
TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Zpracování GPS dat webovou aplikací

GPS data processing web application

Diplomová práce

Autor:	Bc. Lukáš Bardon
Vedoucí práce:	Ing. Igor Kopetschke
Konzultant:	Ing. Igor Kopetschke

V Liberci 12. 5. 2012



Poděkování

Poděkování patří hlavně vedoucímu mé diplomové práce Ing. Igoru Kopetschkemu a to nejen za jeho čas a trpělivost, ale také za vědomosti, které mi během konzultací předal, a bez kterých bych měl velké potíže s dokončením této diplomové práce.

Děkuji také Ing. Luboši Remplíkovi za zapůjčení *GPS* loggeru EU3C Holux M-1200E.



Abstrakt

Obsahem této práce je zpracování *GPS* dat webovou aplikací. Po přečtení dokumentu by měl čtenář mít jasný pohled na problematiku spojenou s principem *GPS* neboli Global Positioning System. Dále by měl porozumět postupu, jak získat data z *GPS* přijímače, to včetně záznamů trasy a v neposlední řadě by měl získat informace o způsobu parsování těchto dat. Tato práce taktéž pojednává o způsobu předání *GPS* dat webové aplikaci, která tyto data dále zpracovává.

Druhou částí diplomové práce je praktická ukázka mashup webové aplikace využívající výše zmíněné poznatky doplněné o standardní webové technologie a implementaci geotaggingu. Toto jádro webové aplikace by mělo sloužit jako základ pro webové aplikace u kterých je hlavním cílem zpracovávat soubory, jež obsahují *GPS* data. . Jako demonstrační aplikaci jsem vytvořil aplikaci cestopisného motorkářského portálu obohaceného o funkci vyobrazení ujeté trasy na mapě a vyobrazení otaggovaných fotografií mapě.

Abstrakt

The content of this work focuses on processing of *GPS* data using web applications. After reading the document, the reader should have a clear view on issues associated with the *GPS* principle, otherwise known as Global Positioning System. Furthermore, the reader should understand the process of getting the data from *GPS* receivers, including the routes, and finally should obtain information on how to parse the data. The work also deals with transmission of *GPS* data to the web application for further processing.

The second part of the thesis is a practical demonstration of mash-up web application, using the above mentioned knowledge together with implementation of standard Java technology and geotagging. This web application should be used as an addition of two functions for motorcycle travel portals. It is a function of showing the distance travelled on a map and illustrating the tagged photos on the map.



Obsah:

Poděkování	- 4 -
Abstrakt.....	- 5 -
Úvod	- 9 -
Slovník zkratk	- 10 -
1. GPS.....	- 11 -
1.1 Kosmický segment	- 11 -
1.1.1 Signály vysílané družicemi GPS	- 12 -
1.1.2 Družicová navigační zpráva.....	- 13 -
1.2 Řídicí segment.....	- 14 -
1.3 Uživatelský segment.....	- 15 -
1.4 Požadovaná přesnost GPS	- 15 -
1.5 Protokol NMEA	- 16 -
1.5.1 Věty ze strany mluvčího.....	- 17 -
1.5.2 Věty proprietární	- 17 -
1.5.3 Dotazovací věty	- 17 -
1.5.4 Příklad věty s definovanými datovými položkami.....	- 18 -
1.6 Souřadný systém ECEF WGS-84	- 19 -
2. GPS logger.....	- 20 -
2.1 GPX soubor	- 22 -
2.2 Parsování GPX souboru	- 23 -
2.3 Získávání dat z GPS přijímačů	- 24 -
2.3.1 Pomocí Bluetooth	- 24 -
2.3.2 Pomocí USB	- 25 -
3. Předání GPS dat webové aplikaci	- 26 -
4. Google Maps API.....	- 28 -
4.1 Typy API.....	- 28 -
4.1.1 Mapy API	- 28 -
4.1.2 AMapy API.....	- 28 -
4.1.3 OpenStreetMap	- 29 -
4.2 Dostupné verze Google Maps API	- 29 -
4.2.1 Static Maps API	- 29 -
4.2.2 Google Maps API V2	- 30 -
4.2.3 Google Maps API V3	- 30 -
4.2.4 Google maps API pro Flash.....	- 30 -
4.3 Využití Google Maps API ve webové aplikaci	- 30 -
4.3.1 Zobrazení okna mapy	- 31 -
4.3.2 Vykreslení ujeté trasy	- 33 -
4.3.3 Vykreslení fotografií na mapě	- 34 -
5. Geotagging	- 39 -



5.1	Způsoby otaggování fotografií.....	- 39 -
5.2	Otaggování fotografie pomocí Zoner Photo Studio	- 40 -
6.	Webová aplikace Cestopisný motorkářský portál.....	- 41 -
6.1	Struktura aplikace a použité technologie	- 43 -
6.1.1	Model.....	- 44 -
6.1.2	View.....	- 44 -
6.1.3	Controller	- 44 -
6.2	Aplikace z hlediska tříd	- 44 -
6.3	Aplikace z hlediska databáze	- 46 -
6.4	Praktická ukázka konkurenceschopnosti aplikace	- 47 -
	Závěr	- 49 -
	Seznam použité literatury	- 51 -



Seznam ilustrací

Obr 1.1	Rozmístění 24 družic tvořících FOC	12
Obr 1.2	Struktura družicové zprávy	13
Obr 1.3	Rozmístění stanic řídicího segmentu GPS	14
Obr 2.1	GPS logger EU3C Holux M-1200E	21
Obr 4.1	Samotná mapa	33
Obr 4.2	Vyobrazení ujeté trasy	34
Obr 4.3	Mapa obsahující ujetou trasu a fotografie pořízené po cestě...	38
Obr 6.1	Adresářová struktura třívrstvého modelu MVC.....	43
Obr 6.2	UML diagram tříd	45
Obr 6.4	Vyobrazená ujetá trasa z cestopisu nejnavštěvovanějšího českého motorkářského portálu	47
Obr 6.5	Vyobrazení ujeté trasy a otaggovaných fotografií na mapě pomocí webové aplikace Cestopisný motorkářský portál	48

Seznam tabulek

Tab 1.1	Příklad datových položek věty RCM	18
Tab 2.1	Technické údaje k zařízení EU3C Holux M-1200E.....	20
Tab 4.1	Parametry pro nastavení mapy	31
Tab 4.2	Parametry objektu google.maps.Polyline	33
Tab 4.3	Nejpoužívanější parametry objektu google.maps.Marker	37
Tab 4.4	Nejpoužívanější parametry objektu google.maps.InfoWindow	37



Úvod

V dnešní době, kdy je *GPS* systém v největším rozmachu, vlastní nějaký *GPS* přijímač téměř každý motorista, motocyklista, cyklista či pěší turista. To je způsobeno velkým počtem firem, vyrábějících *GPS* přijímače a tedy velkou konkurencí, což má za důsledek nízké pořizovací ceny těchto zařízení. Vzhledem k tomuto trendu mě napadlo obohatit cestopisy na motorkářských webových portálech o dvě funkce přinášející důkladnější prozkoumání ujeté trasy. Jedná se o funkce vyobrazení ujeté trasy na mapě a zobrazení otaggovaných fotografií na mapě.

Pro vytvoření mashup aplikace demonstrující výše zmíněné funkce, bylo potřeba osvojit si určité informace a postupy. První část tohoto dokumentu se zabývá problematikou systému *GPS*. Jedná se především o popis všech segmentů *GPS*, o komunikaci družic s *GPS* přijímači a dále o popis protokolu, se kterým systém *GPS* pracuje.

V druhé části diplomové práce jsou popsány postupy získávání dat z *GPS* přijímačů, jejich parsování a následné předání těchto dat webové aplikaci pro další zpracování. Tyto postupy jsou univerzální pro většinu typů dnes dostupných *GPS* přijímačů.

Předposlední část dokumentu obsahuje popis nejdůležitějšího nástroje mashup aplikace. Jedná se o podpůrné rozhraní pro vytváření mashupových aplikací od společnosti Google, *Google Maps API*. Jsou zde popsány funkce pro vyobrazení mapy na stránce, dále funkce pro vyobrazení ujeté trasy na mapě a v neposlední řadě nástroje pro zobrazení otaggovaných fotografií na mapovém podkladu.

Poslední část práce pojednává hlavně o informacích týkajících se samotné mashup aplikace a geotaggingu. Jsou zde popsány postupy otaggování fotografií a podrobně rozebrány jednotlivé důležité části mashup aplikace, která demonstruje výše zmíněné funkce.



Slovník zkratk

MVC	Model View Controller
GPS	Global Positioning System
API	Application Programming Interface
FOC	Plná schopnost globálního polohovacího systému
SPS	Standardní polohovací systém
PPS	Přesný polohovací systém
NMEA	National Marine Electronics Association
DOP	Dilution OF Precision
RMC	Recommended minimum Information
ECEF WGS-84	Earth Centered, Earth Fixed, World Geodetic System 1984
S-JTSK	Souřadnicový systém jednotné trigonometrické sítě katastrální
S42	Systém 1942
PDA	Personal Digital Assistant



1. GPS

Globální polohovací systém (zkr. *GPS*; angl. Global Positioning System) je družicový systém, jehož účelem je stanovení polohy a času na zemském povrchu a v přilehlém prostoru. *GPS* slouží nejen pro navigaci pozemní, letecké i námořní dopravy, ale také pro měřicí účely, geofyzikální výzkumy, mapování, lokalizaci a pro celou řadu dalších aplikací.

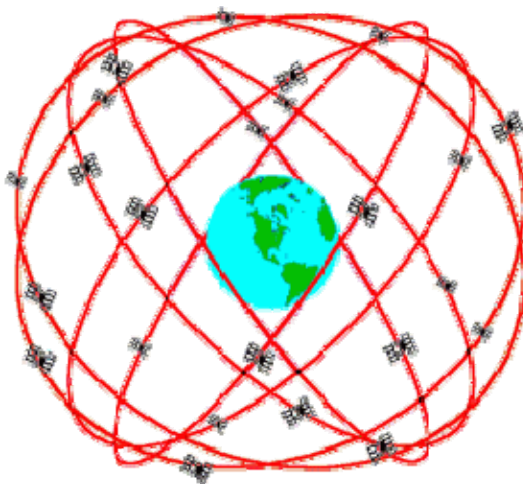
O uvedení navigačního systému do provozu se snažily Spojené státy americké již v 60. letech. Tehdy spatřil světlo světa systém s názvem **Tranzit**. Tento systém byl tvořen šesti družicemi, obíhajícími po polární oběžné dráze a třemi pozorovacími stanicemi rozmístěnými na území Spojených států amerických. Nevýhodou tohoto systému byla absence třetí souřadnice (nebylo možné tento systém použít v letecké dopravě) a dále jen občasná dostupnost signálu. Tyto skutečnosti vedly k dalšímu vývoji družicových polohovacích systémů a tak v roce 1972 vznikl systém nesoucí název **Timotion**. Tento systém byl zaměřen na vysílání přesného časového signálu.

Zkušenosti získané při vývoji a provozování těchto dvou systémů, byly základními stavebními prvky pro vývoj pozdějšího systému *GPS*. Definitivní rozhodnutí o zavedení družicového systému, který by poskytoval údaje o poloze v trojrozměrném prostoru, padlo 17. prosince 1973. Od svého počátku nese tento systém dva názvy **GPS** a **NAVSTAR**. Celý systém *GPS* je tvořen třemi základními segmenty. Jedná se o kosmický segment, řídicí segment a dále o uživatelský segment. Jednotlivé typy segmentů si podrobněji rozebereme v následujících podkapitolách.

1.1 Kosmický segment

Kosmický segment se sestává z družic umístěných na šesti kruhových drahách se sklonem 55° k polární rovině. Každá ze šesti drah má pět pozic pro umístění družic. Z toho plyne, že při dnešní konfiguraci je maximální počet *GPS* družic na oběžné dráze roven třiceti. Pátá pozice na každé dráze je rezervována pro záložní družici, která je neustále v pohotovostním stavu a je připravena nahradit funkci při poruše jiné družice. Je tedy zřejmé, že pro dosažení plné schopnosti globálního polohovacího systému *GPS* (*FOC*) je zapotřebí 24 funkčních družic.

Družice na oběžných drahách jsou ve výšce 20 200 km nad zemským povrchem a pohybují se rychlostí 11 300 km/h. Z toho plyne, že každá družice uskuteční dva oběhy kolem Země za jeden den. Tímto uspořádáním družic je uživateli poskytnut signál ze čtyř až dvanácti družic na kterémkoli místě na Zemi.



Obr 1.1: Rozmístění 24 družic tvořících FOC

1.1.1 Signály vysílané družicemi GPS

Každá družice vysílá signály na dvou nosných frekvencích. Nosná frekvence L1 vzniká vynásobením základní frekvence ($f_0 = 10,23$ MHz) hodnotou 154 a nosná frekvence L2 vzniká vynásobením základní frekvence 120. Z toho vyplývá, že nosná vlna L1 má frekvenci 1 575,42 MHz a nosná vlna L2 má frekvenci 1 227,60 MHz.

Pro modulaci nosné vlny se používá několik pseudonáhodných kódů, označovaných jako *PRN* (Pseudo Random Noise). K modulaci nosné vlny L1 se používá *P-kód*, který může být pro vojenské účely i šifrován. Tento zašifrovaný *P-kód* je pak označován jako *Y-kód*. *P-kód* a *Y-kód* patří do skupiny tzv. přesných kódů. Dále může být nosná vlna L1 modulována pomocí *C/A kódu*. Tento kód patří mezi tzv. hrubé/dostupné kódy. Pro modulaci nosné vlny L2 se používá pouze *P-kód* nebo *Y-kód*.

Je tedy zřejmé, že Signály modulující nosnou frekvenci L1, jsou označovány jako signály standardního polohového systému (*SPS*) a frekvence L2 je používána pro přesný polohovací systém (*PPS*).



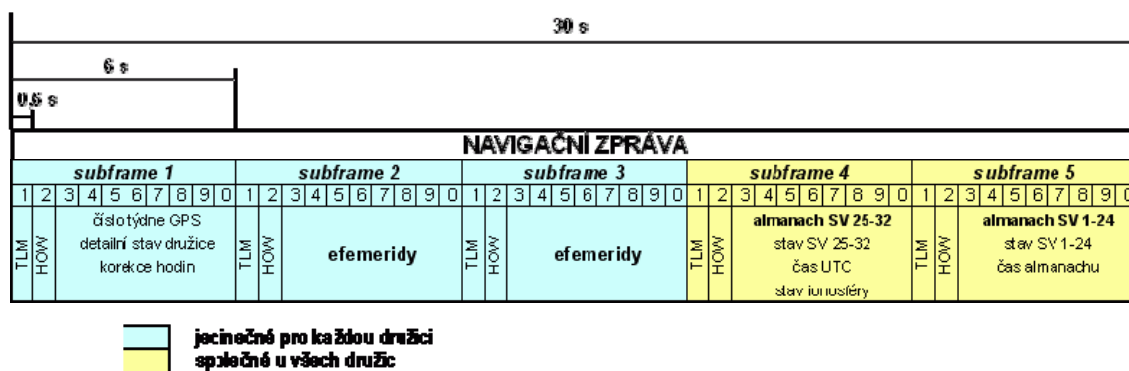
Kromě *P-kódu*, *Y-kódu* popřípadě *C/A kódu* je na nosné vlně přenášen ještě binární kód, reprezentující družicovou navigační zprávu. Binární informace je kódována pomocí fázových posunů nosných vln.

1.1.2 Družicová navigační zpráva

Tato zpráva obsahuje data týkající se stavu družice a informace o její pozici. Právě informace o pozici jsou velice důležité. Nazývají se efemeridy a tyto informace jsou dekódovány *GPS* přijímači. Zpráva je tvořena jedním rámcem (frame) o velikosti 1500 bitů. Tento rámec je dále rozdělen do pěti bloků (subframe), kde každý blok má velikost 300 bitů. Vzhledem k tomu, že družicová navigační zpráva je vysílána frekvencí 50 Hz, trvá přenos jednoho celého rámce 30 sekund a doba potřebná k odeslání jednoho bloku je tedy 6 sekund

První blok zprávy obsahuje informace o korekci hodin, druhá a třetí část je rezervována pro velice důležité efemeridy. Čtvrtý a pátý blok je využit pro přenos dat obsahujících informace o ionosféře, společné časové základně a samozřejmě almanachy (přibližné parametry oběžných drah jednotlivých družic). Výhodou znalosti aktuálních almanachů může být výrazné snížení doby potřebné k nastartování *GPS* přijímače a k získání signálu.

Každý blok ještě obsahuje přesnou časovou značku (čas odeslání bloku z družice) a také šest paritních bitů, které umožňují detekovat přenosové chyby.



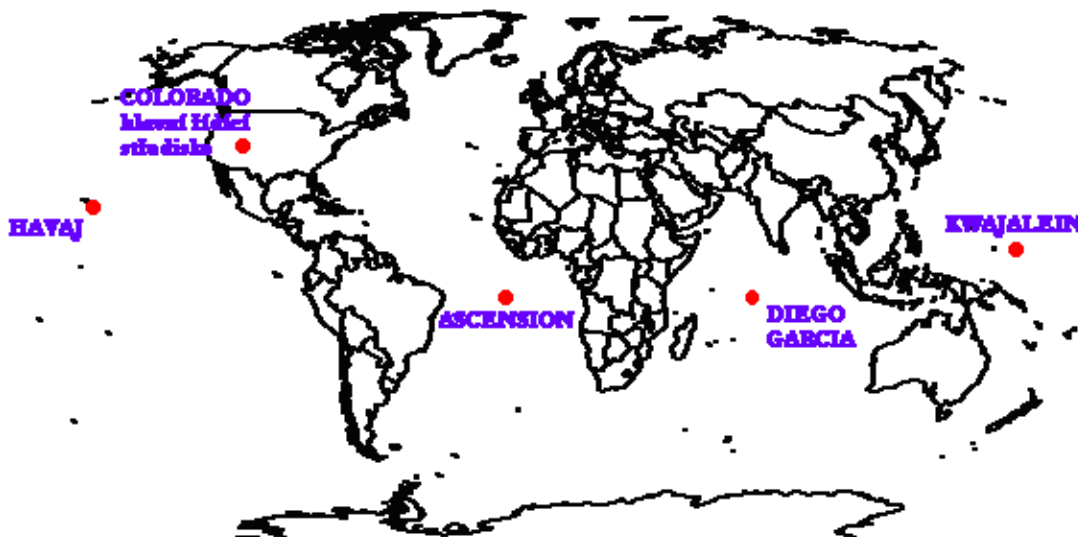
Obr 1.2: Struktura družicové navigační zprávy



1.2 Řídicí segment

Řídicí segment *GPS* se sestává ze čtyřech pozemních monitorovacích stanic a z jedné hlavní řídicí stanice a celý tento segment je zodpovědný za řízení celého globálního polohového systému. Hlavní řídicí stanice se nachází v Colorado Springs na letecké základně Falcon v USA. Další čtyři monitorovací stanice jsou rovnoměrně rozmístěny po celé Zemi a jsou orientovány poblíž rovníku. Jedna pozemní stanice se nachází na ostrovech Ascension ve středním Atlantiku, další na Diego Garcia, další na Kwajalein a poslední na Havaji.

Úkolem monitorovacích stanic je měřit signály, které družice posílají a tyto získané informace přeposílají do hlavní řídicí stanice. Zde jsou na základě provedených měření vypočítány přesné efemeridy a korekce hodin. Tyto výsledky jsou předány vysílací stanici. Tyto stanice pak alespoň jednou denně posílají efemeridy a informace pro nastavení hodin jednotlivým družicím. Družice pak mohou odeslat pomocí radiových signálů přesné efemeridy a přesný čas do *GPS* přijímačů.



Obr 1.3: Rozmístění stanic řídicího segmentu *GPS*



1.3 Uživatelský segment

Uživatelský segment je složen z následujících prvků:

- GPS přijímače
- Uživatelé
- Vyhodnocovací nástroje
- Postupy

Žádoucí akcí *GPS* přijímačů je, aby v daných časových intervalech provedly na základě přijatých signálů předběžné výpočty polohy, rychlosti a času. Abychom získali všechny potřebné souřadnice (x , y , z a t), je zapotřebí přijímat signál alespoň ze čtyřech družic. Tyto *GPS* přijímače jsou využívány např. pro navigaci, stanovení aktuální polohy, k měření, nebo pro stanovení přesného času.

Jelikož jsou *GPS* přijímače vyráběny pro letadla, lodě, pozemní dopravu, kosmické objekty ale také i v ručním provedení, je tedy jasné, že navigace ve třírozměrném prostoru je základní úlohou *GPS*. Pro opravdu přesné určení polohy je zapotřebí referenčních přijímačů umístěných na místech známé pozice. Tyto referenční stanice pak mohou získávat korekce pro opravu výpočtů z mobilních stanic.

1.4 Požadovaná přesnost GPS

Požadavky na přesnost *GPS* jsou uvedeny v tzv. Federálním radionavigačním plánu. Tyto požadavky se dělí na dvě základní úrovně poskytovaných služeb:

- Standardní polohovací systém
- Přesný polohovací systém

SPS mohou uživatelé využívat po celém světě bez omezení a zdarma. Většina volně dostupných *GPS* přijímačů podporuje příjem signálů v rámci této služby. Z hlediska bezpečnosti, má správce a provozovatel *SPS* možnost záměrně snížit přesnost těchto signálů pomocí tzv. selektivní dostupnosti (*S/A*).

Požadavky na přesnost *SPS*:

- Horizontální přesnost do 100 m
- Vertikální přesnost do 156 m
- Přesnost času do 167 ns



PPS mohou využívat pouze oprávnění uživatelé, kteří musí vlastnit speciální *GPS* přijímače, dále pak na základě povolení vlastní kryptografické zařízení a k tomu odpovídající klíče. Mezi tyto uživatele patří některé sektory vlády USA, dále armáda USA a spřátelené armády, nebo také civilní obyvatelé se zvláštním povolením od vlády USA.

Podmínky pro udělení výjimky:

- Národní zájmy USA
- Schopnost uživatele zajistit utajení
- Nemožnost použití jiného způsobu lokace

Požadavky na přesnost *PPS*:

- Horizontální přesnost do 17,8 m
- Vertikální přesnost do 27,7 m
- Přesnost času do 100 ns

1.5 Protokol NMEA

Samotné *GPS* přijímače umějí pouze přijímat data odeslaná z navigačních družic, nedokážou je ale zpracovat a zobrazit výsledky. K těmto úkolům slouží jiné části zařízení. Těmito zařízeními myslíme různé mikročipy atd. *GPS* přijímač komunikuje s těmito zařízeními pomocí protokolu *NMEA*.

Asociace *NMEA* (National Marine Electronics Association) na svých internetových stránkách uvádí, že standart *NMEA* je dokument podléhající copyrightu a lze jej zakoupit pouze od asociace *NMEA*.

Tento protokol má přenosovou rychlost 4800 bps. Na každé datové lince může komunikovat pouze jeden mluvčí (talker) a jeden nebo více posluchačů (listeners). Veškerá data odesílána pomocí tohoto protokolu jsou reprezentována ve formě vět (sentences).

Věty mohou být trojího druhu:

- Věty ze strany mluvčího (talker sentences)
- Proprietační věty (proprietary sentences)
- Dotazovací věty (query sentences)



Věta protokolu může obsahovat pouze znaky *ASCII* a znaky konce řádku, kterými jsou *<CR>* a *<LF>*. Samozřejmě, že každá věta má i svůj začátek. V tomto případě je začátek věty reprezentován znakem \$ (dolar). Každá věta může obsahovat nejvýše 80 znaků a samozřejmě znaky pro začátek a konec věty. Věta může mít tedy velikost maximálně 83 bytů.

1.5.1 Věty ze strany mluvčího

Obecný formát věty má následující tvar: $\$ttss,d1,d2,\dots\langle CR\rangle\langle LF\rangle$

První dvě písmena, následující bezprostředně po znaku \$, představují identifikátor mluvčího. Další tři následující znaky reprezentují identifikátor věty. Následná sekvence znaků odpovídá různým datovým položkám a tato data jsou od sebe oddělena čárkou. Dále věta může disponovat kontrolním součtem, ten však není ve větě povinný. Na konci každé věty je opět očekáván znak konce věty *<CR><LF>*. Význam jednotlivých datových položek může být různý. Význam je jednoznačně definován typem dané věty, tedy identifikátorem *sss*.

1.5.2 Věty proprietární

Jedná se o výčet vět nepříliš používaných. Proprietární věty vlastně umožňují výrobci definovat vlastní věty. Tyto věty pak začínají dvojicí symbolů *\$P*. Dále následuje tříznakový identifikátor výrobce a následně už jen jednotlivé datové položky, které odpovídají přáním výrobců.

1.5.3 Dotazovací věty

Dotazovací věty umožňují posluchačům podat žádost mluvčímu o zaslání určité věty.

Obecný formát má následující tvar: $\$tllQ,sss\langle CR\rangle\langle LF\rangle$. Ten kdo žádá o zaslání věty, je v obecném formátu dotazovací věty reprezentován prvními dvěma znaky umístěnými bezprostředně za symbolem \$. Další dva znaky reprezentují dotazovaného. Dalším znakem je vždy písmeno *Q*, které značí, že se jedna o větu dotazovací. Poslední v sekvenci znaků zbývá už jen třípísmenná datová položka (*sss*) označující, o jaký typ věty se vlastně žádá.



Co se týče výčtu existujících vět v *GPS*, můžeme hovořit o opravdu velkém množství. Avšak z tohoto nepřehledného množství vět můžeme vybrat čtyři věty, které jsou nejdůležitější a zároveň nejvíce používané v dnešních *GPS* přístrojích. Jedná se o věty :

- *GSA* – aktivní satelity a *DOP* (Dilution Of Precision)
- *RMC* Minimální doporučená informace pro navigaci (Recommended minimum Information)
- *GSV* - informace o družicích (Satellites in View)
- *GGA* – zeměpisná délka a šířka, geodetická výška, čas určení souřadnic

1.5.4 Příklad věty s definovanými datovými položkami

V tomto příkladě se bude jednat o větu ze strany mluvčího. Mluvčí bude vyvolávat větu typu *RMC*. Bude tedy podávat minimální doporučené informace pro navigaci.

Věta: *\$GPRMC,170138.615,A,4912.2525,N,01635.0378,E,0.04,16.43,280705,,*32*

#	Formát	Příklad	Komentář
1	hhmmss.ss	170138.615	Čas (UTC)
2	c	A	Status (A=OK)
3	ddmm.mmmm	4912.2525	Zeměpisná šířka
4	c	N	Identifikátor N/S
5	ddmm.mmmm	01635.0378	Zeměpisná délka
6	c	E	Identifikátor E/W
7	d.d	0.04	Vodorovná rychlost
8	d.d	16.43	Kurz pohybu
9	ddmmyy	280705	Datum
10	d.d	N.A.	Magnetická deklinace
11	c	N.A.	Identifikátor E/W
12	*xx	32	Kontrolní součet

Tab 1.1: Příklad datových položek věty RCM



1.6 Souřadný systém ECEF WGS-84

Jak už bylo řečeno, pro výpočet vlastní polohy *GPS* přijímače používá zařízení signál ze všech viditelných družic. Každá družice vysílá informace o své poloze ve formě souřadnic. Tyto souřadnice jsou vztaženy právě k tomuto pravoúhlému souřadnému systému. *ECEF WGS-84* (Earth Centered Earth Fixed, World Geodetic System 1984). Tento systém je definován elipsoidem, který byl v roce 1984 stanoven jako referenční. Počátek souřadnicového systému byl umístěn právě do středu tohoto elipsoidu. Počátek soustavy je tedy reprezentován bodem $[0,0,0]$. Rozvržení os soustavy je následující. Osa z je totožná s osou rotace, osa x prochází průsečíkem rovníku a nultého poledníku a osa y je zvolena tak, aby systém x, y, z byl pravotočivý.

Standardně se veškeré potřebné výpočty provádějí v této souřadné soustavě a výsledná pozice *GPS* přijímače je taktéž určena v tomto systému. Pro běžné použití se tyto výsledky přepočítávají do běžných zeměpisných souřadnic, a sice do zeměpisné délky a šířky.

Třetí souřadnice reprezentuje vzdálenost od povrchu referenčního elipsoidu. Některé *GPS* přijímače znají rozdíl mezi povrchem referenčního elipsoidu a místní střední výškou mořské hladiny. Není tedy potřeba přičítat či odčítat konstanty.

Kromě souřadného systému *ECEF WGS-84* existují ještě další, tzv. lokální souřadné systémy. Mezi ně patří např. *S-JTSK* (souřadnicový systém jednotné trigonometrické sítě katastrální) nebo *S42* (Systém 1942).



2. GPS logger

GPS logger je příruční zařízení, které přijímá *GPS* signály z viditelných družic. Toto zařízení je schopno v pravidelných časových intervalech zaznamenávat informace o své poloze (zeměpisná, šířka, délka, čas a výškový profil).

Pokud *GPS* logger disponuje nějakým *I/O* portem, pak zaznamenaná data mohou být dále zpracovávána různými aplikacemi. Jednou z aplikací zpracovávající tento typ dat je právě webová aplikace, která byla vytvořena jako praktická část této diplomové práce.

Abych mohl otestovat funkčnost aplikace, bylo zapotřebí nasbírat potřebná data. Pro získání těchto dat mi byl zapůjčen *GPS* logger EU3C Holux M-1200E. Toto zařízení disponuje interní pamětí 4MB, což umožňuje uchovat až 200 000 záznamů o dané poloze, datu, času, rychlosti a o výškovém profilu. Tento modul lze, díky bluetooth rozhraní, použít jako *GPS* přijímač pro mapové aplikace, často využívané v mobilních telefonech či v *PDA* zařízeních.

GPS přijímač	MediaTek MT3329
Počet kanálů	66
Přesnost GPS pozice	± 3 m
Přesnost rychlosti	$\pm 0,1$ m/s
Studený start	36 s
Teplý start	33 s
Horký start	1 s
I/O rozhraní	Bluetooth, USB
Interní paměť	4 MB

Tab 2.1: Technické údaje k zařízení EU3C Holux M-1200E

Toto zařízení je tedy schopno pracovat ve dvou režimech. Buď jako klasický navigátor nebo jako trasovač. Zařízení taktéž disponuje *WAAS/ EGNOS /MSAS* demodulátorem. Tento demodulátor slouží k dobrému příjmu signálu v hustě zastavěné oblasti, v údolích či v dalších místech se špatnou dostupností signálu. Tento *GPS* logger pracuje s datovým protokolem *NMEA0183* verze 3.01, což zaručuje kompatibilitu



s většinou navigačních softwarů. Svými malými rozměry a nízkou hmotností je vhodný jak pro pěší turistiku, cyklistiku tak i pro motocykly či automobily. Další výhodou je velká výdrž baterie, která se pohybuje okolo 15 hodin při současné navigaci i záznamu trasy.



Obr 2.1: GPS logger EU3C Holux M-1200E



2.1 GPX soubor

GPX soubor je *XML* soubor obsahující nejdůležitější informace pro zpracování trasy nějakou softwarovou aplikací. Tento *XML* soubor má přesně definovanou formu. První element by měl reprezentovat hlavičku *GPX* souboru. Tato hlavička obsahuje informace o znakovém kódování, informace o transformacích *GPX* souboru a také údaje o výrobci trasovacího zařízení. Dalším elementem bývají *metadata* obsahující informace o názvu dané trasy a datum vytvoření *GPX* souboru. Dále následuje povětšinou velice objemný element, jehož podelementy reprezentují trasované body. Každý trasovaný bod nese informaci o zeměpisné výšce a šířce, o času zaznamenání daného bodu a dále také informaci o výškovém profilu.

Když vezmeme v úvahu, že většina trasovacích zařízení se snaží trasovat každou sekundu, pak je možné, že *GPX* soubor může obsahovat obrovský počet záznamů. Pokud nepožadujeme příliš velkou přesnost, například při vykreslení trasy, je možné frekvenci trasování změnit na větší časový interval. To je možné pomocí různých softwarů, které jsou povětšinou dodávány výrobcem spolu s trasovacím *GPS* přijímačem.

```
<gpx xmlns="http://www.topografix.com/GPX/1/1"
creator="MapSource 6.12.4" version="1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.topografix.com/GPX/1/1
http://www.topografix.com/GPX/1/1/gpx.xsd">
<metadata>
  <link href="http://www.garmin.com"><text>Garmin
International</text></link>
  <time>2008-04-28T15:58:51Z</time>
</metadata>
<trk>
  <trkseg>
    <trkpt lat="49.9327271" lon="17.2178347">
      <ele>706.7062988</ele>
    </trkpt>
    <trkpt lat="49.9327773" lon="17.2177797">
      <ele>708.6289063</ele>
    </trkpt>
    <trkpt lat="49.9326892" lon="17.2154824">
      <ele>716.3195801</ele>
    </trkpt>
  </trkseg>
</trk>
```



```
</trk>  
</gpx>
```

Kód 2.1: Příklad krátkého GPX souboru

2.2 Parsování GPX souboru

Při čtení samotného *GPX* souboru vidíme obrovské množství dat, ale povězte jako navigátor řidiči, že se má přemístit z místa které má souřadnice 49.9327271, 17.2178347, na místo o souřadnicích 49.9326892, 17.2154824. Je tedy jasné, že data z *GPX* souboru musejí být dále zpracována nějakou aplikací, která dokáže tyto data rozparsovat a dále zpracovávat požadovaným způsobem vedoucím k požadovanému výsledku, kterým může být např. zobrazení ujeté trasy na mapě.

Pokud chceme například parsováním *GPX* souboru získat pouze souřadnice polohy, pak je třeba se při parsování dotazovat na hodnoty atributů *lat* a *lon* elementu *<trkpt>*.

Vzhledem k tomu, že ve webové aplikaci Cestopisný motorkářský portál, což je zároveň praktickou částí této diplomové práce, bylo potřeba vyobrazit ujetou trasu na Google mapách, bylo potřeba parsovat *GPX* soubor a získat tak výčet všech souřadnic z *GPX* souboru a uložit je do pole pro další zpracování.

Jak už bylo jednou řečeno, *GPX* soubor je generován *GPS* přijímačem v podobě *XML* souboru. Pro parsování tohoto dokumentu jsem využil základní PHP funkci `simplexml_load_file()`. Této funkci předávám jako vstupní parametr daný *GPX* soubor.

```
public function parseXML($soubor){  
    $pole_lat = array();  
    $pole_lon = array();  
    $xml = simplexml_load_file("images/$soubor");  
    foreach ($xml->rte->rtept as $souradnice){  
        $pole_lat[] = $souradnice['lat']; //obsahuje  
        první souradnici  
        $pole_lon[] = $souradnice['lon']; //odsahuje  
        druhou souradnici  
    }  
    return array('lat' => $pole_lat, 'lon' => $pole_lon);  
}
```

Kód 2.2: Parsování souřadnic z GPX souboru



Aby bylo možné dále zpracovávat GPS data pomocí Google Maps API, je zapotřebí, aby byla data uložena do dvou polí. První pole (`$pole_lat[]`) obsahuje výčet prvních souřadnic a druhé pole (`$pole_lon[]`) obsahuje výčet druhých souřadnic.

Tato pole naplníme pomocí cyklu `foreach`, ve funkci `parseXML()`, který prochází všechny elementy `rtept` a do výše zmíněných polí ukládá hodnoty jeho atributů `lat` a `lon`.

2.3 Získávání dat z GPS přijímačů

Dnes již každý *GPS* přijímač má možnost trasování. To znamená, že v každém časovém intervalu přijímač zaznamená do trasového souboru informace o aktuální poloze. Časový interval zaznamenávání se liší buď přístroj od přístroje, nebo jej lze uživatelsky nastavit. U standardních *GPS* přijímačů se časový interval trasování nastavuje přímo v menu přístroje nebo pomocí dodávaného softwaru, zatímco u *GPS* loggeru, lze tento interval nastavit pouze pomocí softwaru. Trasu lze také zaznamenávat na základě délkového intervalu.

Každé *GPS* zařízení je vybaveno rozhraním pro připojení s počítačem. Dnes se jedná především o rozhraní *USB* a *Bluetooth*. Propojení *GPS* zařízení s *PC* je důležité z hlediska aktualizací mapového vybavení, přenosu trasovacích souborů, přidávání různých funkcí atd.

2.3.1 Pomocí Bluetooth

Získávání dat z *GPS* přijímače vybaveným rozhraním *Bluetooth* je velice jednoduché. Ve správci souborů nalezneme požadovaný soubor a ten je následně odeslán na *PC* (*PC* musí být taktéž vybaveno funkcí *Bluetooth*). Málokteré zařízení má však možnost využití správce souborů, tuto funkci podporují hlavně novější mobilní telefony a *PDA* zařízení.

U ostatních *GPS* přijímačů slouží *Bluetooth* rozhraní především k propojení se zobrazovacími zařízeními. Jedná se například o propojení mobilního telefonu vybaveného mapovými podklady (neobsahuje *GPS* modul) s *GPS* loggerem.



2.3.2 Pomocí USB

USB rozhraní je již standardním vybavením *GPS* zařízení. *GPS* přijímač se k *PC* připojuje pomocí propojovacího *USB* kabelu. Některé *GPS* přijímače podporují *mass storage*, z čehož vyplývá, že pro získání dat z *GPS* zařízení nepotřebujeme speciální ovladač. Pokud zařízení danou funkci nepodporuje, je zapotřebí nainstalovat požadovaný ovladač.

Pro získání požadovaných dat je v obou případech zapotřebí software, který nám umožní stáhnout ze zařízení potřebná data. Tento software je povětšinou dodáván jako příslušenství k *GPS* přijímači.

K *GPS* loggeru, který jsem měl k dispozici, je dodáván software SW HOLUX ezTour for Logger. Tento program má kromě získávání dat z *GPS* loggeru více funkcí. Jedná se především o nastavení dopravního prostředku (*car*, *walk*, *bike*), nastavení časového nebo délkového intervalu pro trasování, zobrazení tras na mapě, převedení trasovacího souboru do různých formátů (*GPX*, *kml*, *csv* nebo *NMEA*).

Kromě těchto funkcí dokáže program SW HOLUX ezTour for Logger sesynchronizovat data z trasovacího souboru s fotografiemi. Tím se do *EXIFU* fotografie přidá informace o poloze, kde byla fotografie pořízena. Avšak tuto funkci plní mnohem lépe jiné programy jako například Zoner Photo Studio.



3. Předání GPS dat webové aplikaci

Před tím, nežli je *GPX* soubor rozparsován a dále zpracován webovou aplikací, je třeba tento soubor nejprve aplikaci předat. Jedná se tedy o *upload* souboru na server. *Upload* souborů na server se zpravidla řeší správným vytvořením formuláře, starajícího se o odeslání souboru. Tento formulář musí splňovat určité náležitosti:

- Formulář musí obsahovat element pro upload souboru na server
- Soubor ukládaný na server musí mít unikátní název
- Je potřeba ošetřit, že se na server nahrává soubor s požadovanou koncovkou (zamezení možnosti nahrát na server nežádoucí script)
- Formulář musí být odeslán metodou POST

V jádru aplikace se o předání *GPX* souboru a dalších informací o trase aplikaci stará formulář `IncludeDriveForm()`.

```
protected function createComponentIncludeDriveForm() {
    $countries =
array('cs'=>'Česká republika', 'sk'=>'Slovensko', 'at'=>'Rakou
sko', 'it'=>'Itálie', 'f'=>'Francie', 'ch'=>'Švýcarsko', 'jiné'
);

    $form = new Form;
    $form->addText('nazev', 'Název trasy:');
    ->setRequired('Prosím název trasy');
    $form->addSelect('zeme', 'Země:', $countries);
    $form['zeme']->setDefaultValue('0');
    $form->addText('lokalita', 'Lokalita:');
    ->setRequired('Prosím lokalitu');
    $form->addTextArea('popis', 'Popis trasy:');
    ->addRule(Form::MAX_LENGTH, 'Poznámka je příliš
dlouhá', 1000)
    ->setRequired('Prosím popis');
    $form->addUpload('gpx', 'GPX Soubor:');
    ->addRule(callback($this, 'fileExtensionValidator'),
"Neplatná přípona souboru. Soubor musí mít příponu 'xml
nebo gpx'", array('xml', 'gpx'));
    $form->addSubmit('submit', 'Vložit');
    $form->onSuccess[] = callback($this,
'includeDriveFormSubmitted');
    return $form;
}
```

Kód 3.1: Formulář pro zadání nové trasy



Z ukázky kódu (*Kód 3.1*) je patrné, že funkce `addUpload()` se stará o vytvoření komponenty formuláře pro upload *GPX* souboru na server a funkce `fileExtensionValidator()` ošetřuje aby se na server nahrály pouze soubory s povolenou koncovkou.

Po odeslání formuláře se zavolá funkce pro obsluhu tohoto formuláře. Funkce `includeDriveFormSubmitted()` je zodpovědná za předání dat z formuláře do databáze a za upload *GPX* souboru na server. Před uložením dat do databáze a nahrání souboru na server je zapotřebí přejmenovat nahrávaný soubor unikátním jménem souboru. Unikátní název získáme pomocí funkce `uniqid()`. Pak už je možné nahrát soubor na server a data z formuláře společně s unikátním názvem souboru předat jako vstupní parametry funkci `vlozit()`. Tato funkce zajišťuje uložení dat o trase do databáze. Pokud zadaná data projdou validačními pravidly formuláře, budou tato data uložena do databáze a dojde k přesměrování na stránku jež obsahuje jednoduchý formulář pro upload ottagovaných fotografií.

```
public function includeDriveFormSubmitted($form) {
    $date=time();
    if($_SESSION['id']){
        $user_id = $_SESSION['id'];
    }
    $file = $form['gpx']->value;
    if($file->isOk()) {
        $name = uniqid().'xml';
        $file->move($this->context->params['wwwDir'] .
'/images/'.$name);
    }
    $_SESSION['gpx_name'] = $name;
    $_SESSION['values'] = $form -> values;
    $model = $this->models->vlozit;
    $_SESSION['last_id'] = $model-
>vlozit($_SESSION['values'],$_SESSION['gpx_name'],$_SESSION
['id'],$date);
    $this -> redirect('foto');
}
```

Kód 3.2: Obsloužení formuláře includeDriveForm



4. Google Maps API

Aby bylo možné *GPS* data z *GPX* souboru nějakým způsobem prezentovat, bylo potřeba do webové aplikace přidat okno s mapou. Dnes už je na internetu více serverů, které poskytují mapové služby. Mezi nejpopulárnější v České republice patří:

- maps.google.cz
- mapy.cz
- amapy.cz
- atd.

Každý z těchto serverů poskytl uživatelům programové rozhraní (dále jen *API*), které umožňuje uživatelům využívat mapových služeb ve svých vlastních aplikacích. Tímto krokem daly výše zmíněné portály, možnost rozmachu mashupovým aplikacím.

V následujících podkapitolách si rozebereme vlastnosti jednotlivých *API* a pak už se budeme věnovat pouze a jen *Google Maps API*.

4.1 Typy API

Jak už bylo jednou zmíněno, maps.google není jediný server poskytující *API* pro vývojáře mashupových aplikací. Proto si rozeberme jednotlivé *API* zvlášť.

4.1.1 Mapy API

Toto *API* poskytuje firma seznam.cz, která disponuje mapovými podklady na stránkách mapy.cz. U tohoto *API* může být považováno za nevýhodu, že mapy.cz poskytují mapové podklady pouze pro Evropu. Naopak výhodou může být perfektní zmapování České republiky a manuál k *API* v českém jazyce.

4.1.2 AMapy API

AMapy API je programové rozhraní pro mapové podklady amapy.cz. Tento server je majetkem Centrum holdings. Za velkou nevýhodu je považováno, stejně jako u předešlého *API*, poskytování mapových podkladů pouze pro Evropu a nepříliš velká používanost tohoto serveru. Naopak mezi výhody bych opět zařadil manuál pro *API* v českém jazyce.



4.1.3 OpenStreetMap

Jako další alternativou *API* pro mapové podklady je *OpenStreetMap*. Jedná se o jedinou open.source alternativu. Tato služba poskytuje podobné možnosti mapování jako výše zmíněné služby, avšak *API* u této služby se od ostatních přeci jen trochu liší.

Ostatní služby mají svá *API* založena především na JavaScriptu, kdežto *OpenStreetMap* má své *API* založeno na tzv. *RESTfull API*, které poskytuje přístup k *RAW* mapovým datům.

4.2 Dostupné verze Google Maps API

K čemu je vlastně *Google Maps API*? Pomocí tohoto *API* je možné do aplikace vložit okno s mapou a využívat tak veškeré mapové služby, které Google Maps poskytuje. Díky celosvětovému zmapování a velice obsáhlým knihovnám se *Google Maps API* stává velice silným nástrojem pro mashup aplikace ať už webové či desktopové.

Google Maps API je k dispozici v několika verzích. Jednotlivé verze se od sebe liší ve výčtu různých objektů, funkcí a možností využití těchto funkcí v mashupových aplikacích. Jednotlivé verze *Google Maps API* si podrobněji rozebereme v následujících podkapitolách.

4.2.1 Static Maps API

Toto *API* slouží hlavně pro generování obrázků s mapou, které odpovídají zadaným parametrům. Využití tohoto *API* by mohlo být například na webových stránkách firmy v sekci „Kde nás najdete“. V těchto případech nám bohatě postačí statická mapa, jelikož nepotřebujeme s mapou nějakým způsobem pohybovat či zoomovat.

Rozhraní pro toto *API* nalezneme na adrese :
<http://maps.google.com/maps/api/staticmap?parameters> ,kde za parameters dosazujeme požadované parametry.



4.2.2 Google Maps API V2

Verze 2 nám již poskytuje velké množství knihoven pro udělení dynamiky dané mapě. S mapou je pak možné pohybovat, zoomovat, dynamicky na mapu umisťovat markery atd.

Nevýhodou verze 2 je její rychlost a poměrně velká velikost na zpracování. Další nevýhodou je nutnost obstarat si API KEY, který je nutný si vygenerovat na oficiální stránce *Google Maps API*.

4.2.3 Google Maps API V3

Ve verzi 3 nám odpadá povinnost generovat API KEY. Verze 3 je také mnohem rychlejší a její knihovny nezabírají tolik místa. Vzhledem k tomu, že verze 3 je stále ještě ve vývoji, nejsou některé knihovny ještě dodefinovány. Například metoda `GDownload()`, která ve verzi 2 fungovala bez problémů, ve verzi 3 úplně chybí a tak jsem v praktické části diplomové práce musel využít pomocného skriptu `Util.js`, ale o tom až později.

Lze tedy říci, že až na malé rozdíly, verze 3 plní stejnou funkci jako verze 2.

4.2.4 Google maps API pro Flash

Toto *API* bylo vytvořeno pro flashové aplikace a nabízí podobné možnosti jako verze 3.

4.3 Využití Google Maps API ve webové aplikaci

Pro demonstraci usilovaných cílů v této diplomové práci byla vytvořena webová mashupová aplikace využívající *Google Maps API* verze 3. Verzi 3 jsem zvolil, jelikož u této verze odpadá povinnost generovat API KEY. Toto *API* je založené na JavaScriptu a v následujících podkapitolách budou popsány metody pro zobrazení okna mapy, vykreslení ujeté trasy a dále také zobrazení otaggovaných fotografií.

Velkou oporou při práci s *Google Maps API* mi byla perfektně zpracovaná dokumentace od společnosti Google. V této dokumentaci jsou podrobně popsány veškeré objekty a metody, které *Google Maps API* poskytuje.



4.3.1 Zobrazení okna mapy

Abychom mohli používat mapové podklady od společnosti Google ve svých aplikacích, je potřeba načíst *Google Maps API*. To provedeme tak, že do hlavičky HTML dokumentu vložíme JavaScriptovou knihovnu a to tak, že do tagu `<script>` přidáme parametr `src`, jehož hodnotou bude URL adresa s přesně definovanými parametry.

```
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?sensor=false&region=CZ">
</script>
```

Kód 4.1: Načtení Google Maps API

Samotné zobrazení okna mapy je poměrně snadné. Je potřeba této mapě nadefinovat parametry, které jsou podstatné pro správné vyobrazení. Dále je potřeba v HTML dokumentu vytvořit `div` s určitým `id` na místě, kde požadujeme okno mapy vložit. Toto `id` je pak předáno konstruktoru mapy.

```
<div id="mapa" style="width: 500px; height: 500px"></div>
```

Kód 4.2: Na místě tohoto divu bude vloženo okno mapy

Samotná inicializace okna mapy spočívá v definování parametrů pro načtení okna mapy v definici konstruktoru. Nejdůležitější parametry jsou přehledně uvedeny v tabulce Tab 4.1

Zoom	Úroveň defaultního přiblížení
Center	Defaultní pozice středu mapy
mapTypeId	Typ zobrazované mapy: ROADMAP SATELLITE HYBRID TERRAIN

Tab 4.1: parametry pro nastavení mapy



Po nadefinování parametrů nastavení okna mapy je potřeba definovat konstruktor. Vytváříme objekt typu `google.maps.Map`. V konstruktoru předáváme dva parametry, kde první parametr označuje místo v HTML dokumentu, kde bude okno mapy vloženo. Druhým parametrem je záznam dříve definovaných parametrů pro nastavení okna mapy.

```
var latlng = pole[0];
    var myOptions = {
        zoom: 8,
        center: latlng,
        mapTypeId:
google.maps.MapTypeId.ROADMAP
    };
    map = new
google.maps.Map(document.getElementById("mapa1"),
    myOptions);
```

Kód 4.3: nastavení okna mapy a definice konstruktoru

Pro zobrazení okna mapy je zapotřebí kód 4.3 zapouzdřit do nějaké funkce (V mém případě jsem funkci pojmenoval `function initialize()`). Pokud je funkce správně nadefinována, nic nám nebrání, abychom umožnili načtení okna mapy. Načtení se provede zavoláním funkce `function initialize()`. Funkci budeme volat pomocí tagu `<body>` a jeho parametru `onload`.

```
<body onload="initialize()" onunload="GUnload()">
```

Kód 4.4: volání funkce initialize() v tagu body

Výsledkem tohoto snažení bude vyobrazení okna s mapovými podklady od společnosti Google. Toto okno se objeví při načtení dané stránky a to v místě, kde je definovaný výše zmiňovaný `div`.



Obr 4.1: Samotná mapa

4.3.2 Vykreslení ujeté trasy

V kapitole 2 tohoto dokumentu bylo popsáno parsování *GPX* souboru. Výsledkem parsování souboru jsou dvě pole s mnoha souřadnicemi. Z těchto souřadnic bylo třeba vytvořit křivku vykreslenou na mapě a tato křivka by měla zobrazovat ujetou trasu. O vytvoření této křivky se stará objekt typu `google.maps.Polyline`, jehož parametry jsou přehledně uvedeny v tabulce Tab 4.2.

Path	Seznam bodů, které křivka spojuje
<code>strokeColor</code>	Barva křivky
<code>strokeOpacity</code>	Průhlednost křivky
<code>strokeWeight</code>	Tloušťka křivky

Tab 4.2: Parametry objektu `google.maps.Polyline`

O samotné vykreslení křivky na mapu se stará funkce `setMap()`. Funkce `setMap()` je funkce objektu `google.maps.Polyline`. Této funkci je předáván jediný parametr a tento parametr musí být objekt typu `google.maps.Map`.



```
var fp = new google.maps.Polyline ({  
    path: pole,  
    strokeColor: "#ff0000",  
    strokeOpacity: 1.0,  
    strokeWeight: 2  
});  
fp.setMap(map1);
```

Kód 4.5: Vykreslení ujeté trasy na mapě

Vyobrazení ujeté trasy na mapě je pak patrné na obrázku Obr 4.2.



Obr 4.2: Vyobrazení ujeté trasy

4.3.3 Vykreslení fotografií na mapě

Další nedílnou součástí této diplomové práce bylo využití geotaggingu. Ve zkratce se geotagging zabývá otaggováním fotografií, což znamená, že do EXIFU fotografie jsou přidány informace o pozici, kde byla fotografie pořízena. Podrobnější informace o Geotaggingu jsou obsaženy v kapitole 5.

Geotagging je v praktické části této diplomové práce využit formou vyobrazení pořízených fotografií z cest na mapě. Abychom mohli otaggované fotografie vyobrazit na mapě, je potřeba několika podpůrných nástrojů. Jedná se především o *Google Maps API* a nástroje pro získání GPS dat z EXIFU nahraných fotografií.



K získání souřadnic z EXIFU fotografie dochází již při obsluze odeslání formuláře includeFotoForm pomocí funkce `exif_read_data()`. Této funkci se jako vstupní parametr předá nahraná fotografie. Funkce `exif_read_data()` vrátí pod klíči `['GPSLongitude']`, `['GPSLongitudeRef']`, `['GPSLatitude']` a `['GPSLatitudeRef']` potřebná data.

```
public function includeFotoFormSubmitted($form){
    $file = $form['foto']->value;
    if($file->isOk()) {
        $name = uniqid().''.jpg';
        $file->move($this->context->params['wwwDir'] .
            '/images/foto/' . $name);
    }
    $exif = exif_read_data($this->context->
        >params['wwwDir'] . '/images/foto/' . $name);
    foreach($exif["GPSLongitude"] as &$ref){
        $hodnoty = explode('/', $ref);
        $hodnota = $hodnoty[0] / $hodnoty[1];
        $ref = $hodnota;
    }

    foreach($exif["GPSLatitude"] as &$ref){
        $hodnoty = explode('/', $ref);
        $hodnota = $hodnoty[0] / $hodnoty[1];
        $ref = $hodnota;
    }
}
```

Kód 4.6: Získání dat z EXIFU fotografie

```
}
```

Z kódu 4.6 je patrné, že v proměnných `$exif['GPSLongitude']`, `$exif['GPSLongitudeRef']`, `$exif['GPSLatitude']` a `$exif['GPSLatitudeRef']` jsou zeměpisná data o pořízení fotografie. Jediným problémem, který se musí ještě vyřešit, je nekompatibilita formátu *GPS* souřadnic získaných z *EXIFU* fotografie s formátem používaným v *Google Maps API*. Zeměpisná data pořízena z *EXIFU* jsou ve formátu *hhmmss*, kdežto objekt `google.maps.LatLng` vyžaduje formát desetinného čísla. Pro další použití těchto dat je potřeba *GPS* data pořízená z *EXIFU* přetvořit do formátu vyhovujícího danému *API*.



```
$aLon = array($exif["GPSLongitude"],
$exif['GPSLongitudeRef']);
$aLat = array($exif["GPSLatitude"],
$exif['GPSLatitudeRef']);
$gLat = ($aLat[0][0] + $aLat[0][1]/60 +
$aLat[0][2]/3600) * ($aLat[1] == "N" ? 1 : -1);
$gLon = ($aLon[0][0] + $aLon[0][1]/60 +
$aLon[0][2]/3600) * ($aLon[1] == "W" ? -1 : 1);
```

Kód 4.7: Převod souřadnic z formátu hhmss do desetinného formátu

Jakmile jsou GPS data pořízené z EXIFU fotografie ve správném formátu, pak jsou tato data společně s názvem fotografie uložena do databáze.

Poté je potřeba získané informace uložit do instance objektu `google.maps.LatLng` a vytvořit na mapě značku (*marker*) právě na místě, kde byla fotografie pořízena. Této značce pak přidáme informační okno, jehož obsahem bude právě naše fotografie. Po kliknutí na danou značku se vyobrazí informační okno a zároveň i pořízená fotografie.

V kódu 4.8 je patrný posluchač na danou značku. Tento posluchač naslouchá kliknutí na nějakou značku. Pokud klikneme na nějakou značku na mapě, posluchač zavolá funkci, která otevře informační okno s fotografií.

```
google.maps.event.addListener(marker_{$iterator-
    >getCounter()}, 'click', function() {
        infowindow_{$iterator-
    >getCounter()}.open(map,marker_{$iterator->getCounter()});
    });
```

Kód 4.8: Posluchač kliknutí na daný marker

Značka je vlastně objekt typu `google.maps.marker` a informační okno je objekt typu `google.maps.InfoWindow`. V kódu 4.8 je patrné, že instancemi na tyto objekty jsou položky pole daného typu. Důvodem je možnost umístění více markerů s informačními okny na mapu. Nejdůležitější použité parametry těchto objektů jsou přehledně vypsány v následujících tabulkách.



position	Na jaké pozici bude marker vykreslen
map	Do jaké mapy se má marker vykreslit
title	Zobrazovaný titulek

Tab 4.3: *Nejpoužívanější parametry objektu google.maps.Marker*

content	Obsahuje to, co bude v informačním okně vyobrazeno (text, kus HTML kódu, atd.)
position	Na jaké pozici bude informační okno vyobrazeno
MaxWidth	Maximální šířka informačního okna

Tab 4.4: *Nejpoužívanější parametry objektu google.maps.InfoWindow*

Kompletní výčet parametrů a metod daných objektů je k dispozici v dokumentaci *Google Maps API* verze 3. Vytváření jednotlivých instancí daných objektů je patrné v ukázce kódu 4.9.

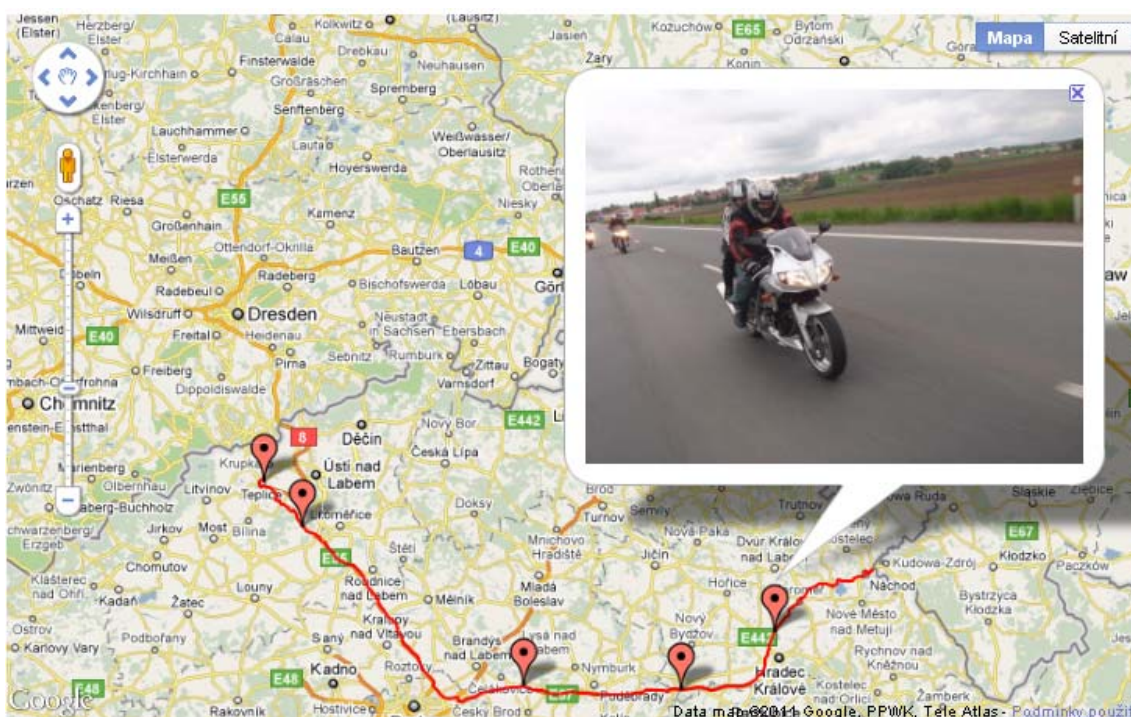
```
{foreach $fotos as $foto}
    var myLatLng=new google.maps.LatLng({$foto->lat},
    {$foto->lon});
    var picture_{$iterator->getCounter()} = {$foto->foto};
    var contentString_{$iterator->getCounter()}='';
    var infowindow_{$iterator->getCounter()} = new
    google.maps.InfoWindow({ content:
    contentString_{$iterator->getCounter()} });
    var marker_{$iterator->getCounter()} = new
    google.maps.Marker({
        position: myLatLng,
        map: map,
        title:{$foto->foto}
    });
    google.maps.event.addListener(marker_{$iterator->
    getCounter()}, 'click', function() {
    infowindow_{$iterator->
    getCounter()}.open(map,marker_{$iterator->
    getCounter()}); }); }
```



```
{/foreach}
```

Kód 4.9: deklarace instancí objektů `google.maps.Marker(InfoWindow)`

Výsledky celého snažení jsou vyobrazeny na obrázku Obr 4.3. Obrázek obsahuje mapu, kde je vyobrazena ujetá trasa a markery, které jsou umístěny na pozicích odpovídajících místům, kde byly fotografie pořízeny. Dále je také vidět informační okno, které obsahuje fotografii odpovídající umístění daného markeru.



Obr 4.3: Mapa obsahující ujetou trasu a fotografie pořízené po cestě



5. Geotagging

Ve zkratce je geotagging metoda, pomocí které se do fotografií ukládají zeměpisné souřadnice. Pokud se vezmeme do situace, že procestujeme poměrně velkou oblast v delším časovém horizontu a nafotíme velké množství fotografií, může se stát, že při následném prohlížení zapomeneme, kde vlastně byla fotografie pořízena. Pokud fotografie otaggujeme, tak se výše zmíněnému problému vyhneme. Otaggované fotografie můžeme pomocí různých programů zobrazit přímo na mapě a víme tedy, kde byla jaká fotografie pořízena. Geotagging je tedy i velice příjemným pomocníkem při prezentaci fotografií druhé osobě.

Informace o zeměpisné poloze pořízení fotografie se ukládají do *EXIFU* a tudíž není nijak změněna kvalita fotografie. *EXIF* je specifikací pro formát *metadat*, která jsou vkládána do souborů digitálními fotoaparáty.

Metadata v *EXIFU* mohou obsahovat následující informace:

- Značku a model fotoaparátu
- Datum a čas pořízení fotografie
- Citlivost, clonu, expoziční čas, ohniskovou vzdálenost, ostření, zoom atd.
- Informace o místu pořízení fotografie (geotagging)
- Komentáře a informace o autorovi

5.1 Způsoby otaggování fotografií

Fotografiím lze přidat informace o zeměpisné poloze dvojím způsobem. To je závislé na fotoaparátu, kterým fotografie pořizujeme. Pokud vlastníme fotoaparát, který disponuje *GPS* modulem je otaggování fotografií velice jednoduché. V menu přístroje aktivujeme možnost přidání zeměpisné informace ke snímku. K samotnému přiřazení zeměpisné informace do *EXIFU* fotografie dochází při stisknutí spouště fotoaparátu. Tento způsob geotaggingu je velice jednoduchý a intuitivní avšak obrovskou nevýhodou je poměrně vysoká pořizovací cena fotoaparátu, který disponuje *GPS* modulem.

Fotografové, kteří nevlastní výše zmíněný přístroj však nemusí zoufat. Existuje další způsob otaggování fotografií. Jedná se o spárování *GPX* souboru a galerie fotografií. Existuje velká škála programů, které toto spárování dokáží uskutečnit. Tyto



programy pracují na principu synchronizace času záznamu v *GPX* souboru a času pořízení fotografie uloženého v *EXIFU*.

Nejpřívětivějším nástrojem pro otaggování fotografií je program Zoner Photo Studio. Tento program je k dispozici v češtině a práce v něm je velice přehledná. V následující podkapitole bude stručně popsán postup otaggování fotografie v nástroji Zoner Photo Studio 11.

5.2 Otaggování fotografie pomocí Zoner Photo Studio

V prostředí Zoner Photo Studio je práce s fotografiemi velice jednoduchá. Veškeré panely jsou uživatelsky přívětivě rozmístěné. Otaggování fotografií je pomocí tohoto nástroje velice intuitivní.

V levé části okna programu se nachází okno navigátor. Zde stačí vybrat adresář, ve kterém jsou fotografie, ke kterým chceme přidat informaci o poloze jejich pořízení. Tyto fotografie, respektive jejich miniatury, se vyobrazí v okně průzkumník. V pravé části obrazovky je pak vyobrazen kompletní *EXIF* aktuální fotografie.

Pro samotné otaggování je třeba označit fotografie, kterým budeme předávat informace o poloze pořízení. V hlavním menu zvolíme Informace a následně položku *GPS*. Poté zvolíme možnost přidat *GPS* údaje. Následně se objeví okno, které nám umožňuje nahrát *GPX* soubor, pomocí kterého se spárují data fotografie a data ze záznamu trasy.

Po potvrzení se informace o poloze pořízení přidají do *EXIFU*. Pokud nyní Zkontrolujeme *EXIF* fotografie, zjistíme že byly přidány informace o *GPS*. Tyto informace obsahují zeměpisnou šířku, délku a nadmořskou výšku. Souřadnice polohy pořízení fotografie jsou ve formátu *hhmmss*.

Pokud zvolíme v hlavním menu programu volbu Informace, následně volbu *GPS* a dále položku Zobrazit *GPS* údaje... , objeví se okno, které obsahuje mapu a *marker*. Tento *marker* je umístěn přímo na souřadnicích odpovídajících místu pořízení fotografie. Dokonce máme možnost zvolit si z několika mapových podkladů. Na výběr jsou mapy od firmy Google, Seznam, nebo možnost zadání vlastního odkazu.



6. Webová aplikace Cestopisný motorkářský portál

Na nápad vytvoření cestopisného motorkářského portálu mě přivedla vlastní zkušenost. Já sám jsem vášnivým motorkářem a víkend co víkend ve společnosti přátel, kteří taktéž holdují jednostopým silničním strojům, vyrážíme na vyjížděky po naší krásné vlasti. Často se však stává, že v místě setkání před vyjížděkou trávíme spoustu času nad diskuzí, kudy že ta dnešní trasa vlastně povede. Tato debata se často natahuje až na několik desítek minut. Vzhledem k tomu, že motorkářské oblečení není zrovna styl plážových chlapců, je postávání ve třicetistupňových teplotách opravdu nepříjemnou záležitostí.

Tyto nemilé zkušenosti mě přivedli na nápad vytvoření dynamického a interaktivního cestopisného portálu. Takže místo dlouhých diskuzí někde na parkovišti stačí pouze otevřít internetový prohlížeč, zadat adresu a v klidu si prohlédnout trasy a fotografie pořízené našimi kolegy motorkáři. Pak nezbývá nic jiného, než vybrat nejvhodnější trasu pro výlet, rozeslat přátelům odkaz a o možných změnách debatovat prostřednictvím internetových komunikačních nástrojů z pohodlí domova.

V dnešní době rozmachu satelitních navigací a různých *GPS* zařízení mě napadlo tohoto trendu využít. Již existujícím cestopisným motorkářským portálům chybí dvě zásadní funkce. Jedná se o vyobrazení ujeté trasy na mapě a možnost zobrazení otaggovaných fotografií přímo na mapě. Tyto dvě funkce zásadně mění pohled uživatele na funkci motorkářského cestopisného portálu. Pokud si uživatel může prohlédnout, kudy přesně daná trasa vede a prohlédnout si fotografie přímo v místě jejich pořízení, pak je jeho rozhodování ohledně zvolení trasy mnohem jednodušší. Někteří motorkáři volí raději přímé a rovné úseky dálnic, někteří preferují klikatící se cesty po okresních komunikacích a někteří zase preferují jízdu lesem a po polních cestách. Uživateli se stačí podívat na profil trasy a hned ví, zdali tato trasa vyhovuje jeho požadavkům. Možnost uživatele prohlédnout si fotografie pořízené během ujetí trasy je také obrovskou výhodou. Uživatel přesně ví kde se daná scenérie, či zajímavé místo nachází a tudíž může předem naplánovat zastávky.

Pro perfektní fungování portálu je důležité nejen data získávat, ale taky přidávat nové záznamy o ujetých trasách a nahrávat otaggované fotografie.



Jak již jsem zmiňoval, v praktické části této diplomové práce jsem se zaměřil na vytvoření doplňujících funkcí pro cestopisné motorkářské portály. Jedná se o funkci zobrazení ujeté trasy na mapě a zobrazení otaggovaných fotografií na mapě.

Aby uživatel mohl vkládat trasu a otaggované fotografie, musí vlastnit nějaký *GPS* přijímač, digitální fotoaparát a samozřejmě počítač připojený k síti internet. Při zahájení cesty je potřeba na *GPS* přijímači zapnout funkci trasování. Trasování je třeba vypnout po dokončení vyjížďky. Následně uživatel vygeneruje *GPX* soubor (pomocí *SW* dodávaného k *GPS* zařízení), otagguje fotografie (pokud nevlastní fotoaparát s vestavěným *GPS* modulem) a *GPX* soubor a otaggované fotografie nahraje pomocí rozhraní portálu na server.

Aby bylo možné vykreslit ujetou trasu, musí aplikace rozparsovat *GPX* soubor. *GPX* soubor je vlastně *XML* soubor obsahující velké množství záznamů o zeměpisné poloze, nadmořské výšce a datu a času pořízení daného záznamu. Tyto záznamy vznikají během trasování *GPS* zařízení. Ukázka části *GPX* souboru je patrná v kódu 6.1.

```
<?xml version="1.0" encoding="UTF-8" ?>
<gpx>
  <trk>
    <name>Tour-20100312-1505</name>
    <trkseg>
      <trkpt lat="51.1740106344223"
lon="12.007450461387634"><time>2010-03-
12T15:05:46Z</time></trkpt>
      <trkpt lat="51.1740106344223"
lon="12.007450461387634"><time>2010-03-
12T15:05:47Z</time></trkpt>
      <trkpt lat="51.17485284805298"
lon="12.010400891304016"><time>2010-03-
12T15:05:47Z</time></trkpt>
      <trkpt lat="51.17526590824127"
lon="12.011865377426147"><time>2010-03-
12T15:05:49Z</time></trkpt>
```

Kód 6.1: Ukázka části GPX souboru

Jedná se tedy o mashup aplikaci. Mashup (v kontextu internetu) je definován jako webová stránka nebo aplikace, která užívá obsah z více zdrojů k vytvoření nové služby. Obsah využívaný v mashup aplikaci bývá zprostředkován ze zdrojů třetí strany pomocí veřejného rozhraní. V tomto případě se jednalo o Google Maps API.

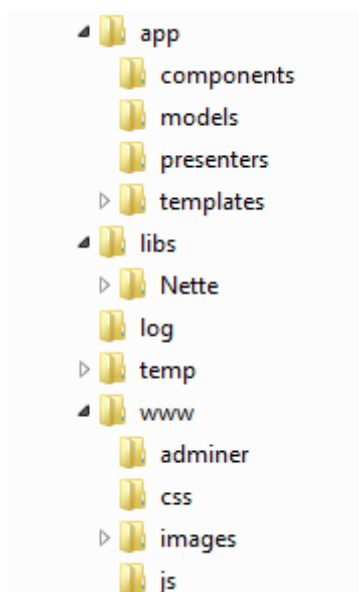


Podle univerzálnosti jádra aplikace je patrné, že při jednoduchém předělení obsahu je aplikaci možné nasadit na celou řadu jiných odvětví. Dovedu si představit použití aplikace jako „databázi“ realit jednotlivých agentů v realitní kanceláři, silný nástroj pro dopravní zpravodajství, apod.

6.1 Struktura aplikace a použité technologie

Celá aplikace je napsaná v PHP frameworku Nette. Jedná se tedy o aplikaci s třívrstevným modelem MVC (Model, View, Controller). Tato softwarová architektura vznikla za účelem oddělit u aplikací s grafickým rozhraním kód obsluhy aplikace (controller) od kódu aplikační logiky (model) a od kódu jež zobrazuje výsledné data (view). To vše má za důsledek zpřehlednění celé aplikace a usnadnění budoucího vývoje aplikace. Další výhodou modelu MVC je fakt, že je možné testovat aplikaci po jednotlivých částích což se ocení hlavně při odlazování aplikace.

Aplikace má pevně danou adresářovou strukturu. V adresáři app jsou umístěny modely (models), views (templates) a controlls (presenters). Adresář libs obsahuje veškeré knihovny Nette frameworku a adresář www je kořenovým adresářem aplikace. Zde je obsažen soubor index.php a adresáře v nichž jsou obsaženy soubory jako například definice kaskádových stylů a různé podpůrné skripty.



Obr 6.1: Adresářová struktura třívrstvé aplikace (MVC)



6.1.1 Model

Model je datovým a funkčním základem celé aplikace a obsahuje aplikační logiku. Tato vrstva komunikuje s databází. Jakákoli akce uživatele, u které je potřeba přistupovat k databázi představuje akci modelu. Pro snadnější práci s modely je v aplikaci použita továrna pro dynamické načítání modelů.

6.1.2 View

Tato vrstva aplikace se stará o zobrazení požadavku uživatele. Obvykle používá šablonovací systém a definuje, jak se má daný výsledek zobrazit. View obsahuje povětšinou HTML tagy a latte makra.

6.1.3 Controller

Je vlastně řadičem, který zpracovává požadavky od uživatele a na jejich základě volá patřičnou aplikační vrstvu (model). Dále controller požádá view o vykreslení výsledných dat. Ve frameworku Nette funkci controllerů zastávají tzv. presentery.

6.2 Aplikace z hlediska tříd

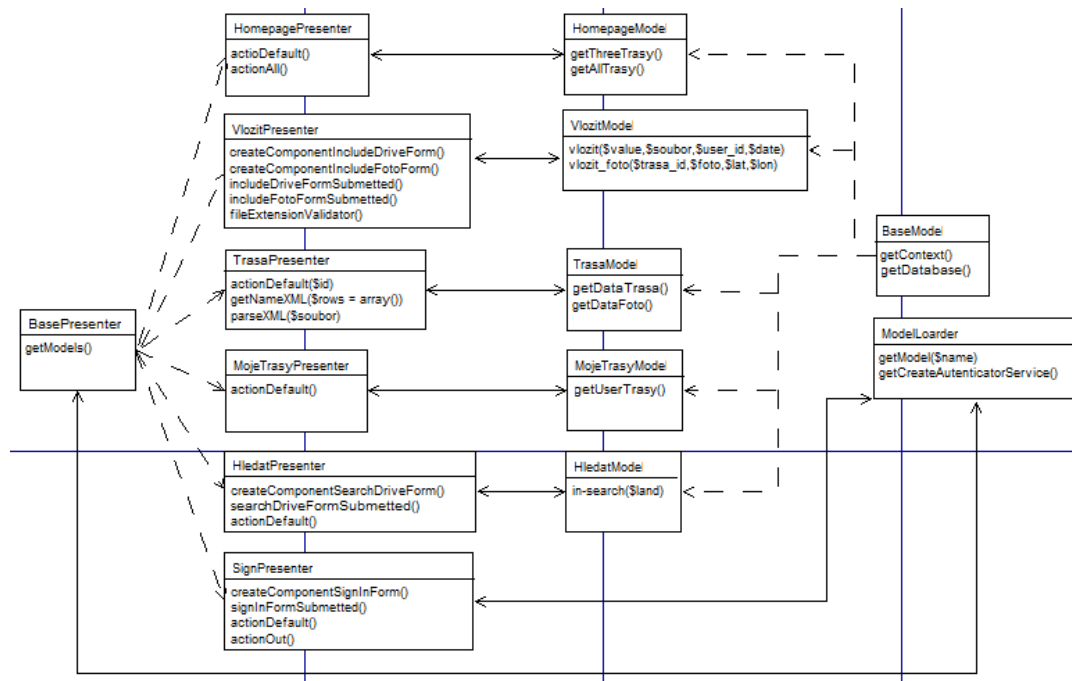
Aplikace cestopisný motorkářský portál poskytuje uživatelům přehled o uložených trasách ve formě výpisu obsahujícího nejen základní informace o trasách v podobě textového popisu, ale i interaktivní pohled na ujetou trasu. Interaktivita pohledu na ujetou trasu je zajištěna vykreslením ujeté trasy na mapovém podkladu Google Maps a možnosti prohlédnout si fotografie pořízené po dobu jízdy přímo v místě jejich vzniku. Implementování těchto dvou funkcí bylo zajištěno pomocí zpracování vstupních souborů (*GPX* soubor a exif otagovaných fotografií) a dále využitím Google Maps API. Podrobný popis získávání a zpracování *GPS* dat z *GPX* souboru byl popsán v kapitole 2.2 a získávání a zpracování dat z exifu fotografie bylo popsáno v kapitole 4.3.3.

Aplikace tedy umožňuje uživateli prohlédnout si trasy, vyhledávat trasy podle země, zobrazit trasy uživatele a samozřejmě možnost vkládat vlastní trasy.

Na hlavní stránce je umístěna tabulka se třemi nejaktuálnějšími trasami. Tato data jsou získána za pomoci presenteru `HomepagePresenter`, modelu `HomepageModel` a latte šablon v podadresáři `Homepage` adresáře `templates`.



Podobným způsobem jsou zajištěny i ostatní funkce aplikace. Ke každé funkci je přiřazen určitý presenter model a šablona. Vše je patrné na UML diagramu (Obr 6.2).



Obr 6.2: UML diagram tříd

Z UML diagramu (Obr 6.2) je patrné logické rozdělení aplikace. Každá z významných funkcí aplikace je reprezentována vlastním presenterem a příslušným modelem. V diagramu už nejsou vyobrazeny šablony, protože se nejedná o třídy jako takové. Latte šablony obsahují HTML kód a latte makra pomocí nichž se dá do HTML kódu vkládat určitá logika a funkcionalita, hlavně podmínky, cykly a proměnné ve kterých jsou data získaná z presenteru. Příklad šablony obsahující latte makra je patrný v kódu (Kód 6.2)

```
<table border="1">
  <tr>
    <th>Název</th>
    <th>Země</th>
    <th>Lokalita</th>
    <th>Datum</th>
  </tr>
  {foreach $rows as $row}
  <tr>
    <td><a n:href="Trasa:default $row->id">{$row->nazev}</a></td>
    <td>{$row->land}</td>
    <td>{$row->lokalita}</td>
```



```
<td>{date("d-m-Y", $row->createdat)}</td>
</tr>
{/foreach}
</table>
```

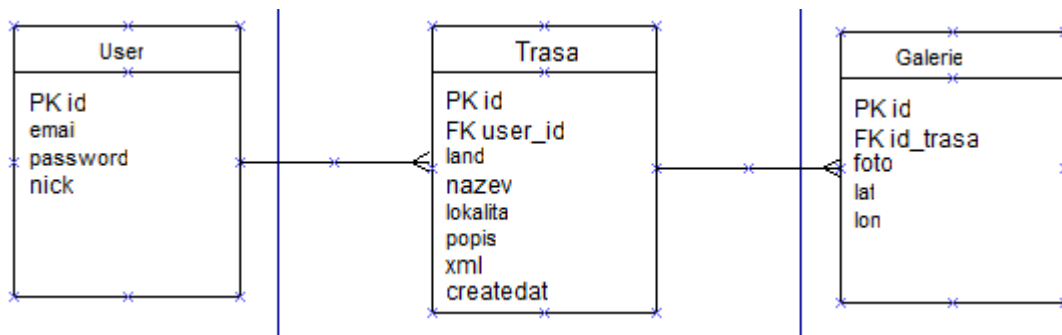
Kód 6.2: Ukázka latte šablony

Latte makra se zapisují do složených závorek `{}`. Tato šablona se stará o předání výsledného výpisu tras, jež přidal přihlášený uživatel. Pomocí cyklu `foreach` se projdou všechny záznamy požadovanými daty a v každém průchodu cyklem se vygeneruje řádek tabulky, obsahující odkaz pro vykreslení trasy a otaggovaných fotografií na mapě, zemi, lokalitu a datum vložení trasy do systému.

6.3 Aplikace z hlediska databáze

Jak už jsem několikrát zmiňoval, jádro této aplikace je možné použít pro jakékoli aplikace kde je zapotřebí nějakým způsobem zpracovávat GPS data z *GPX* souboru a z exifu fotografií a tyto data poté vyobrazit na mapovém podkladu.

Pro aplikaci cestopisného motorkářského portálu s možností vyobrazit ujetou trasu a otaggované fotografie na mapovém podkladu stačilo vytvořit poměrně jednoduchou databázovou strukturu. Tato databáze se sestává ze třech tabulek *user*, *trasa* a *galerie*. Tabulka *user* obsahuje data o uživateli (email, nick a heslo), tabulka *galerie* obsahuje data o nahraných fotografiích (identifikátor trasy, název fotografie, GPS souřadnice) a tabulka *trasa* uchovává informace o vložené trase (identifikátor trasy a uživatele, zemi, lokalitu, popis, název trasy, název *GPX* souboru a časové razítko).

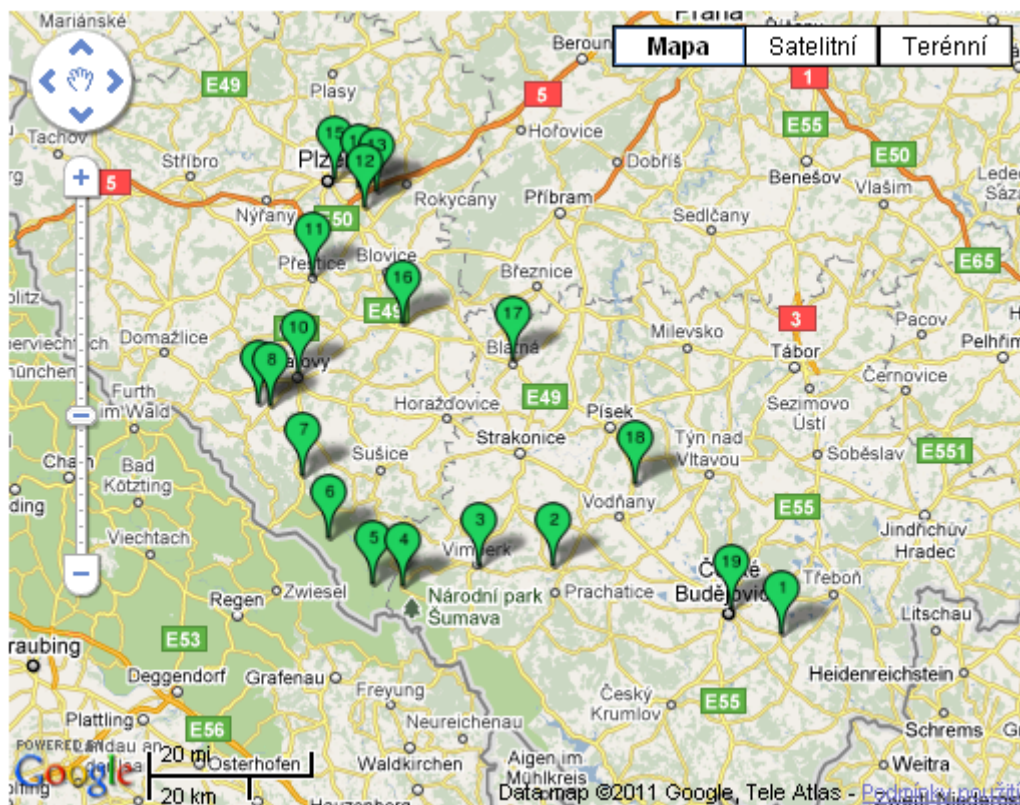


Obr 6.3: ERD diagram



6.4 Praktická ukázka konkureschopnosti aplikace

Na následujících obrázcích si dovoluji prezentovat rozdíl v zobrazení ujeté trasy v projektu Cestopisný motorkářský portál, což je praktická část mé diplomové práce, se zobrazením ujeté trasy z cestopisů nejnavštěvovanějšího českého motorkářského portálu na webu.



Obr 6.4: Vyobrazená ujetá trasa z cestopisu nejnavštěvovanějšího českého motorkářského portálu

Z obrázku (Obr 6.4) je patrné, že ujetá trasa je vyznačena diskrétně pomocí očíslovaných *markerů*. Není tedy jasné, kudy trasa vedla mezi jednotlivými *markery*. Samozřejmě je zde patrná absence vyobrazení otaggovaných fotografií na mapě. Toto řešení je dle mého názoru neefektivní a uživateli přináší jen malé množství informací o ujeté trase.

Na následujícím obrázku (Obr 6.5) je vyobrazena trasa, kterou jsme podnikli minulý rok v létě. Jedná se o trasu z Plzně do Náchoda. Na obrázku je patrné, že ujetá trasa je zobrazena spojitě, tudíž uživatel si může trasu prozkoumat téměř metr po metru. Dále jsem vybral zajímavé fotografie z cesty, otaggoval a pomocí aplikace je zobrazil



na mapě. Rozdíl mezi prvním a druhým způsobem vyobrazení ujeté trasy na mapě posuďte sami.



Obr 6.5: *Vyobrazení ujeté trasy a otaggovaných fotografií na mapě pomocí webové aplikace Cestopisný motorkářský portál*



Závěr

V této diplomové práci jsem popsal princip *GPS* systému a princip jeho komunikace s *GPS* přijímači. Dále jsem popsal postupy zabývající se získáváním dat z *GPS* přijímačů, především pak záznamy ujetých tras a jejich následné parsování. Samozřejmě jsem popsal i způsob předání *GPS* dat webové aplikaci, pro jejich následné zpracování. Nakonec jsem popsal i problematiku geotaggingu a veškeré výše zmíněné informace a postupy jsem využil k vytvoření webové aplikace demonstrující dosažené cíle této diplomové práce.

Webová aplikace byla naprogramována jako třívrstvá architektura MVC. Aplikace je tedy napsaná velice přehledně a je tak snadno rozšiřitelná. Jádro aplikace je použitelné pro širokou škálu aplikací, u kterých je žádoucí zpracovávat *GPS* souřadnice. K vytvoření aplikace cestopisného motorkářského portálu mě vedla vlastní iniciativa a vášeň pro jednostopé stroje a absence kvalitně prezentovaných cestopisů na současném webu.

Pro práci s mapovými podklady jsem využil *Google Maps API*. Práce s tímto podpůrným programovacím rozhraním pro vkládání mapových podkladů do webové aplikace byla velice příjemná. Veškeré objekty, funkce a vzorové příklady jsou přehledně uvedeny v dokumentaci *Google Maps API*. Pro testování chování funkcí a objektů, které *Google Maps API* poskytuje, jsem využil nástroje Google Code Playground. Mapové podklady od společnosti Google jsem zvolil z důvodu mapového pokrytí celého světa a pro skvělé *API*.

Jak již bylo v tomto dokumentu několikrát zmíněno, jako praktickou část práce jsem vytvořil mashup aplikaci, která využívá znalostí a postupů uvedených v tomto dokumentu. Aplikace disponuje dvěma funkcemi, které postrádám u dnes dostupných motorkářských cestopisných portálů. Jedná se o vyobrazení ujeté trasy na mapě a o zobrazení otaggovaných fotografií na mapovém podkladu. Tyto dvě funkce se mi nakonec opravdu podařilo vytvořit. Během vývoje jsem narazil na problém zvolení správného objektu pro vykreslení trasy, ale nakonec se mi testováním v Google Code Playground podařilo zvolit vhodný objekt.



Veškeré informace jsem čerpal z materiálů, které jsou uvedeny v seznamu použité literatury. Tímto jsem splnil veškeré body zadání diplomové práce a největší přínos mé práce vidím v možnosti rozšířit současné motorkářské cestopisné portály o funkce, které demonstruji v praktické části této práce a zároveň v univerzálnosti jádra této aplikace. Uživatel má mnohem jasnější pohled na zkoumanou trasu, může si ji prozkoumat takřka metr po metru a prohlédnout si fotografie z cesty, jež jsou vyobrazeny na mapě, přímo na místě jejich vzniku.

Tato aplikace slouží k demonstračním účelům pro obhajobu této diplomové práce, ale bylo by možné ji okamžitě nasadit jako rozšíření současných motorkářských portálů. Aplikace se dá dále snadno rozšiřovat o další funkce. Informace o trase by se dala rozšířit třeba o výškový profil trasy, délce trasy, průměrné rychlosti nebo informace o časové náročnosti ujetí dané trasy.



Seznam použité literatury

- [1] Gary Bollinger, Bharathi Natarajan JSP Java Server Pages podrobný průvodce začínajícího tvůrce webu. Grada Publishing, a.s., 2003. Vydání první. ISBN: 80-247-0340-8.
- [2] Barry Burd JSP: Java Server Pages podrobný průvodce. Computer Press, a.s., 2003. Vydání první. ISBN: 807226804X.
- [3] Steiner I., GPS od A do Z, eNav Praha, 2006. Vydání první. ISBN: 80-239-7516-1
- [4] Apache Tomcat. [online] . [2010-10-14]. URL: <http://tomcat.apache.org>
- [5] GLOBAL POSITIONING SYSTEM. [online] . [2010-10-14]. URL: <http://www.gps.gov/>
- [6] Hojgr R., GPS: praktická uživatelská příručka , Computer Press, Brno, 2007. Vydání první. ISBN: 80-251-1734-7
- [7] Dobrovodský J., Geotagging v praxi [online] . [2009-9-7]. URL: <http://www.fotoaparát.cz/article/10819/1>
- [8] *Google Maps JavaScript API V3.* [online] . [2010-10-14]. URL: <http://code.google.com/intl/cs/apis/maps/documentation/javascript/>

