouze „na pozadí“, kdy lze získat pouze informaci o vzdálenosti a čase, další služba této společnosti, JavaScript API v3, nabízí kompletní zobrazení mapy. To umožní vykreslit jak samotná místa, tak i plánování trasy mezi dvěma místy. Navíc na sledované trase lze přidávat i další průjezdní místa. Možná je i dodatečná úprava již zobrazených míst na mapě pomocí jednoduchého uchopení a následného posunutí prostřednictvím počítačové myši [6]

JavaScript API v3 je služba umožňující vložení plnohodnotného mapového podkladu na internetové stránky třetích stran. Výsledkem je zobrazení totožné mapy, jež má společnost Google.com na svých internetových stránkách (maps.google.com). Mapa je zobrazena prostřednictvím programu JavaScript, který se nachází v těle HTML stránky. Ve službě JavaScript API v3 je možno také provádět plánování tras pěšky, na kole, automobilem či hromadnou dopravou [6].

Při použití JavaScript API v3 je nutné načtení všech definic a symbolů. O načítání se stará URL odkaz v tagu *script*, který odkazuje na umístění souboru. Společnost Google.com se snaží o vysokou míru zabezpečení na internetu, proto doporučuje využívat protokol HTTPS. Je-li potřeba použít protokol http, je nutné změnit URL adresu na <http://maps.googleapis.com/>, popř. pro uživatele z Číny na <http://maps.google.cn> [6].

```

<html>
  <head>
    <scripttype="text/javascript"src="https://maps.googleapis
.com/maps/api/js?key=API_KEY"></script>

```

Někdy je potřeba, aby byl program JavaScript načten až po načtení internetové stránky. Pro tyto účely slouží nastavení *window.onload = loadScript*;. Jde o metodu, která má být volána až po načtení stránky. V tomto případě jde o metodu, která zahájí zpracování JavaScriptu. Obdobně lze přidáním parametru *callback=initialize* do adresy URL říci, že hlavní metoda JavaScriptu *initialize()* má být volána až po úplném načtení API. Pro zobrazení mapy je v HTML stránce vytvořen element `div` `<div id="map-canvas" style="width: 100%; height: 100%"></div>`, do kterého se mapa vkládá [6].

Před začátkem načítání mapy je nutné nadefinovat proměnnou *mapOptions*, díky níž jsou nastaveny veškeré informace zobrazování tras. V proměnné *mapOptions* definujeme základní parametry *center* a *zoom*, které uvádí vycentrování a přiblížení mapy. Pomocí příkazu `new google.maps.LatLng(-34.397, 150.644)` vytvoříme objekt třídy *LatLng*, který obsahuje informace o zeměpisné šířce a délce hledané adresy. Důležité je ale zadání ve správném pořadí. Jinou možností je formát zadání: {lat: -34.397, lng: 150.644}. The Google Maps JavaScript API za běhu vytvoří na pozadí nový objekt typu `google.maps.LatLng` [6].

```

var mapOptions = {
  center: new google.maps.LatLng(-34.397, 150.644),
  zoom: 8
};

```

Mapa, jež má být zobrazena v elementu `<div>`, je instancí třídy *Map*, při jejichž vytváření je nutné zadat element, do kterého bude mapa vložena. K tomuto účelu slouží příkaz `document.getElementById("map-canvas")`. Dále musí být předán objekt *mapOptions*, podle kterého je trasa hledána.

```

var map = new google.maps.Map(document.getElementById("map-
canvas", mapOptions);

```

Je nezbytné podotknout, že je služba JavaScript API v3 provozována zdarma při splnění předem stanovených podmínek dané společností Google.com. Konkrétně není-li překročena hranice 25 000 načtení map během jednoho dne a dále během 90 po sobě jdoucích dnů. V opačném případě společnost Google.com zkouší kontaktovat provozovatele a nabídne mu možnosti zpoplatnění za využívání jejích služeb. Nicméně v případě překročení výše uvedených stanovených limitů nehrozí odpojení aplikace od služeb API [6].

4.1.4 Google Places API Web Service

Google Places API je služba podporující vyhledání pravděpodobného textu zadávaným uživatelem. Pomocí jednoduchého HTTP dotazu ve formě URL adresy: <https://maps.googleapis.com/maps/api/place/autocomplete/output?parameters> je předán uživatelem již zadaný text. Zpětná vazba služby ukazuje nejlepší shodu známých adres dle předem zadaného textu [7].

V zadávané adrese URL proměnná *output* udává formát výsledku (XML nebo JSON). Proměnná *parameters* zastupuje hodnoty potřebné pro získání výsledku. Parametr *input* obsahuje zadaný text uživatelem, ke kterému má být adresa vyhledána. Nezbytnou součástí URL adresy je také parametr *key*, který obsahuje hodnotu uživatelského API_KEY, pomocí nichž společnost Google.com monitoruje použití jednotlivých API. V případě překročení předem stanovených limitů jednotlivých API může společnost Google požadovat jejich zpoplatnění [7].

Kromě toho HTTP dotaz může obsahovat mnohem více parametrů, které zpřesňují požadované hledání. To lze uplatnit při psaní specifických textů, kdy lze okruh vyhledávání nějakým způsobem zúžit (např. zadávání adresy, země). Parametr *language* určuje, v jakém jazyce má být výsledek prezentován. Prostřednictvím parametru *types* lze zvolit typ vyhledání, resp. hledání adres, měst nebo zeměpisných souřadnic. Pro zjednodušené hledání lze využít parametr *components*, ve kterém lze nastavit zemi vyhledávání. Příkladem může být *components=country:cz*, kde kód země je tvořen dvěma znaky, odpovídající ISO 3166-1 Alpha2 [7].

```

<AutocompletionResponse>
  <status>OK</status>
  <prediction>
    <description>Labský Palouk, Pardubice, Česká
      republika </description>
  </prediction>
  <prediction>
    <description>Labská kotlina, Hradec Králové, Česká
      republika </description>
  </prediction>
  <prediction>
    <description>Labská Louka, Hradec Králové, Česká
      republika </description>
  </prediction>
  <prediction>
    <description>Labská, Špindlerův Mlýn, Česká
      republika </description>
  </prediction>
</AutocompletionResponse>

```

Ukázka kódu 4.1.4 XML Dokument ve službě Places API společnosti Google – výsledek pro `input=labsk`

V ukázce kódu 4.1.4 je znázorněn zjednodušený výsledný XML dokument, jakožto výsledek na požadavek s parametrem `input=labsk`. Jako v přechozích XML dokumentech, i zde se nachází element `status`, který uvádí, zda bylo vyhledávání úspěšné, anebo proč nebylo úspěšné. V jednotlivých elementech `prediction` se nachází element `description`, který obsahuje adresu s ulicí a obcí.

4.2 Mapy.cz

Jediným českým zástupcem poskytovatelů mapových podkladů pro vytvoření aplikace pro evidenci jízdnic cest zaměstnanců firem, jež je cílem této bakalářské práce, je Seznam.cz s internetovou službou Mapy.cz. Stejně jako Google.com poskytuje mapové podklady pro komerční i nekomerční využití, ve kterých jsou zobrazeny státy, města, ulice, a také možnost plánování tras automobilem nebo na kole.

Ve srovnání se službami poskytovanými společností Google.com nelze prostřednictvím služby Mapy.cz zjistit vzdálenost a délku trvání jízdy na pozadí, tzn. bez zobrazení samotné mapy. Mapy.cz poskytují pouze zobrazení mapy na vlastní HTML stránce pomocí služby Mapy API verze 4.8. Na druhou stranu výhodou služby Mapy.cz oproti ostatním poskytovatelům mapových podkladů je snadný způsob tisku konkrétní mapy prostřednictvím tlačítka pro tisk přímo u zvolené mapy.

4.2.1 Mapy API verze 4.8

Služba Mapy API verze 4.8 poskytuje třetím stranám mapové podklady s možností využívání některých funkcí mapové aplikace společnosti Seznam.cz. Mapy API jsou naprogramovány obdobně jako u konkurenční společnosti Google.com. Tzn. v programu JavaScript, který je následně vnořen do těla HTML souboru. Navíc pro vložení mapových podkladů je třeba přidat do hlavičky HTML stránky dva řádky JavaScriptu, které zajišťují, načtení potřebných definic a symbolů [8].

```
<script type="text/javascript" src="//api.mapy.cz/loader.js">
</script>

<script type="text/javascript">Loader.load()</script>
```

Vlastní nastavení mapy se provádí v těle HTML stránky, kde je vložen JavaScript. V JavaScriptovém kódu se pak následně nastavují všechny vlastnosti daného mapového podkladu. V ukázce JavaScriptového kódu vyobrazeném výše je znázorněno nastavení pro základní zobrazení mapy, kde je prezentovaná mapa vycentrována na souřadnice 50.12655N a 14.41790E bez možnosti ovládání. Je ale nutné zmínit, že na rozdíl od

konkurenčních mapových podkladů společnosti Google.com jsou zde souřadnice vkládány v obráceném pořadí [8].

```
<script type="text/javascript">
var center = SMap.Coords.fromWGS84(14.41790, 50.12655);
var m = new SMap(JAK.gel("m"), center, 13);
m.addDefaultLayer(SMap.DEF_BASE).enable();
</script>
```

Je také potřeba nastavit základní prvky pro ovládání mapy prostřednictvím příkazu v JavaScriptu. Pro přidání všech ovládacích prvků je potřeba na instanci třídy *SMap* zavolat metodu *addDefaultControls* (*m.addDefaultControls()*). Lze přidat i jeden ovládací prvek, například jen možnost přiblížení mapy.

```
var o = {title:"Posun mapy"};
var c = new SMap.Control.Compass(o);
m.addControl(c, {left:"8px", top:"9px"});
```

Je-li cílem naplánování či změření trasy, je třeba definovat počáteční a cílovou adresu. V tomto případě je možné přidat až tři průjezdní body. Souřadnice se ukládají do proměnné *coords* ve formátu zeměpisné šířky a délky. Na příkladu uvedeném níže je vidět zadání dvou adres. Vzhledem k tomu, že se adresy zapisují za sebe, je důležité dodržet pořadí adres [8].

```
var coords = [
    SMap.Coords.fromWGS84(14.434, 50.084),
    SMap.Coords.fromWGS84(16.600, 49.195)
];
```

Souřadnice adres, které byly zadány uživatelem, jsou dále předány do konstruktoru třídy *Route*. Zde se předávají již nadefinované souřadnice a reference na funkci, která má být volána po vyhledání trasy.

```
var route = new SMap.Route(coords, nalezeno);
```

Volaná funkce pro vyhledání trasy, tedy *nalezeno*, slouží pro zobrazení výsledné mapy. Tato funkce také určuje velikost zoomu a také způsob zobrazení nalezené trasy.

Mapy.cz společnosti Seznam.cz představují tuzemského zástupce poskytovatelů mapových podkladů pro vytvoření aplikace pro evidenci jízdnic cest zaměstnanců firem. Fakt, že jde o tuzemského zástupce, může usnadnit práci s těmito podklady s využitím případné podpory přímo společnosti Seznam.cz.

4.3 Here.com

Mapy Here.com jsou internetová aplikace poskytovaná společností Nokia. Tato společnost patří mezi světovou špičku v mapování. Důkazem toho je i využívání jejich mapových podkladů společnostmi BMW, Garmin či Toyota. Společnost Nokia usiluje o bohaté, vysoce aktuální mapové poklady s velkou mírou spolehlivosti spojující virtuální a fyzické světy ve 2D a 3D rozměru [9].

Stejně jako poskytovatelé uvedení v předchozích kapitolách i Here.com poskytují mapové podklady se základními funkcemi (např. volba trasy pro automobil, kolo, chůzi pěšky či jízdu prostřednictvím hromadné dopravy) a možnostmi (např. vyhýbání se dálnicím nebo mýtným branám, hledání nejkratší nebo nejrychlejší cesty). K vyhledané trase je také možné zobrazit alternativní trasy. Here.com mapy umožňují využívat API za účelem měření vzdáleností prostřednictvím odkazu URL. Navíc Here.com mapy také zobrazují klasické mapy, které jsou k dispozici na jejich internetových stránkách.

Pro účely vytvoření aplikace, která je výstupem této bakalářské práce, je možné použít z Here.com map služby *Matrix Routing API*, *JavaScript API 3.0* a *Routing API*.

4.3.1 Matrix Routing API

Matrix Routing API je webová služba internetové aplikace Here.com poskytující řešení pro výpočet vzdáleností a času mezi více vybranými místy současně. API zde funguje na principu matice $N \times M$, kdy se předává N výchozích adres a M cílových adres. Pro předanou matici služba vypočte všechny možné kombinace mezi výchozími a cílovými adresami. Konečný výsledek je tvořen jednotlivými řádky, kde každý řádek představuje informace obsahující časový údaj a vzdálenost počáteční a cílové adresy. Navíc oproti konkurenčním mapovým podkladům společnosti Google.com poskytuje služba *Matrix Routing API* detailnější informace o nalezené trase (např. aktuální informace o průjezdnosti jednotlivých úseků silnic) [9].

Služba dále poskytuje základní funkce (např. volba nejkratší/nejrychlejší cesty či způsobu dopravy). Lze vybrat i možnost jízdy nákladním automobilem, kdy je možné zvolit délku a váhu vozidla či zda má vozidlo přívěs [9].

Všechny dostupné informace, dle kterých má být výsledná cesta vyhledána, jsou odeslány prostřednictvím jednoduchého HTTP dotazu. Základní tvar URL adresy, do které se vkládají informace má tvar `http://route.st.nlp.nokia.com/routing/6.2/calculatematrix.xml?app_id=?&app_code=?&start0=?&destination0=?&mode=?`. Přičemž příkazem `calculatematrix.xml` je využita služba *Matrix Routing API*. Výsledek se následně zobrazí buď ve formátu XML, anebo formátu JSON [9].

Další nedílnou součástí URL adresy služby *Matrix Routing API* je `APP_ID` a `APP_CODE`. Tyto klíče definují oprávnění pro využívání dané služby. Jsou spojeny s účtem jednotlivých uživatelů registrovaných na webových stránkách Here.com společnosti Nokia. Nechce-li být uživatel registrován či mu postačí základní nabízené služby, je možno použít standardní hodnoty `APP_ID: DemoAppId01082013GAL`; a `APP_CODE: AJKnXv84fjrb0KIHawS0Tg` [9].

Služba *Matrix Routing API* je poskytována zdarma do limitu využití 100 000 transakcí za měsíc. Je-li využito větší množství transakcí, tzn. do limitu 150 000 transakcí za měsíc, je služba zpoplatněna 59 \$/měsíc. Plná verze této služby stojí přibližně 999 \$/měsíc a poskytuje limit 2 000 000 transakcí měsíčně.

Zadávání vybraných adres do této služby se děje prostřednictvím parametrů `startN` a `destinationN`, přičemž N vyjadřuje číslo, které odlišuje jednotlivé adresy. Zápis

jednotlivých adres je formou zeměpisné šířky a délky. Dalším kritériem zadání je parametr *searchrange*, který udává maximální vzdálenost vybrané cesty [9].

```
<rtcm:CalculateMatrix>
  <Response>
    <MetaInfo>...</MetaInfo>
    <MatrixEntry>
      <StartIndex>0</StartIndex>
      <DestinationIndex>0</DestinationIndex>
      <Route>
        <Summary>
          <Distance>21652.0</Distance>
          <BaseTime>977.0</BaseTime>
        </Summary>
      </Route>
    </MatrixEntry>
  </Response>
</rtcm:CalculateMatrix>
```

Ukázka kódu 4.3.1 – XML Dokument ve službě Matrix Routing API mapových podkladů Here.com společnosti Nokia – výsledek vyhledávání pro adresy Hradec Králové a Pardubice

Zobrazení výsledného XML formátu službou Matrix Routing API mapových podkladů Here.com společnosti Nokia je uvedeno v ukázce 4.3.1. Zadání obsahuje jednu výchozí adresu v Hradci Králové a jednu cílovou adresu v Pardubicích. V elementu *MetaInfo* jsou uloženy informace typu, kdy byl požadavek odeslán a kdy byl naposledy updatován. Důležité informace se nacházejí v elementu *route*, kde jsou elementy *distance* a *baseTime*, které obsahují vzdálenost v metrech a čas v sekundách.

4.3.2 JavaScript API 3.0

The HERE JavaScript API 3.0 API je služba podporující tvorbu webových aplikací s plnohodnotnými mapovými podklady. Umožňuje zobrazovat informace dle zadaných parametrů. Díky této službě patří Here.com mapy mezi špičku ve světě, protože si společnost Nokia zakládá na poskytování kvalitních a aktualizovaných mapových podkladů. Z názvu je patrné, že tato služba je naprogramována programovacím jazykem JavaScript, který se následně vkládá do těla HTML stránky. Také je vhodné v této službě nastavit jazyk v parametru *language*, ve kterém má být zobrazován text [10].

URL odkaz v hlavičce těla HTML stránky v této službě slouží k načtení všech potřebných definic. Následuje nastavení všech potřebných parametrů, které se ukládají do proměnné *routeRequestParams* (ukázka kódu 4.3.2.).

Výchozí nebo cílové adresy jsou v této službě zadávány jako zeměpisné souřadnice. Parametr *mode* v této službě slouží pro nastavení typu užitého dopravního prostředku a zvolené cesty (nejkratší vs. nejrychlejší cesta). Dalším nastavitelným parametrem je *representation*, který slouží pro nastavení způsobu zobrazení výsledné mapy. Může to být např. způsobem jako mapa s vyobrazenou trasou a možností její úpravy pomocí posunutí bodu na mapě [10].

```
routeRequestParams = {
    mode: 'fastest;truck',
    representation: 'display',
    waypoint0: '41.9798,-87.8801', // Chicago O'Hare
    waypoint1: '41.9043,-87.9216', // Eisenhower
Expressway
    instructionformat : 'html'
};
```

Ukázka kódu 4.3.2 - nastavení proměnné *routeRequestParams* ve službě JavaScript API 3.0 mapových podkladů společnosti

Příkladem může být ukázka kódu 4.3.2, kde je nastavena možnost nejrychlejší cesty pro nákladní automobil s výchozí adresou z Hradce Králové do cílových Pardubic. Další

vloženou informací je podoba doplňujícího text k zobrazované mapě. Výsledná mapa bude zobrazena s nalezenou trasou bez možnosti pohybovat body v mapě [10].

4.3.3 HERE Routing API

Další službou mapových podkladů Here.com společnosti Nokia je *Here Routing API*, která je webovou službou poskytující snadné a rychlé hledání tras.

V této službě nabízí API i možnost vyhledávání tras pro zadané průjezdní body. Požadavek je opět zadáván jako HTML dotaz, ve kterém jsou předávány upřesňující informace. Základní tvar adresy URL je `http://route.cit.api.here.com/routing/7.2/calculateroute.json?app_id=?&app_code=?&waypoint0=?&waypoint1=?&mode`

Výsledek této služby je nabízen ve formátech JSON nebo XML. Nutným zadávaným parametrem je definování počáteční a cílové adresy. Dále pak parametrem *mode* lze nastavit druh dopravního prostředku pro zvolenou trasu a její možnosti (např. nejkratší vs. nejrychlejší cesta). Výchozí a cílové adresy jsou zadávány ve formátu zeměpisných souřadnic podobně jako další průjezdní body, a to do parametru *waypointX*, kde X zastupuje číslici v pořadí adres [11].

```
<rtcr:CalculateRoute xmlns:rtcr="http://www.navteq.com/lbbsp/Routing-CalculateRoute/4">
```

```
<Response>
```

```
<MetaInfo>...</MetaInfo>
```

```
<Route>
```

```
<Waypoint>
```

```
<LinkId>-1609300611616932926</LinkId>
```

```
<MappedPosition>...</MappedPosition>
```

```
<OriginalPosition>...</OriginalPosition>
```

```
<Type>stopOver</Type>
```

```
<Spot>0.351145</Spot>
```

```
<SideOfStreet>left</SideOfStreet>
```

```
</Waypoint>
```

```
<Waypoint>
```

```
<LinkId>+1609300624501834261</LinkId>
```

```
<MappedPosition>...</MappedPosition>
```

```

        <OriginalPosition>...</OriginalPosition>
        <Type>stopOver</Type>
        <Spot>1.0</Spot>
        <SideOfStreet>neither</SideOfStreet>
    </Waypoint>
    <Mode>...</Mode>
    <Leg>...</Leg>
    <Summary>
        <Distance> 23733</Distance>
        <TrafficTime> 1377</TrafficTime>
        <BaseTime> 1311</BaseTime>
        <Flags>builtUpArea</Flags>
        <TravelTime>1311</TravelTime>
    </Summary>
</Route>
</Response>
</rtcr:CalculateRoute>

```

4.3.3 - Ukázka kódu ve formátu XML dokumentu ve službě *Here Routing API* mapových podkladů *Here.com* společnosti *Nokia* – výsledek pro výchozí adresu *Hradec Králové* a cílovou adresu

Z ukázky kódu 4.3.3 je zřejmé, že ve srovnání s konkurenční službou *The Google Directions API* je výsledný XML dokument služby *Here Routing API* poměrně rozsáhlý a nepřehledný. Element *MetaInfo* obsahuje informace, dle kterých byla mapa nastavena. Jednotlivé adresy jsou rozděleny do elementů *Waypoint*. Uvnitř elementu se nachází informace o souřadnicích zeměpisné šířky a délky a pozici na mapě. Element *Mode*, obsahuje údaje, pro jaký dopravní prostředek má být trasa hledána a zda hledat nejrychlejší nebo nejkratší trasu. V elementu *Leg* jsou informace pro potřeby GPS, kde je naplánována cesta po jednotlivých křižovatkách. V posledním elementu *Summary* se nachází vzdálenost v metrech a čas v sekundách [11].

5 Analýza a návrh

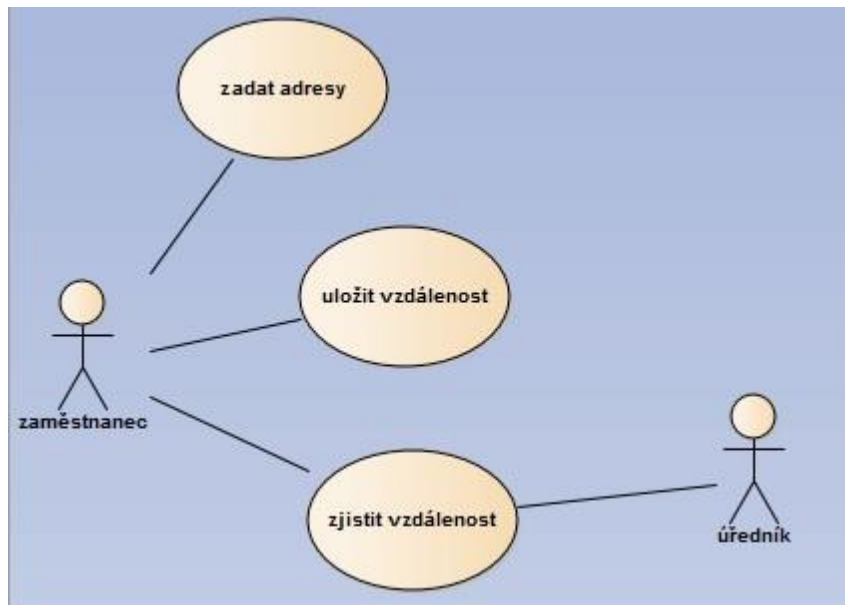
Základním předpokladem pro správnou funkci prototypu aplikace pro záznam jízdnic cest zaměstnanců je schopnost definovat vzdálenost a přibližný čas strávený na cestě. Vstupními informacemi je tedy nutné definovat počáteční a cílové adresy. Všechny vybrané mapové podklady jednotlivých společností detailněji popsaných v předchozích kapitolách nabízí i průjezdní body na trase. Jejich počet se ale může lišit, proto je potřeba tyto body přidávat dynamicky. Při vývoji prototypu aplikace nesmí být opomenuty další důležité aspekty, kterými lze upravovat cestovní trasy (např. možnost vyhnout se dálnicím či mýtným branám). Pro snadnější a rychlejší zadávání adres je vhodné implementovat tzv. „našeptavač“, který uživateli nabízí vyhledané výsledky podle již zadaného textu. Např. společnost Google.com tuto službu poskytuje a doporučuje při používání svých mapových podkladů.

První z možností implementace je vytvoření grafického rozhraní sloužící ke komunikaci s uživatelem bez následného zobrazení odpovídající mapy. Zde by mohl uživatel po vyzvání zadat upřesňující informace vedoucí k vyhledání nejvhodnější trasy (např. počáteční a cílová cesta). Další informace týkající se cestovní trasy, jako např. průjezdní body, eliminace dálnic a mýtných bran, nelze považovat za nezbytné informace. Uživatel po vyplnění jen odešle požadavek a aplikace sama následně zobrazí zjištěnou vzdálenost a časový údaj dané cesty. Druhou možností implementace je vytvoření identického grafického rozhraní jako v prvním případě, ale s tím rozdílem, že uživateli se zobrazí další okno, ve kterém bude vykreslena uživatelem požadovaná mapa. V obou případech implementace by měl mít uživatel možnost s vyobrazenou trasou nesouhlasit nebo možnost poupravit průjezdní body tak, aby odpovídaly skutečné trase služební jízdy.

Po upravení trasy uživatelem by v ideálním případě již nemusel nic potvrzovat. Výsledná vzdálenost a časový odhad služební jízdy by se tedy automaticky zobrazily v grafickém rozhraní, ze kterého uživatel vycházel. V samotné mapě by už posléze nebyla možnost vytvářet další průjezdní místa na trase. Toto lze pouze v uživatelském rozhraní. Kromě toho by v zobrazené mapě měla být uživateli povolena pouze změna umístění již zobrazených míst.

Návrh lze tedy vytvořit pro dvě možnosti implementace mapových pokladů do vyvíjené aplikace pro evidenci jízdnic cest – verzi bez mapy a verzi s mapou.

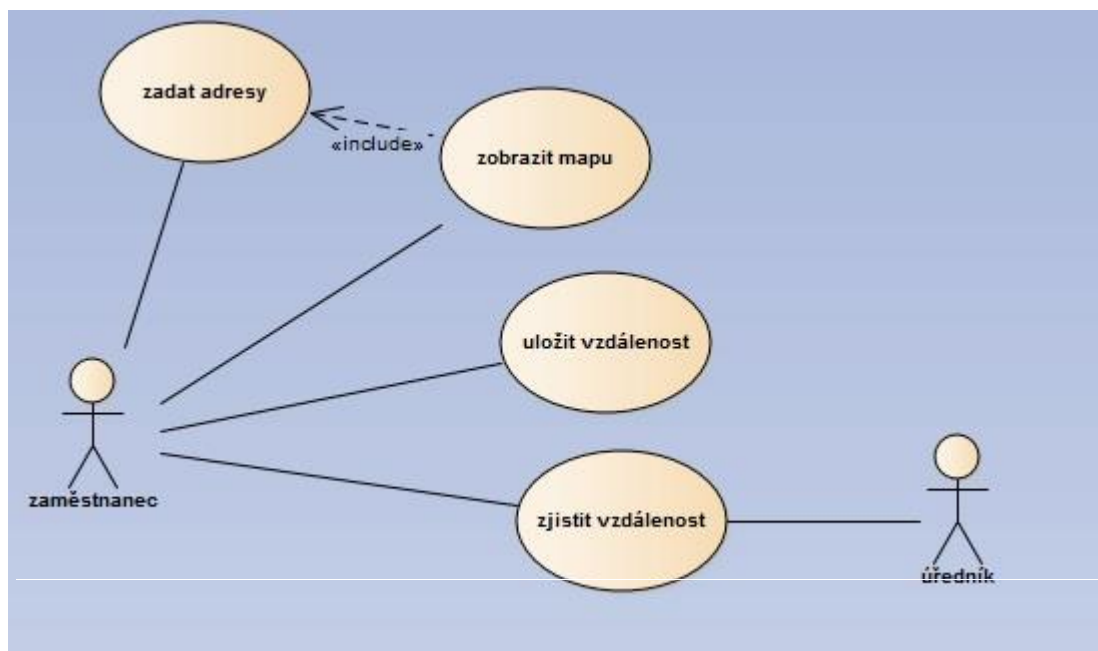
5.1 Verze aplikace evidence jízdních cest pro uživatele bez zobrazování mapy



Obrázek 5.1 Use Case Diagram bez mapy

Z příkladu uvedeného v obr. 5.1 vyplývá, že zaměstnanec firmy má možnost zadat počáteční a cílovou adresu, a to včetně přidání průjezdních bodů. Zaměstnanec musí následně potvrdit odeslání požadavku pro vygenerování vzdálenosti dle jím zadaných parametrů. Vyhledanou trasu si po té bude moci uložit.

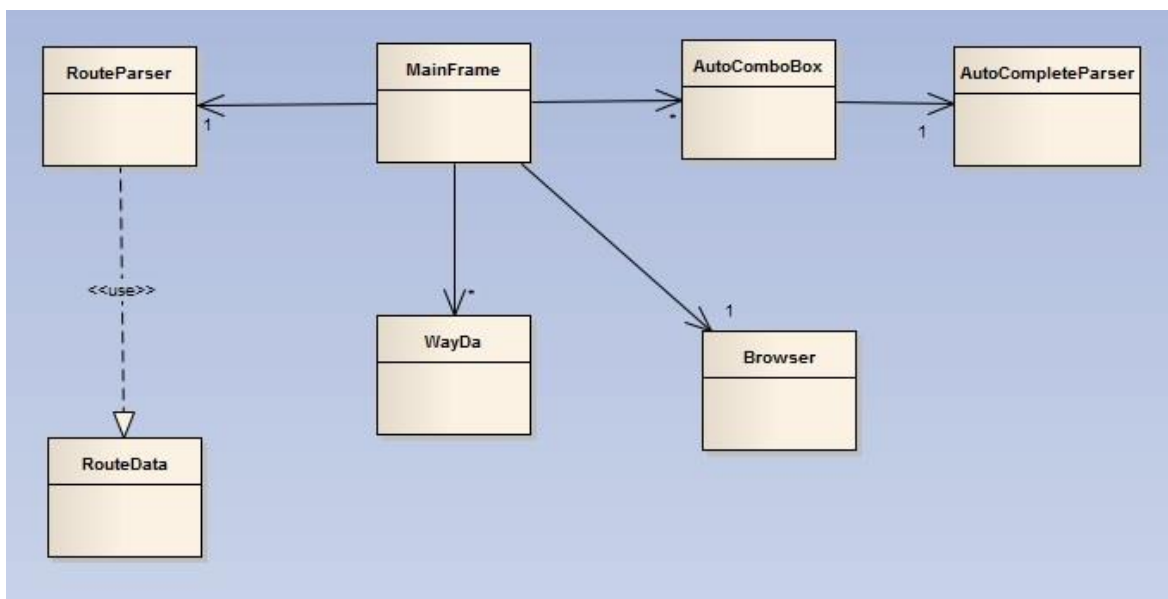
5.2 Verze aplikace evidence jízdních cest pro uživatele se zobrazením mapy



Obrázek 5.2.1 Use Case Diagram se zobrazovanou mapou

Z obrázku 5.2.1 vyplývá, že na počátku zaměstnanec zadá počáteční a cílovou adresu. Oproti přechozímu případu zobrazení aplikace verze bez mapy bude mít možnost zobrazit trasu služební jízdy na mapě. Ta může být dále upravována prostřednictvím posouvání jednotlivých bodů pomocí počítačové myši na vyobrazené trase. Konečná verze zobrazené trasy včetně mapy může být uložena. Pracovník zpracovávající knihu jízd v daném podniku si následně zobrazí adresy a jejich výslednou vzdálenost tak, jak byly zadány zaměstnancem užívající služební vozidlo.

Je známo, jaké úkoly bude aplikace muset vykonávat. V dalším kroku je potřeba vytvořit základní návrh jednotlivých tříd, které budou tvořit prototyp aplikace. Bude potřeba vytvořit uživatelské rozhraní. Dále bude je plánována třída, která bude obstarávat čtení XML dokumentů. Základní návrh Class diagramu je znázorněn na obrázku 5.2.2.



Obrázek 5.2.2 - Návrh Class diagramu

Na obrázku 5.2.2 je pak znázorněn návrh jednotlivých tříd pro zjednodušení bez atributů a metod. Spouštěcí třída MainFrame bude vytvářet hlavní okno aplikace, do kterého bude uživatel zapisovat adresy. Funkci tzv. „našeptavače“ zajišťuje třída AutoComboBox. Tato třída se stará o JComboBox, kterému nastavuje hodnoty do rozbalovací nabídky. Informace do rozbalovací nabídky získává ze třídy AutoCompleteParser, která pomocí HTTP dotazu získá XML soubor, ze kterého jsou požadované informace a předá je zpět. Po zadání adres a potvrzení vytvoří třída RouteParser HTTP dotaz. A výsledek z dokumentu XML uloží do objektu typu RouteData. Třída WayDa představuje průjezdní bod, který se bude dynamicky přidávat a odebírat. Jediným rozdílem od verze aplikace evidence jízd bez mapy je výskyt třídy Browser, která vytváří prohlížeč s mapovým podkladem.

6 Technologie využívaná při vývoji aplikace pro evidenci jízdních cest

Velmi důležitou roli při tvorbě aplikace záznamu jízdních cest hraje použitá technologie. Je třeba podotknout, že výsledná aplikace, jež je cílem této bakalářské práce, nebude sloužit jako samostatný celek, ale bude implementována do ERP systému OrsoftOpen. Ten je založen na platformě Java. Proto je pro snadnou implementaci vhodné, aby aplikace byla také založena na této platformě.

Podkladem vytvářené aplikace jsou mapy. Výstupem aplikace je pak vyhledání a zpracování cesty dle parametrů zadaných uživatelem. Při výběru vhodného poskytovatele mapových podkladů byly brány v úvahu zejména samotnost aplikace a jednoduchost případného implementování map. Proto byli vybráni tito kandidáti: Google.com mapy, Here.com mapy a Mapy.cz.

Z těchto tří kandidátů byly vyřazeny mapové podklady od českého poskytovatele společnosti Seznam.cz – Mapy.cz. Důvodem byla možnost naplánování jízdni trasy maximálně třemi průjezdními body a také omezený rozsah mapových podkladů, které pokrývají pouze Evropu a okraje přilehlých kontinentů. V budoucnu by se totiž nedalo vyloučit, že by používaný systém OrsoftOpen nepoužívala firma potřebující zmapovat služební cesty zaměstnanců i mimo tyto zobrazované oblasti.

Zbývá poskytovatelé map, tj. společnost Google.com a společnost Nokia, nabízejí téměř identické možnosti použití mapových podkladů při vytváření aplikace evidence jízd. Nicméně s ohledem na implementaci do vyvíjené aplikace jsou vhodnějším kandidátem mapy společnosti Google.com. Neboť mapové podklady poskytované touto společností a jejich API pro práci s nimi jsou více propracované a nabízejí jednodušší možnosti následných úprav zdrojového kódu pro vlastní potřebu aplikace. S mapovými podklady Hero.com společnosti Nokia lze případně počítat jako s odpovídající alternativou.

6.1 Programovací jazyk Java

Jednou z použitých technologií při vyvíjení prototypu aplikace evidence jízdních tras je platforma Java. Jde o programovací jazyk vydaný společností Sun Microsystems v r. 1995 [12]. V současnosti jde o jeden z nejpoužívanějších programovacích jazyků. Podle Tiobe indexu je dnes Java druhý nejpopulárnější programovací jazyk [13].

Tento program je objektově orientovaný a psaný formou strukturovaného jazyka. Je využíván pro vývoj programů na různých systémech od desktopových aplikací, přes mobilní telefony po rozsáhlé systémy pracující na více počítačích rozmístěných po celém světě. V květnu r. 2007 společnost Sun Microsystems uvolnila zdrojové kódy Javy, které obsahovali 2,5 miliónu řádků. Od této doby je dále Java vyvíjena jako Open Source.

Syntaxe jazyka Java by se dala přirovnat ke zjednodušené a lehce upravené verzi jazyka C a C++. Java využívá osm primitivních datových typů, mezi které patří *byte*, *int*, *short* a další. Umožňuje různé úrovně síťového spojení, což slouží k vytváření klientských aplikací a serverů. Také obsahuje zabezpečení při používání síťového spojení, které chrání před nebezpečnými operacemi a před napadením operačního systému počítače nepřátelským kódem. Ale programovací jazyk Java je nezávislý na operačním systému a architektuře. Výsledná aplikace funguje na všech platformách, které mají nainstalovaný správný virtuální stroj. Java podporuje tzv. vícevláknové aplikace, které umožňují chod více částí programu nezávisle na sobě.

6.2 Verze aplikace evidence jízdních cest pro uživatele bez zobrazování mapy

K vytvoření prototypu aplikace pro uživatele bez možnosti zobrazení map byla využita služba *The Google Directions API*. Tím lze získat vzdálenost mezi body A a B. V případě potřeby je možno přidat i několik průjezdních bodů. Pomocí této služby je možno změřit vzdálenost mezi dvěma místy a také čas, za který je možno tuto trasu ujet. Použitím služby *Directions API* lze dostat výsledek ve formátu XML nebo JSON. V rámci této bakalářské práce je preferován výstup ve formátu XML, který je podporován ERP systémem OrsoftOpen. Pro práci s dokumentem ve formátu XML se využije open source knihovna

Dom4j, která podporuje načítání dat podle jednotlivých elementů. Tato knihovna také umožňuje vytváření potřebných XML dokumentů [5].

6.2.1 XML

Extensible Markup Language zkráceně XML, česky rozšiřitelný značkovací jazyk. XML je zjednodušenou verzí staršího jazyka SGML. Jazyk je především využíván pro výměnu dat, vzhled moc neřeší. Nejčastěji bývá definován do kaskádového stylu. XML dokument se skládá z jednotlivých značek (tagů), které obsahují jednotlivé části textu [14].

6.2.2 JSON

JavaScript Object Notation zkráceně JSON je způsob zápisu dat nezávislých na platformě. Podobně jako XML je určený především pro přenos dat. Vstupem je libovolná datová struktura, číslo, řetězec či objekt. Výstupem je pak vždy řetězec. Data zapsaná formou JSON, mohou být uložena a přenášena jako soubory. Nejčastěji se však používá pro přenos intranet nebo internet [14].

6.2.3 Open Source knihovna Dom4j

Dom4j je open source knihovna pro práci s formáty XML, XPath a XSLT na platformě Java. Dom4j je kompatibilní se standardy DOM, SAX a JAXP. Nespornou výhodou této knihovny je získání informací z velkých dokumentů typu XML s minimálním využitím paměti [15].

6.3 Verze aplikace evidence jízdnicích cest pro uživatele se zobrazováním mapy

K vytvoření aplikace pro uživatele s možností zobrazení map byla využita služba *Google Maps JavaScript API v3*. Díky této službě lze vybrat možnosti zobrazení typu mapy, popř. informace zobrazené na mapě.

Pro vytvoření aplikace byla zvolena možnost mapy s tzv. *Draggable directions*, které zobrazí trasu z bodu A do bodu B s možnostmi možného posunutí adres na vyobrazené mapě. Dále bude mít uživatel k dispozici eventualitu vytvoření na trase jízdy libovolný další bod, který je možno umístit na mapu jako rozšíření zadané trasy. Pro zobrazení mapy poslouží *embedded prohlížeč*, který bude vytvořen pomocí služby *JavaFX*, konkrétně *WebView*. Pomocí služeb *WebView* v *JavaFX* se načítají a zobrazují webové stránky. Navíc služba *WebView* zvládá i funkce *JavaScriptu*. Proto je tato služba schopna zajistit komunikaci mezi *JavaScriptem* a vyvíjenou aplikací oběma směry [16].

6.3.1 Služba JavaFX

JavaFX je softwarová platforma založena na platformě *Java*. Slouží pro vývoj internetových aplikací. Jedná se především o prvky, které podporují interaktivitu internetových stránek, které s uživatelem nějakým způsobem komunikují. *JavaFX* nepodporuje stylování pomocí *CSS* jazyka, ale využívá vlastní atributy. Pro přehrávání videa a zvuku mohou být použity pouze kodeky, které licencuje společnost *Oracle*. *JavaFX* je stále rozvíjen a s každou novou verzí přibývá mnoho užitečných funkcí pro tvorbu webových aplikací [16].

6.3.2 Služba JavaScript

JavaScript je multiplatformní a objektově orientovaný scriptovací jazyk vyvinutý společností *Netscape*. Je využíván především jako programovací jazyk pro webové stránky, jenž je vkládán přímo do těla *HTML* stránky. Syntaxí se řadí mezi jazyky jako např. *C*, *C++* a *Java* [17].

Slovo Java v názvu je v tomto případě jen čistě z marketingových důvodů, neboť JavaScript spojuje s Javou jen čistě podobná syntaxe. Na rozdíl od ostatních interpretovaných jazyků (např. PHP nebo ASP) je JavaScript překládán na straně klienta po načtení webových stránek. V případě potřeby se dá JavaScript spouštět i na straně serveru [17].

7 Implementace aplikace pro evidenci jízdních cest do ERP systému

V rámci implementace budou propojeny jednotlivé části vyvíjené aplikace pro záznam jízdních cest uživatelem. Zvolená technologie pro implementaci musí být co nejjednodušeji propojena s ostatními celky aplikace. Je proto důležité vybrat adekvátní technologii a podpořit jejich funkcionalitu jako celek. Výsledná aplikace by měla být fungujícím prototypem. Nicméně její jednotlivé třídy nutně nepředstavují třídy, které budou následně implementovány do ERP systému OrsoftOpen. Spouštěcí třída MainFrame, slouží pouze pro účely ověření prototypu aplikace.

7.1 Třída MainFrame

MainFrame je třída, která vytváří hlavní uživatelské rozhraní pro komunikaci s uživatelem. Hlavní okno aplikace je vytvořeno pomocí třídy JFrame. Pro zadávání adresy jsou v okně umístěny JComboBoxy, které v rozbalovací nabídce nabízí předem vyhledané adresy, dle již zadaného textu. Pro odeslání požadavku slouží tlačítko JButton *propočítat*. To zavolá metodu nactiVzdalenost() (viz obrázek 7.1.1). Metoda nactiVzdalenost() má za úkol získat dokument typu XML pomocí vytvořené URL adresy ze zadaných adres. V této metodě je použit SwingWorker pro přenesení práce na pozadí. Což umožňuje provádění dalších akcí v aplikaci, aniž by musela čekat na výsledek. Dále je zavolána metoda parse(), třídy Parser. Metoda parse() vrací instanci na třídu ParseAdress. Třída ParseAdress obsahuje informace o vzdálenosti, názvy všech zadaných adres a časový údaj říkající, jak dlouho přibližně trvá zdolání trasy

```
private void nactiVzdalenost() {  
  
    txtVzdalenost.setText("Načítám...");  
  
    SwingWorker<RouteData, String> swingWorker = new  
SwingWorker<RouteData, String>() {  
  
        protected RouteData doInBackground() throws Exception {
```

```

JTextField a = (JTextField)
cbAdresa1.getEditor().getEditorComponent();

JTextField b = (JTextField)
cbAdresa2.getEditor().getEditorComponent();

RouteData data = parser.parse(a.getText(), b.getText(),
udelejWayPoints());

return data;
    }

    protected void done() {
try {
RouteData data = get();
nastav(data);
} catch (InterruptedException | ExecutionException e) {
e.printStackTrace();
}
super.done();
    }
};

swingWorker.execute();
}

```

Ukázka kódu 7.1.1 Třída testovací - metoda nactiVzdalenost()

V rozšířené verzi aplikace o embedded prohlížeč přibude v hlavním okně aplikace tlačítko *JButton Zobrazit Mapu*. Po kliknutí na tlačítko se otevře embedded prohlížeč, ve kterém se načte HTML soubor *googlemap.html*. Soubor obsahuje JavaScript, kterému se při spouštění prohlížeče předávají adresy. Nejprve se vytvoří instance třídy *RouteInformation*, která obsahuje zadané adresy, případně průjezdní body a další možnosti zvolené uživatelem. Následně je na instanci třídy *Browser*, volána metoda *zobrazit(RouteInformation)*. Ta zobrazí mapu dle parametrů zadaných uživatelem.

7.1.1 Třída `SwingWorker`

Za předpokladu, že program musí vykonat úlohu zabírající delší časový úsek, je vhodné použít jedno z pracovních vláken, jež jsou někdy nazývána jako vlákna na pozadí. Sám `SwingWorker` je abstrakční třída, ve které je potřeba definovat podtřídy pro vytvoření `SwingWorker` objektu. V aplikaci je `SwingWorker` použit jako anonymní třída, což je vhodné při použití pro velice jednoduché objekty [18].

`SwingWorker` má několik metod pro komunikaci s běžícím vláknem [18]:

- `SwingWorker.done` je metoda volána po zakončení práce vlákna.
- `SwingWorker.future` je metoda pro komunikaci s vláknem, kdy lze výsledek z vlákna přesunout na jiné vlákno. Dále také tato metoda umožňuje zrušení úlohy.
- `SwingWorker.publish` je metoda pro získání průběžných výsledků na běžícím vlákně.

7.2 Třídy `RouteParser` a `RouteData`

Třída `RouteParser` slouží pro získání informací z dokumentu ve formátu XML, který získá prostřednictvím služby *The Google Directions API*. Pro načtení souboru je použita třída `SAXReader` knihovny `Dom4j`. Ve třídě `RouteParser` se řeší ošetření proti možným chybám, které by mohly nastat při chodu aplikace (např. nesprávně zadaná adresa či neexistující cesta mezi místy). Ochrana před chybami se děje prostřednictvím XML dokumentu, který se získá ze služby *The Google Directions API*. Tento XML dokument obsahuje informace o vyhledání požadovaných adres a možnosti jízdní trasu vyhledat [5]. Pro ilustraci je v obr. 7.1 znázorněn XML dokument s nevyhledanou trasou mezi Prahou a New Yorkem. Naproti tomu obr. 7.2 představuje ukázkou XML dokumentu se správně zadanou adresou z Prahy do Brna a úspěšně vyhledanou cestou. Pro uložení informací získaných z XML dokumentu slouží třída `RouteData`.

```
<DirectionsResponse>
<status>ZERO_RESULTS</status>
</DirectionsResponse>
```

Obrázek 7.1, ukázka XML dokumentu s chybným zadáním pomocí služby Directions API společnosti Google.com.

```
<DirectionsResponse>
  <status>OK</status>
  <route>
    <summary>D1/E65</summary>
    <leg>
      <step>...</step>
      <step>...</step>
      <duration>
        <value>6955</value>
        <text>1 hodin, 56 minut</text>
      </duration>
      <distance>
        <value>205460</value>
        <text>205 km</text>
      </distance>
      <start_address>Praha, Česká republika</start_address>
      <end_address>Brno, Česká republika</end_address>
    </leg>
  </route>
</DirectionsResponse>
```

Ukázka k'du 7.1 - Ukázka upraveného XML dokumentu bez chybného zadání pomocí služby Directions API společnosti Google.com. Pro trasu z Prahy do Brna.

Hlavním ukazatelem XML dokumentu je element `status`. Obsahuje informaci, zda byla trasa nalezena, příp. proč nebyla trasa vyhledána. Element `summary` uvádí, po kterých konkrétních hlavních silnicích trasa povede. Hlavní informace o trase jsou v elementech `distance` a `duration`, které obsahují údaje o vzdálenosti a čase. V elementech `start_address` a `end_address` jsou uvedeny přesné adresy v textové podobě, pro které byla trasa hledána [5].

7.3 Třída Browser

Třída `Browser` zajišťuje vytvoření `embedded` prohlížeče. Nejdůležitější částí této třídy je vytvoření komunikace mezi `JavaFX`, ve které byl prohlížeč vytvořen, a programem `JavaScript`, který je použit v souboru ve formátu `HTML` pro zobrazování mapy pomocí služby `Google Maps JavaScript API v3`. Prostřednictvím této komunikace je možno z vyvíjené aplikace pro evidenci jízdnicích cest předávat do `JavaScriptu` adresy. A následně lze z `JavaScriptu` předávat výslednou vzdálenost a časový údaj zpět do této aplikace. Komunikace probíhá prostřednictvím třídy `WebEngine`. Díky ní může být `JavaScript` vyvolán (viz ukázka kódu 7.3.1). V souboru `googlemap.html` je dále vytvořena proměnná `výsledek` a `myTime` (viz ukázka kódu 7.3.2). Posléze je zavolána metoda v aplikaci, které jsou tyto proměnné předány jako parametr. Adresy jsou nastaveny do těla `JavaScriptu` pomocí přepsání připravených hodnot v `HTML` souboru. Přepsání probíhá pomocí metody `replace()`. Před přepsáním adres je důležité adresy nejdříve upravit do požadované formy. K tomu slouží `settry` při předávání adres z třídy `MainFrame` do třídy `MyBrowser` [6]. Nutná úprava adres je zobrazena v ukázce kódu 7.3.3.

```
JSObject jsobj = (JSObject) engine.executeScript("window");  
jsobj.setMember("java", this);
```

Ukázka kódu 7.3.1 – Nastavení komunikace mezi aplikací a `JavaScriptem`

První řádek kódu 7.3.1 znázorňuje vytvoření objektu třídy `JSObject`, který obsahuje referenci na globální objekt `window` v `JavaScriptu`. Na následujícím řádku je předávána

reference na objekt, na kterém bude moci JavaScript volat veřejné metody. V tomto případě je předávána reference na objekt, ve kterém je kód vykonáván. Řetězec "java" poukazuje na jméno proměnné v JavaScriptu [6].

```
var adresy = "";

for (var i = 0; i < myroute.legs.length ; i++) {
    adresy = adresy + myroute.legs[i].start_address + ";" ;
}

var myTime = "";

for (var i = 0; i < myroute.legs.length; i++) {
    myTime += myroute.legs[i].duration.value;
}

adresy = adresy + myroute.legs[myroute.legs.length-
1].end_address + ";";

java.setPruAdresy(adresy);

java.setVzdalenost(total);

java.setMyTime(myTime);
```

Ukázka kódu 7.3.2 – volání metod třídy Browser z JavaScriptu.

V kódu 7.3.2 je vytvořena proměnná adresy, ve které jsou následně uvedeny přesné názvy všech zadaných míst. Adresy jsou postupně přidávány za sebe pomocí for cyklu. Jednotlivé adresy jsou rozděleny středníkem. V proměnné myTime se uloží informace o celkovém potřebném čase stráveném na cestě. Proměnná myTime je plněna pomocí for cyklu, kdy je potřeba projít jednotlivé části trasy a sečíst dílčí časy pro získání celkového výsledného času. Po naplnění všech proměnných jsou zavolány jednotlivé metody třídy Browser. Jednotlivé settery posléze nastaví získané hodnoty z JavaScriptu do prostředí Javy, a ty jsou následně zobrazeny uživateli.

```
public void setAdresa1(String adresa1) {  
    adresa1 = "'" + adresa1 + "'";  
    this.adresa1 = adresa1;  
}
```

Ukázka kódu 7.3.3 – Metoda setAdresa, nutné přidání apostrofu před a za konec adresy

Při zadávání adresy do JavaScriptu je nutné, aby jednotlivé adresy byly uvnitř dvou apostrofů. Přidávání apostrofů se provádí v metodě setAdresa() třídy RouteInformation, která slouží pro předávání informací třídě Browser.

Embedded browser poskytnutý v JavaFX je založený na open source projektu WebKit. WebKit poskytuje zdarma engine pro webové prohlížeče. Díky tomu je možno využívat funkcí CSS kaskádových stylů, JavaScriptu, Document Object Model a HTML5. Komponenta WebView podporuje funkce HTML5 jako Canvas, média pro přehrávání, historie, DOM, SVG a doménová jména v národních jazycích, atd. [19].

8 Výsledky a doporučení pro vývoj aplikace pro evidenci jízdních cest

8.1 Testování aplikace pro evidenci jízdních cest

Před konečným implementováním prototypu aplikace pro záznam jízdních cest do ERP systému je nutné provést její otestování. Aplikace bude testována na verzi JDK 1.6, což je verze používaná softwarem OrsoftOpen. V případě použití novějšího JDK by nebyla možná implementace do ERP systému. Testování se zaměří na správnou funkcionální aplikaci, aby nedocházelo ke zkreslení zjišťovaných výsledků, a na správnou funkcionální aplikaci v jednotlivých operačních systémech.

Aplikace byla testována v operačním systému Windows 7 64bit, Windows7 32bit a Windows 8 64bit. Na operačních systémech Win7 a Win8 64bit. Výsledky testování neukazaly žádné znepokojující informace, vše probíhalo naprosto v pořádku. Nicméně na operačním systému Win7 32bit bylo nutné použít jinou verzi knihovny JavaFX, neboť zobrazená mapa byla špatně naformátována a celkový výsledek byl špatně naformátovaný. Po změně knihovny běžela aplikace bez komplikací a nebylo třeba provádět zásadní změny.

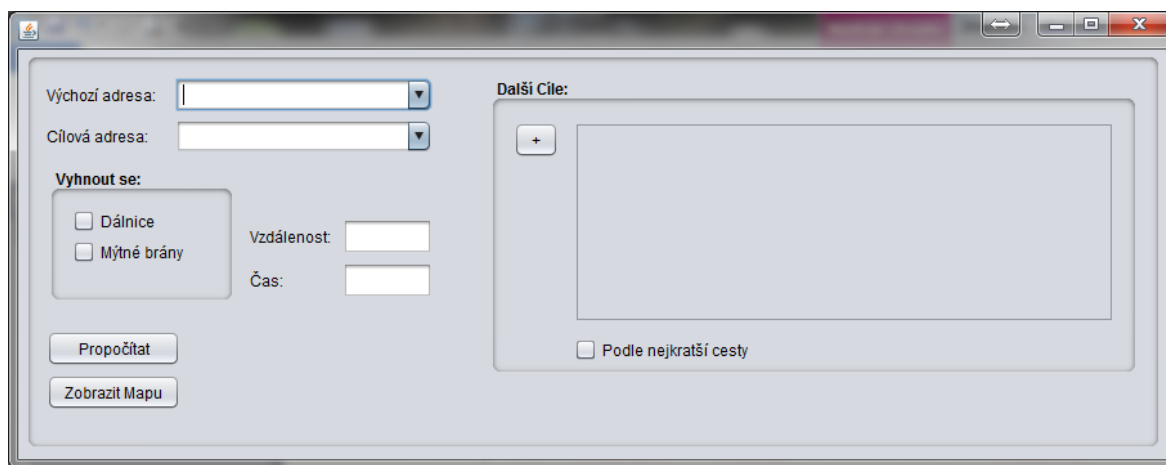
Při testování funkcionality vyvíjené aplikace byla pro porovnání výsledků provedena kontrola pomocí webového rozhraní poskytovatele mapových podkladů, tj. společnosti Google.com. Výsledky zobrazené aplikací a výsledky na internetových stránkách poskytovatele byly totožné. Což prokazuje tvrzení, že mapové podklady poskytované společností Google.com prostřednictvím API jsou totožné s mapovými podklady na internetových stránkách tohoto poskytovatele.

Také bylo nutné otestovat chyby, které mohou vzniknout při samotném běhu aplikace. Mezi tyto chyby lze zařadit např. špatné parsování dokumentu XML. Bylo tedy nutné ověřit, zda obdržený XML dokument obsahuje validní údaje. Z XML dokumentu byly všechny potřebné informace bez komplikací získány a zobrazeny uživateli. Problém také mohl nastat při komunikaci pomocí JavaFX, které umožňuje obousměrnou komunikaci mezi aplikací Java a JavaScriptem vytvářejícím mapu. Ačkoliv v komunikaci pomocí JavaFX nebyl shledán problém, samotná inicializace JavaFX je poměrně pomalá. První načtení a zobrazení okna s mapovým podkladem trvá cca 4 sekundy, ovšem další načtení mapy při změně souřadnic je již okamžité.

8.2 Výsledek aplikace pro evidenci jízdnicích cest

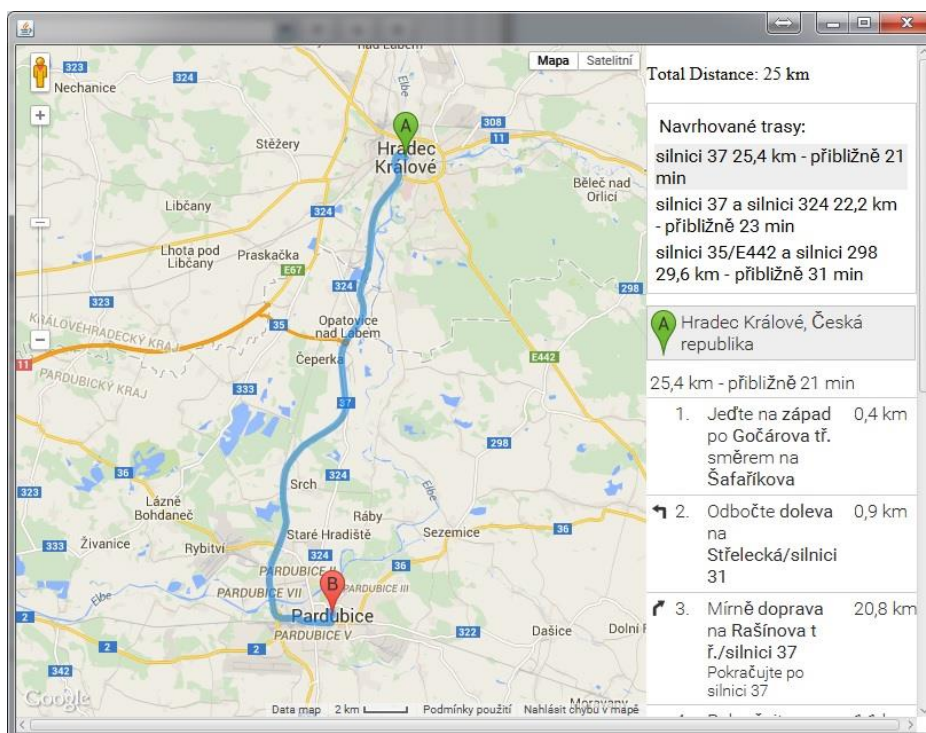
Výsledkem je vytvoření fungující aplikace za účelem monitorování jízdnicích cest uživatelů v podnicích. Její vzhled, možnosti a funkcionalita je popsán níže.

Jedním z hlavních cílů bylo vytvoření aplikace s co nejjednodušším rozhraním pro komunikaci s uživatelem (viz obr. 8.2.1). Při vyvíjení aplikace byly využity základní ovládací prvky jako, JComboBox nebo JTextField. Pro zadávání adres slouží JComboBox, který zobrazuje nápovědu dle textu zadaného uživatelem. Výsledky jsou vyobrazeny v JTextFieldu. Uživatel může označit možnost vyhnout se dálnicím a mýtným bránám. Dynamické přidávání průjezdních bodů je vyřešeno tlačítkem se znaménkem plus. Případná eliminace jednotlivých cílů se děje pomocí tlačítka se symbolem písmena X vedle JComboBoxu pro zadávání adresy. Mezi jednotlivými průjezdními body je možné měnit pořadí pomocí tlačítek se symbolem šipek nahoru a dolů, které jsou umístěny vedle JComboBoxu pro zadávání adresy.



Obrázek 8.2.1 – Hlavní okno vyvíjené aplikace pro evidenci jízdnicích tras uživateli v rámci systému OrsoftOpen společnosti Ortex s r. o.

Vyplněním údajů v hlavním okně aplikace lze zobrazit mapu. Příkladem může být obrázek 8.2.2, ve kterém je znázorněna mapa vykreslující cestu z Hradce Králové do Pardubic. Jak je vidět na obrázku, na výběr jsou pro uživatele tři alternativní trasy. Uživatel může dále vybranou trasu upravovat posouváním nejen počáteční a cílové cesty, ale i posunutím samotné vyobrazené trasy prostřednictvím uchopení a přetažení bodu vytvořeného na trase pomocí počítačové myši.



Obrázek 8.2.2 Okno s mapou vyvíjené aplikace pro evidenci jízdních tras uživateli v rámci systému OrsoftOpen společnosti Ortex s r. o.

8.3 Další vývoj aplikace pro evidenci jízdních cest

Při vývoji jakýchkoliv aplikací je dobré myslet na budoucnost a předvídat nastávající komplikace. Jedna z těch závažnějších by mohla být ukončení provozování služby Google maps API společnosti Google.com. V tomto případě by se daly alternativně využít mapové podklady služby Here.com od společnosti Nokia. Stačilo by zaměnit soubor HTML, ve kterém se nachází JavaScript API společnosti Google.com, za HTML soubor obsahující JavaScript API společnosti Nokia. Lze předpokládat, že by neměly nastat komplikace při implementaci již vytvořené metody pro potřeby samotné aplikace do JavaScriptu. Je ale nutné konstatovat, že by výsledek nebyl totožný. Neboť Here.com mapy neposkytují výběr z více možných tras pro jedno zadání. Tato vlastnost ale neomezuje funkcionalitu vyvíjené aplikace.

Další komplikací při využívání aplikace v budoucnu by mohlo být ukládání XML souboru, který obsahuje všechny důležité informace pro opětovné zobrazení mapových podkladů zadaných uživatelem. Je tedy nutné najít vhodné řešení ukládání těchto dokumentů, které by pak bylo možné znovu zobrazit.

8.4 Omezení vyvíjené aplikace pro evidenci jízdních cest

Při vyvíjení aplikací je také nutné počítat s určitými omezeními. Zvláště v případech, kdy jsou některé podklady pro vytvoření těchto aplikací k dispozici zdarma na webu.

Při vývoji aplikace evidující jízdni trasy uživatelů automobilů firem pro společnost Ortex s r. o. byly využity mapové podklady společnosti Google.com, které jsou poskytovány zdarma. Společnost Google.com uvádí, že v případě, kdyby aplikace generovala 25 000 přístupů za den více než 90 po sobě jdoucích dní, snažila by se uživatele kontaktovat za účelem zpoplatnění svých služeb. V současnosti lze ale předpokládat, že by takové zatížení systému nemělo nastat ani v těch nejkrajnějších případech. Je ale nutné s touto eventualitou počítat do budoucna.

Dalším omezením může být implementovaný „našeptávač“ pro snadné zadávání informací uživatelem. V tomto případě společnost Google.com uvádí, že bezplatně je možné využívat autocomplete API pouze na 1000 požadavků za jeden den, což představuje značné omezení služby. Nicméně případné vyčerpání limitu nemá vliv na správnou funkci této aplikace. Na druhou stranu v důsledku toho společnost Google.com přestane odesílat výsledky požadované aplikací.

Vzniklá omezení je třeba nebrat na lehkou váhu a pamatovat na ně při rozvoji systému s ohledem na volbu určitých alternativ tak, aby byl konečný uživatel maximálně spokojen.

9 Závěr

V dnešní době existuje řada poskytovatelů mapových podkladů pro možnou implementaci těchto map do aplikace zaznamenávající jízdni trasy uživatelů. Pro účely této bakalářské práce byli jednotliví poskytovatelé a výhody a nevýhody jimi poskytovaných mapových podkladů detailně rozepsány a přijatelné API služby a funkce podrobně vysvětleny.

S ohledem na všechny skutečnosti byly při vývoji aplikace zvoleny mapové podklady společnosti Google.com, díky nimž bylo dosaženo požadovaného a kvalitního výsledku. Vyvíjení samotné aplikace probíhalo prostřednictvím služeb společnosti Google.com.

Vhodným řešením se nabízela knihovna JavaFX, která nabízí třídy WebView a WebEngine. Tyto třídy slouží pro vytvoření vlastního embedded prohlížeče, který podporuje mnoho moderních funkcí. Velikou výhodou při použití knihovny JavaFX byla možnost komunikace mezi JavaScriptem a JavouFX.

Důležitým aspektem při vyvíjení aplikace je samotná platforma Java a její verze. Navrhovaná aplikace byla vyvíjena pro verzi JDK 1.6 s využitím knihovny JavaFX. Pro správnou funkci vyvíjené aplikace je také důležité, v jakém operačním systému je JavaFX využívána.

Výsledná aplikace nabízí uživateli vyhledání a změření trasy pro dvě a více adres. Výsledná vzdálenost je uživateli okamžitě zobrazena. Pro kontrolu se také nabízí možnost zobrazení mapového podkladu s nalezenou trasou. V zobrazené mapě může dále uživatel měnit výchozí/cílové adresy tak, aby co nejvíce odpovídaly skutečnosti. Uživateli se nabízí možnost vyhnout se mýtným bránám nebo dálnicím.

Testování vyvíjené aplikace probíhalo s ohledem na její funkcionalitu a její správné fungování v různých operačních systémech. Vše probíhalo v pořádku, jen knihovna JavaFX pro Win7 64bit nefungovala na Win7 32bit zcela správně. Problém byl vyřešen nahráním správné verze JavaFX pro Win7 32bit.

Nutné bylo také zhodnotit možnost vzniku budoucích omezení vyvíjené aplikace. Ty vznikly zejména stanovenými limity rozhraní API poskytované společností Google.com k jejím mapovým podkladům, které by mohly omezit služby aplikace. Vyvíjená aplikace využívá celkem tři různé API této společnosti. Nejohroženější službou je Google Places API, v jejímž případě při vyčerpání limitů sice nehrozí chyby v chodu aplikace, ale dojde

k vypnutí „našeptávače“, a aplikace tak nebude nabízet různé možnosti adres po zadání textu uživatelem.

Obecně lze říci, že hlavním cílem vyvíjené aplikace bylo vytvoření snadného uživatelského rozhraní, které by usnadnilo práci všem uživatelům, s dokonale fungujícími službami.

10 Seznam použité literatury

- [1] ASL, Josef a Roman BLAŽÍČEK. Podnikové informační systémy: podnik v informační společnosti. 2., výrazně přeprac. a rozš. vyd. Praha: Grada, 2008, 283 s. Management v informační společnosti. ISBN 978-80-247-2279-5.
- [2] DOSTÁLKOVÁ, Dana. Vývoj ERP systémů z pohledu malých a středních podniků. ERPForum [online]. 24. 4. 2012 [cit. 2014-07-17]. Dostupné z: <http://www.erpforum.cz/erp-trendy-109.html>
- [3] ORTEX S.r.o. ERP SYSTÉM ORSOFT OPEN. [online]. [cit. 2015-04-14]. Dostupné z: <http://web.ortex.cz/produkty/orsoft/>
- [4] GOOGLE. The Google Distance Matrix API. [online]. [cit. 2015-04-14]. Dostupné z: <https://developers.google.com/maps/documentation/distancematrix/>
- [5] GOOGLE. The Google Directions API. [online]. [cit. 2015-04-14]. Dostupné z: <https://developers.google.com/maps/documentation/directions/>
- [6] GOOGLE. JavaScript API v3. [online]. [cit. 2015-04-14]. Dostupné z: <https://developers.google.com/maps/documentation/javascript/tutorial>
- [7] GOOGLE. Google Places API Web Service. [online]. [cit. 2015-04-14]. Dostupné z: <https://developers.google.com/places/webservice/intro>
- [8] SEZNAM.CZ. Mapy API verze 4.8. [online]. [cit. 2015-04-14]. Dostupné z: <https://api.mapy.cz/>
- [9] NOKIA. Matrix Routing API. [online]. [cit. 2015-04-14]. Dostupné z: <https://developer.here.com/rest-apis/documentation/matrix-routing>

- [10] NOKIA. JavaScript API 3.0. [online]. [cit. 2015-04-14]. Dostupné z: <https://developer.here.com/javascript-apis>
- [11] NOKIA. HERE Routing API. [online]. [cit. 2015-04-14]. Dostupné z: <https://developer.here.com/rest-apis/documentation/routing>
- [12] What is Java technology and why do I need it?. Java [online]. [cit. 2014-08-03]. Dostupné z: https://www.java.com/en/download/faq/whatis_java.xml
- [13] TIOBE Programming Community Index for May 2013 [online]. TIOBE Software BV, květen 2013, [cit. 2013-05-10]. Dostupné z: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- [14] VEČEŘA. JSON: první kroky aneb používáme PHP a jQuery. [online]. [cit. 2014-08-01]. Dostupné z: <http://programujte.com/clanek/2008073101-json-prvni-kroky-aneb-pouzivame-php-a-jquery/>
- [15] MAVEN. Welcome to dom4j 2.0! [online]. [cit. 2015-04-24]. Dostupné z: <http://dom4j.sourceforge.net/>
- [16] ČÁPKA, David. 1. díl - Úvod do JavaFX. [online]. [cit. 2014-08-03]. Dostupné z: <http://www.itnetwork.cz/java-tutorial-uvod-do-javafx>
- [17] LEHOCKÝ, Z. – BOBEK, T. JavaScript – 1. Lekce. [online]. 2014 [cit. 13. 3. 2014]. Dostupné z: <http://programujte.com/clanek/2005123003-javascript-1-lekce>
- [18] Worker Threads and SwingWorker. Oracle [online]. [cit. 2014-08-03]. Dostupné z: <http://docs.oracle.com/javase/tutorial/uiswing/concurrency/worker.html>
- [19] Class WebEngine. Oracle [online]. [cit. 2014-08-03]. Dostupné z: <http://docs.oracle.com/javafx/2/api/javafx/scene/web/WebEngine.html>

Obsah CD

/BakalarskaPrace/Aplikace/.. Uložený adresář, obsahující zdrojové kódy prototypu aplikace. Nebyla přiložena knihovna JavaFX.



UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Rokitanského 62, 500 03 Hradec Králové, tel: 493 331 111, fax: 493 332 235

Zadání k závěrečné práci

Jméno a příjmení studenta:

Michal Renčín

Obor studia:

Aplikovaná informatika

Jméno a příjmení vedoucího práce:

Pavel Kříž

Název práce:

Integrace mapových služeb do ERP systému.

Název práce v AJ:

Map Integration into ERP System

Podtitul práce:

Podtitul práce v AJ:

Cíl práce: Analyzovat možnosti integrace veřejně dostupných mapových služeb do podnikového ERP systému, především za účelem měření dojezdových vzdáleností pro evidenci jízd. Navrhnout a implementovat vhodné řešení.

Osnova práce:

- Úvod
- ERP systémy
- Dostupné mapové služby
- Analýza a návrh
- Technologie
- Implementace
- Výsledky a doporučení
- Závěr

Projednáno dne:

Podpis studenta

Podpis vedoucího práce